Evolutionary and quantum-inspired algorithms for the optimization of magnetic cooling systems

Anna Ouskova Leonteva

UNIVERSITÉ DE STRASBOURG

ÉCOLE DOCTORALE MATHEMATIQUES, SCIENCES DE L'INFORMATION ET DE L'INGENIEUR – ED269

Laboratoire ICube (UMR CNRS 7357)

THÈSE présentée par :

Anna OUSKOVA LEONTEVA

soutenue le : 24 **Mai 2024**

pour obtenir le grade de : **Docteur de l'université de Strasbourg**

Discipline/ Spécialité : Informatique

Evolutionary and Quantum-inspired Algorithms for the Optimization of Magnetic Cooling Systems

THÈSE dirigée par :

M COLLET Pierre	Professeur, université de Strasbourg, ICube
M PARREND Pierre	Professeur HDR, EPITA, ICube

RAPPORTEURS :

Mme LUTTON Evelyne	Directrice de recherche, INRAE, EKINOCS
M BALLI Mohamed	Professeur, université internationale de Rabat

AUTRES MEMBRES DU JURY :

M VEREL Sébastien	Professeur, université du Littoral Côte d'Opale
M FRUCHART Daniel	Directeur de recherche, institut Néel CNRS Grenoble UPR 2940
Mme JEANNIN-GIRARDON Anne	Maitresse de conférences, université de Strasbourg, ICube
M WHITLOCK Shannon	Professeur, université de Strasbourg, ISIS

INVITÉS : M RISSER Michel Coordinateur scientifique du projet, Ubiblue

Dedicated to my parents.

Abstract

Magnetic refrigeration is an innovative technology, which has the potential to reduce energy consumption, depletion of the ozone layer and to slow down the rate of global warming. The principle of magnetic refrigeration is based on the Magneto Caloric Effect (MCE) manifested in Magneto Caloric Materials (MCMs). Even though this technology seems very promising, there are still some challenges, impeding its large-scale deployment. In order to develop a commercially applicable prototypes, in the frame of the *CoolMagEvo* ANR project (ANR-17-CE05-0036), the present thesis focuses on two major research problems: (i) to reproduce and study physical properties of different MCMs; (ii) to investigate the Active Magnetic Regenerator (AMR) design for various specific applications needs: e.g., a Magnetic Refrigeration System (MRS) and a Thermo-Magnetic energy Generator (TMG).

We propose to solve both research problems by applying a flexible optimization tool for various case studies of the computationally intensive simulation models of MCMs and the AMR. Thus, each case study has to be formulated as a new optimization problem. In this context, the major contributions are as follows.

Firstly, for solving computationally intensive problems, two optimization algorithms are proposed: (i) a parallel archive-based algorithm for multi-/many-objective problems, FastEMO; (ii) a quantum-inspired algorithm for single-objective problems, QAES.

Secondly, for studying various research cases with the same algorithm, we developed a unified optimization algorithm, QIU-NSA, which is scalable w.r.t. the number of objectives and decision variables;

Next, we integrated the algorithms in the updated version of the EASEA (*EAsy Specification of Evolutionary Algorithms*) platform for providing a convenient software tool with a user-friendly interface for scientists;

Finally, the developed tool is used as a software support for solving research problems thanks to the following provided contributions: (i) a novel optimization-based method for reproducing and studying the physical properties of different MCMs; (ii) an optimization-based study of the impact of the control and design parameters of the dual-mode (MRS and TMG) operating AMR model on its performance.

The experimental validations of the proposed contributions are provided. A common design of the AMR model for the MRS and the TMG modes has been found.

Acknowledgements

I would like to express my special appreciation and thanks to my thesis directors, Pierre Collet and Pierre Parrend, for their support and supervision. I appreciate all their contributions of time, ideas and constructive discussions to help me overcome the difficulties during my research.

I am very grateful to my tutor, Anne Jeannin-Girardon, for all help and good advice she has given me.

I would like as well to express my deep gratitude to the *CoolMagEvo* ANR project for its financial support and all my colleagues from this project, especially Radia Hamane (the *Crismat* Laboratory) and the scientific coordinator of the project, Michel Risser (the *Ubiblue Company*) for their efficient collaboration.

I would like to use an opportunity to thank Shannon Whitlock for his wonderful quantum mechanics course, for his helpful discussions and insightful comments.

Next, I am grateful to Ulviya Abdulkarimova and Tobias M. Wintermantel for their collaboration.

I sincerely thank Evelyne Lutton and Mohamed Balli for reviewing my thesis. I also would like to thank Sébastien Verel, Daniel Fruchart and Shannon Whitlock for kindly accepting to be members of the Jury, it is an honor for me.

I would like also to thank all the CSTB team, ICube, for their help, support and enjoyable environment.

I am especially grateful to Olivier Poch and Julie Thompson for their help at the beginning of my research way.

A special and particular thanks to Anne Ney for her friendship and great support.

Finally, I want to thank my family.

Contents

List of Figures

List of Tables

List of Acronyms

ACD Alternative Crowding Distance

AMR Active Magneto caloric Regenerator

ANR National Research Agency

API Application Programming Interface

ASREA Archived-based Stochastic Ranking Evolutionary Algorithm

BIPOP-CMAES Bi–Population Covariance Matrix Adaptation Evolution Strategy

BBOB Black-Box Optimization Benchmarking

BEG-I Blume–Emery–Griffiths–Ising model

BFGS Broyden–Fletcher–Goldfarb–Shanno algorithm

CD Crowding Distance

CDAS Controlling Dominance Area of Solutions

CMA-ES Covariance Matrix Adaptation Evolution Strategy

COP Coefficient Of Performance

DRS Dominance Resistant Solutions

DTLZ Deb–Thiele–Laumanns–Zitzler

EA Evolutionary Algorithm

EASEA Easy Specification of Evolutionary Algorithms

ECDF Empirical Cumulative Distribution Function

ERT Expected Run Time

ES Evolution Strategy

FastEMO Fast Evolutionary Multi-objective Optimization

FFT Fast Fourier Transform

FOT First Order phase Transition

GPU Graphics Processing Unit

HV Hypervolume

IBEA Indicator Based Evolutionary Algorithm

MAPE Mean Absolute Percentage Error

MCE Magneto Caloric Effect

MCM Magneto Caloric Material

MFT Mean Field Theory

MO-CMA-ES Multi-Objective Covariance Matrix Adaptation Evolution Strategy

MOEA Multi-Objective Evolutionary Algorithm

MOED-D Multi-Objective Evolutionary Algorithm Based on Decomposition

MOP Multi-objective Optimization Problem

MQHOA Multi-scale Quantum Harmonic Oscillator Algorithm

MRS Magnetic Refrigeration System

NDS Non-Dominated Sorting

NISQ Noisy Intermediate Scale Quantum

NSGA-II Non-dominated Sorting Genetic Algorithm II

NSGA-III Non-dominated Sorting Genetic Algorithm III

PSO Particle Swarm Optimization

QAES Quantum-inspired Algorithm with Evolution Strategy

QAOA Quantum Approximate Optimization Algorithm

QI Quality Indicator

QIU-NSA Quantum-Inspired Unified Non-dominated Sorting Algorithm

QPSO Quantum-behaved Particle Swarm Optimization

SBX Simulated Binary Crossover (Xover)

TMG Thermo-Magnetic energy Generator

SOT Second Order phase Transition

SPEA-II Strength Pareto Evolutionary Algorithm II

VQE Variational Quantum Eigensolver

WFG Walking Fish Group

ZDT Zitzler–Deb–Thiele

Part I

General Introduction

Chapter 1

Introduction

1.1 Context

1.1.1 Environmental Problems Caused by Vapor-Compression Refrigerators

Nowadays, the demand for cooling systems is increasing, due to the climate change and universal problems: such as the current pandemic of Covid-19. Indeed, as vaccines are rolled out, they need enormous cold-storage systems for their manufacture, distribution and storage. According to a report by the *United Nations Environment Program* and the *International Energy Agency*, the number of global cooling devices is estimated to increase from 3.6 billion to 9.5 billion by 2050.

In the context of the high demand for cooling, refrigeration systems already account for around 17% of the worldwide electricity consumption [Coulomb et al. (2015)]. This is explained by the fact that the conventional vapor-compression refrigeration systems used for domestic and industrial needs, are not power efficient, because of their working principle, which is based on the mechanical compression and expansion of a gas.

Furthermore, the impact on global warming is one of the major challenges faced by the refrigeration industry, since the vapor-compression technology is not environmentally friendly: a compressor requires energy, which leads to off-site carbon dioxide (CO_2) emissions due to energy creation. Additional energy is required to either pump cooling water or to use a fan to move air, leading to additional indirect CO_2 emissions [Blowers & Lownsbury (2010)]. Another problem is that some refrigerants available in the market have a high ozone depletion potential, and especially those releasing chlorine. Refrigerants with low ozone depletion potential, such as ammonia or CO_2, have been introduced, but have safety-related issues such as flammability, toxicity (NH_3), high pressure (CO_2) and leakage of the refrigerant that results in direct CO_2-equivalent emissions [Agrawal & Matani (2012)], [Blowers & Lownsbury (2010)].

Consequently, with increasing attention to the problems of energy consumption, depletion of the ozone layer and the global warming together with significant growing of cooling and air-conditioning demand, the refrigeration industry needs innovative solutions to replace the vapor-compression technology.

1.1.2 Magnetic Refrigeration as a Solution for Environmental Problems

Magnetic refrigeration is an alternative technique to the vapor-compression technology based on the magneto caloric effect, where the cycles of compression/expansion of gas are replaced by the phases of magnetization/demagnetization of a magneto caloric material.

The **Magneto Caloric Effect** (MCE) is a temperature change of a solid refrigerant under application of magnetic field.

A **Magneto Caloric Material** (MCM) is a solid refrigerant endowing the MCE. A MCM heats up when a magnetic field is applied, and cools down, when the magnetic field is removed. Consequently, for commercial applications, magnetic refrigeration requires the combination of a relatively strong magnetic field and a refrigerant with a large magneto caloric effect [Balli et al. (2011)].

However, the MCE produced during these processes of magnetization/demagnetization, accounts for few Kelvin, which is not sufficient to achieve a commercially applicable refrigeration. In order to increase the magneto caloric effect, active magnetic regenerator is usually applied [Balli et al. (2011)].

An **Active Magnetic Regenerator** (AMR) comprises several magneto caloric materials, which are thermodynamically cycled for providing the refrigeration over an extended temperature range. The magneto caloric effect and a structure of the active magnetic regenerative cycle will be detailed in Section 2.3.2.

Magnetic refrigeration has been appointed as a promising technology to overcome the drawbacks of vapor-compression refrigeration [Balli et al. (2011)]. The main potential advantages of magnetic refrigeration system are:

1. No greenhouse gases emission, due to the use of a solid refrigerant - magneto caloric materials [Gombi & Sahu (2020)], [Brück (2005)], [Balli et al. (2011)].

2. Reduced noise during the operation of magnetic cooling system, due to the absence of compression and expansion processes [Brück (2005)].

3. The potential of electricity consumption reduction of 20% [Gombi & Sahu (2020)], comparatively with the vapor-compression refrigeration, due to the absence of compression and expansion processes as well.

4. The potential applicability as thermo-magnetic energy generators [Solomon (1991)], which transform the magnetic energy into electrical energy.

1.1.3 The *CoolMagEvo* ANR Project

Aiming at developing of a commercially applicable magnetic refrigeration system, the *CoolMagEvo* ANR project (ANR-17-CE05-0036, *A clean, safe and efficient energy*) was launched in 2017 as a collaborative research consortium with specialists from different scientific fields:

1. Multi-physics numerical simulation modeling magnetic cooling system and thermo-magnetic energy generators — the *Ubiblue* company[1]— the scientific coordinator of project.

[1] `https://ubiblue.com/`

2. Experimental measurements and theoretical simulations of physical properties of materials — the laboratory of Crystallography and Materials Science on Physics and Structure of Magnetic Oxides (*Crismat*)[2] of ENSICAEN (UMR CNRS 6508).

3. Evolutionary optimization — the Complex Systems and Translational Bioinformatics (CSTB)[3] research team of the ICube laboratory, Strasbourg (UMR CNRS 7357).

In order to reduce the global energy consumption, the global warming and the depletion of the ozone layer, the main aim of the *CoolMagEvo* ANR project is to explore the possibility to create a innovative commercially applicable magnetic system, working in the two modes: a magnetic cooling system and a thermo-magnetic energy generator.

1.2 Research Problems and Objective

This thesis can be presented by the following description.

Research Context: Since this thesis is carried out within the frame of the *CoolMagEvo* ANR project, its context matches with the main aim of this project defined in Section 1.1.3.

Research Problems: According to the context, the research problems are defined as follows:

1. Modeling/reproducing physical properties of magneto caloric materials for their further application in an active magnetic regenerator model or for creating databases. A reliable reproduction of physical properties of magneto caloric materials is required in order to guarantee the thermodynamic consistency and the perfect energy conservation in the active magnetic refrigerator model.

2. Investigating the impact of different combinations of control and design parameters of the active magnetic refrigerator model, working in two modes (a magnetic cooling system and a thermo-magnetic energy generator) on its performance.

Research Objects: The research objects of this thesis are the two following simulation models:

1. The model of physical properties of magneto caloric materials, provided by the *Crismat* laboratory. The description of this model is presented in Section 2.2.2.

2. The multi-physical and multi-scales numerical model of active magnetic regenerative refrigerator, provided by the *Ubiblue* company. The description of this model is presented in Section 2.3.3.

Each research problem correlates with the corresponding research object.

Both models have two common particularities: they are (i) computationally intensive and (ii) under active development. The first particularity imposes restrictions on the

[2] `https://crismat.cnrs.fr/`
[3] `https://cstb.icube.unistra.fr/`

number of possible simulations. The second one implies frequent modifications of models that introduce some changes into the results of model's work.

In order to better explain the second particularity, we provide the following example. The majority of the existing prototype devices for magnetic refrigeration are based on a thermodynamic cycle with an active magnetic regenerator, which operates as a Brayton-type regenerative magnetic refrigeration cycle. However, there are several other cycles that may potentially influence not only the efficiency, but also cost, compactness and simplicity of magneto caloric devices. Consequently, the *Ubiblue* company assumes to investigate different thermodynamic cycles with its active magnetic regenerator model: e.g., a hybrid Brayton-Ericsson cycle, where the good balance between Brayton and Ericsson cycles is a result of a compromise between the device complexity, its cost, its power density and its energy efficiency. Consequently, the modification of thermodynamic cycle has an impact on the working results of the model. Another example is connected with the properties of the used magneto caloric materials: a change of materials inside the active magnetic regenerator model also has the impact on the performance.

Moreover, such modifications require an individual investigation of the impact of different combinations of control and design parameters of the active magnetic regenerator model, where the parameter combinations can be different depending on the modifications.

Taking into account a large number of parameters, their combinations, modifications of the model and the model computational intensiveness, it is obvious that this huge number of research cases cannot be investigated by hands. Consequently, a special technique and software support is required in order to automate the investigation process of the design of active magnetic regenerator.

In the next paragraph, we introduce the research subjects of this thesis, which are applied to these research objects in order to solve the research problems.

Research Subjects: The research subjects of this thesis are defined as evolutionary and quantum-inspired optimization algorithms applied to the model of magneto caloric materials and the active magnetic regenerator model for solving the research problems.

Our choice of optimization algorithms as the research subjects of the thesis is explained by the following reasons.

1. The working principle of optimization algorithms fits for solving both research problems of this thesis: i.e., a selection of the best solution, with regard to some criteria, from some set of available alternatives. In the simplest case, an optimization process consists of maximizing or minimizing an evaluation function by systematically choosing the input values from an allowed set and computing the value of the function.

2. Evolutionary optimization algorithms demonstrate the capability to improve and accelerate a design process of different complex numerical simulation models when the best solution can't be found by hands. It has been proved that an optimization approach can reduce time, financial and technical cost for solving the research problems in different scientific areas.

3. We take into account the nature of the models and their particularities, in order to apply the most appropriate optimization algorithms for efficiently solving the

research problems. E.g., we turn to quantum-inspired optimization algorithm for solving the problems of the model of magneto caloric materials, due to its quantum structure.

As already mentioned in the previous paragraph, both models are under active development and can be modified. Consequently, these modifications have an impact on the formulation of the optimization problems. Many different optimization problems can be formulated for each model according to its modifications: i.e., the size of search space (the number of input variables), the size of objective (the number of objectives), etc.

Moreover, different optimization problems can be formulated for the same model. E.g., for modeling physical properties of magneto caloric materials, single- and multi-objective optimization problems can formulated, depending on the number of studied physical properties of materials, which have to be considered.

Consequently, an optimization algorithm scalable w.r.t. the number of objectives and the number of decision variables is required.

For investigating and developing appropriate optimization algorithms, we explored both models and defined the following features of its hypothetical optimization problems:

1. From an optimization point of view, a problem for the active magnetic regenerator model is considered as:
   - a multi-objective (2, 3 objectives) or many-objective problem (> 3 objectives);
   - a small, medium or large scale problem;
   - a problem with Dominance Resistant Solutions (DRS): i.e., some solutions can have extremely good values for some objectives and extremely bad values for other objectives;
   - a problem, where several non-dominated solutions can be identical in the objective space and different in the search space;
   - a black-box optimization problem whose detailed internal structure and code can be unavailable, since the optimization functions correspond to a commercial software;
   - a dynamic problem (in the future works).
2. From an optimization point of view, a problem for the model of magneto caloric materials can be considered as:
   - a single- or multi-objective problem;
   - a small scale problem;
   - a separable (or partially separable problem): i.e., a function can be expressed as a product or sum of sub-functions, where each of them depends on only one independent variable (or a fewer number of variables).

It is important to note that the potential modifications of the models can introduce some coding errors. These errors must be quickly debugged and fixed. For reducing an error searching area, a comfortable optimization tool is necessary not only to automate the coupling process between the model and the chosen optimization algorithm, but also to avoid additional errors during the coupling process.

Research Objective: The objective of the thesis is to explore, to develop and to apply appropriate optimization algorithms to solve the research problems defined above.

Research Strategy: This objective has to be achieved by contributing in the following research challenges:

1. Based on the existing optimisation techniques, to investigate and to develop a multi-/many-objective evolutionary optimization algorithm with a parallel evaluation of objective functions in order to efficiently solve the computationally intensive problems of the active magnetic regenerator design. The developed algorithm has to take into account the particularities of the active magnetic regenerator model and the features of its optimization problem.

2. Taking into account the quantum structure of the model of magneto caloric materials and the features of its optimization problems, to develop a quantum-inspired algorithm for solving computationally intensive separable or partially separable small scale problems for modeling/reproducing physical properties of different magneto caloric materials.

3. To develop an optimization algorithm, which provides scalability w.r.t. the changes of number of objectives and number of input variables of optimization problem, because several numbers of optimization problems with different numbers of objectives and different numbers of input variables can be formulated for each model.

4. To develop a flexible and user-friendly software tool based on the developed algorithms for solving optimization problems of both simulation models, which allows users to quickly set up and launch different experiments.

5. To develop a new method based on the optimization of the model of magneto caloric materials, for automating the process of modeling/reproducing physical properties of different magneto caloric materials.

6. To explore the impact of different parameters of the active magnetic regenerator model (operating in two modes: a magnetic cooling system and a thermo-magnetic energy generator), on its performance, by applying the developed algorithms and the optimization tool.

1.3 Relevance and Importance

Environmental Relevance:

The environmental relevance of this thesis is explained by the current growing demand for refrigeration systems, and as a result, by the problems of global energy consumption, the emission of carbon dioxide and harmful gases into the atmosphere, which become more critical. All these factors lead to environmental degradation and negative impacts on human health. For this reason, the problem of limiting energy consumption and environmental protection is constantly on the agenda of international environmental organizations, like *World Wildlife Fund* or *International Energy Agency*.

Scientific Relevance:

The scientific relevance of this thesis is confirmed by the presented facts below:

1. Various aspects of the research problems of this thesis (i.e., the method for modeling/reproducing physical properties of the magneto caloric materials, the optimization algorithms for solving the simulation models problems, an optimization approach for improving the active magnetic regenerator design and etc.) are published in different scientific papers and journals, and discussed on different international scientific conferences.

2. Many state-of-the-art non-measurement methods for studying, modeling and reproducing physical properties of magneto caloric materials are presented in the scientific literature [Sokolovskiy et al. (2009)], [Sokolovskiy et al. (2010)], [Maiorino et al. (2019)], [de Castro et al. (2020)]. However, all of them have serious disadvantages: (i) a lack of generalization, which makes the experiments highly time consuming and expensive [Sokolovskiy et al. (2009)], [Sokolovskiy et al. (2010)]; (ii) a dependency on databases, which can restrict a research [Maiorino et al. (2019)], [de Castro et al. (2020)].

   In order to overcome the mentioned drawbacks, we introduce a new reliable indirect method based on the optimization, for modeling/studying/reproducing physical properties of magneto caloric materials. The relevance of this method is in its originality, which formulates the problem of modeling physical properties of magneto caloric material as an optimization problem, which allows users to generalize the process of modeling/reproducing physical properties for different magneto caloric materials. Even though the model fitting to experimental results is not new, it has never been implied with the aim to solve the problem of modeling physical properties of magneto caloric material.

3. The model of magneto caloric materials has a quantum structure and potentially, can be implemented on a quantum hardware and optimized by one of the hybrid quantum-classical algorithms, e.g., Variational Quantum Eigensolver [Peruzzo et al. (2014)] or Quantum Approximate Optimization Algorithm [Farhi et al. (2014)], [Verdon et al. (2019)]. These algorithms have already been used to solve several problems in the fields of quantum chemistry and materials [McArdle et al. (2020)], [Cao et al. (2019)]. However, an application of the hybrid quantum-classical algorithms are still limited by the complexity of quantum circuits and the complexity of classical optimization problems. This fact partially explains recently growing interest to the quantum-inspired algorithms based on the quantum physics, because they are useful for theoretically understanding different quantum features in an optimization process and do not require to take into account the features of a quantum hardware. Aiming at making a step forward the further development of the quantum-inspired optimization algorithms for solving different computationally intensive problems of the separable simulation model of magneto caloric materials, we investigate the idea to achieve a potential speedup by simulating quantum physics phenomena.

4. Applying the existing optimisation techniques, we compile them in order to produce new algorithms with necessary features for solving the computationally intensive problems based on the simulation models of the active magnetic regenerator and the model of magneto caloric materials.

5. Many different researches were made for optimizing the performance of active magnetic regenerators (AMRs) by applying single- [Teyber et al. (2018)] and multi-objective [Ganjehsarabi et al. (2016)], [Roy et al. (2017)] evolutionary algorithms. However, an AMR design with commercially relevant performance for real world applications has not been demonstrated yet. This can be explained by two issues: (i) a lack of analysis of the problems of an AMR model from an optimization point of view; (ii) a lack of flexible optimization tools with appropriate algorithms to allow users to easily set up and run different experiments: e.g., if a problem is modified so that its number of objectives is changed, to conduct a new experiment is a complicated and time-consuming task, because a new algorithm has to be selected and a joining process of the selected algorithm with the modified problems requires some additional time and can introduce extra errors [Deb & Tiwari (2008)]. Because of the second issue, the works aiming at optimizing the AMR performance focus on only a single problem for the selected application mode. Thus, they do not allow to explore innovative architectures of AMR, which can operate in different application modes.

   In contrast to the existing optimization-based works, we take into account both issues and present a user-friendly optimization tool, which allows users not only to easily set up different optimization experiments independently from any modifications of the model and its problems, but also to investigate the AMR design for the two operating modes: a magnetic cooling system and a thermo-magnetic energy generator.

Importance:
The scientific and practical importance of this thesis is confirmed by the following arguments:

1. The theory and practice of magnetic material science can obtain a further progress by applying the proposed new method for modeling/reproducing physical properties of magnetic materials.

2. The magnetic cooling industry (Ubiblue) will have a convenient powerful software tool, capable to solve computationally intensive optimization problems w.r.t. different sizes of the search spaces and objective spaces.

3. A further progress in the development of quantum-inspired algorithms can be achieved by applying the results of this thesis.

4. The solution of the presented research problems has a direct impact on the development of material science, magnetic cooling technology, and has an indirect impact on the improvement of global environmental problems.

1.4 Main Contributions

According to the defined research strategy, we present the contributions, which were made for solving the research problems of this thesis.

An Archive-based Multi-objective Optimization Algorithm (FastEMO)

We present a multi-/many-objective evolutionary algorithm for computationally intensive problems, which is scalable w.r.t. the population size: i.e., it reduces the overall execution time by working with a large population size in a small number of generations, where the simulation models are run in parallel. Moreover, the large population size can be efficient for solving large-scale problems, in which the simulation model of an active magnetic regenerator can be formulated.

In order to ensure a low complexity, the proposed algorithm, called FastEMO, is derived from the state-of-the-art Archive-based Stochastic Ranking Evolutionary Algorithm (ASREA) [Sharma & Collet (2010*a*)] and presented in Section 4. FastEMO improves the accuracy of ASREA, while keeping its low algorithmic complexity. A key contribution compared to the ASREA algorithm is an application of a technique of control the dominance area of solutions [Sato et al. (2007)] instead of the conventional Pareto dominance for improving efficiency of the algorithm on many-objective problems.

The FastEMO algorithm is benchmarked on the Deb-Thiele-Laumanns-Zitzler [Deb et al. (2002)], Walking Fish Group [Huband et al. (2005)] and Black-Box Optimization Benchmarking (BBOB) [Hansen et al. (2012)] multi and many-objective test suites. The expected main limitation of FastEMO is the required large population size, which prevents the algorithm application in the cases, when the simulation model cannot be run in parallel.

Quantum-inspired Algorithm with Evolution Strategy (QAES)

In order to deal with single-objective computationally intensive separable and/or partially-separable optimization problems, such as the model of magneto caloric materials, the Quantum-inspired Algorithm with Evolution Strategy (QAES) is derived from the Diffusion quantum Monte Carlo method [Kosztin et al. (1996)] and presented in Section 5.1.

We investigated the efficiency of the use of the Diffusion Monte Carlo (DMC) method for an evolutionary optimization process, which manifests as regulating the population size of the algorithm according to the diffusion process. This feature of the DMC method can be useful for computationally intensive problems, because it allows the algorithm to start with a very small population size and to automatically increase/decrease this size, and thereby to reduce the overall computational efforts.

In quantum-inspired optimization algorithms based on quantum physics phenomena, like the DMC method, the quantum processes are not just emulated on a classical hardware, they are involved in the optimization algorithm in order to improve the performance over conventional optimization algorithms. Thus, in order to adapt the DMC method to the optimization process, we apply the (1+1)–Evolution Strategy [Hansen et al. (2015)], which was selected because of its simplicity.

To the best of our knowledge, the Quantum Diffusion Monte Carlo method has not been adapted to the optimization process earlier.

The proposed algorithm is benchmarked on the BBOB test suites against two classical algorithms (BFGS [Ros (2009)] and BIPOP-CMAES [Hansen (2009)]) and the Quantum-behaved Particle Swarm Optimization Algorithm (QPSO) [Sun, Feng & Xu (2004)]. It was shown that QAES can find the global minimum with a smaller search cost than the QPSO algorithm.

Fusion-based Unified Optimization Algorithm: QIU-NSA

In order to ensure the property of scalability w.r.t. the number of objectives and the number of decision variables, we produce a unified fusion-based algorithm, called Quantum-Inspired Unified Non-dominated Sorting Algorithm (QIU-NSA). The term "unified" is borrowed from [Seada & Deb (2014)], which corresponds to algorithms capable to solve from single- to many-objective problems. The description of QIU-NSA is presented in Section 5.2.

The proposed algorithm is based on a fusion method, which allows us to combine the solutions from several algorithms. In general, the fusion method is applied to reduce the challenge to choose an appropriate algorithm to solve complex problems [Ibrahim, Martin, Rahnamayan & Deb (2017)] and it fuses the solutions from only single- or only multi-objective optimization algorithms. To the best of our knowledge, the fusion method has not been applied earlier for ensuring the scalability w.r.t. the number of objectives. In this regard, this is a new contribution.

In this work, the many-objective algorithm, Non-dominated Sorting Genetic Algorithm III (NSGA-III) [Deb & Jain (2013)], is selected as a baseline algorithm, because its efficiency on different many-objective problems (which are having more than three objectives). However, the NSGA-III algorithm is not suitable for solving single-objective problems [Seada & Deb (2014)]. The proposed algorithm improves the NSGA-III ability to solve single-objective problems by fusing the solutions from NSGA-III with the solutions of the Quantum-behaved Particle Swarm Optimization Algorithm (QPSO) [Sun, Feng & Xu (2004)]. According our analysis, in different works based on the fusion method, the algorithm PSO is applied, due to its explorative power [Ibrahim, Martin, Rahnamayan & Deb (2017)]. Aiming at investigating the applicability of a quantum-inspired algorithm as a part of the fusion method, we select quantum version of QPSO. Technically, the proposed QPSO version maintains the population diversity, which improves the performance of the algorithm on single-objective problems without any loss of efficiency on many-objective functions, because of the NSGA-III structure is not changed.

The proposed algorithm is experimentally validated on different BBOB test suites, where the improved results over the original NSGA-III on single-objective problems is confirmed. Moreover, the proposed algorithm shows the high efficiency on solving the large scale separable problems and outperforms such reference algorithms as the separable CMA-ES algorithm [Hansen & Ostermeier (2001), Hansen et al. (2003)] and the Limited Memory CMA-ES algorithm [Loshchilov (2014)].

Optimization Tool

A new software optimization tool for solving different problems of the simulation models used in this thesis, has been developed within the EASEA (*EAsy Specification of Evolutionary Algorithms*) platform and is presented in Section 6.1. The proposed algorithms, as well as several state-of-the-art algorithms, have been included into the EASEA platform as its templates. The concept of the templates provides a flexible and user-friendly interface for easily joining the codes of a given problem and the selected algorithm, setting up different experiments and comparing the results, as well as it ensures the accessible and transparent source codes of the optimization algorithms. Another important feature is the EASEA Application Programming Interface (API), which makes the proposed tool invariant w.r.t. the programming language of models.

Novel Evolutionary Method to Study Physical Properties of MCMs

In order to overcome the drawbacks of the existing non-measurement methods for modeling, reproducing and studying physical properties of various Magneto Caloric Materials (MCMs), we propose the new method presented in Section 6.2, which is based on evolutionary optimization.

The key contribution is to apply optimization for solving the computationally intensive problems based on the Hamiltonian model of MCMs provided by the *Crismat* laboratory, in order to find the combination of parameters of this model, which corresponds the physical properties of a required/studied material. These parameters of the Hamiltonian model, further called the free parameters, are required in order to reproduce the physical properties needed for the design of Active Magnetic Regenerator (AMR): the temperature dependencies of magnetization and the temperature dependencies of heat capacity. In fact, this method can be considered as a method of model fitting to experimental/desirable results, however it has never been implied earlier for solving the problem of reproducing/studying physical properties of MCMs. Moreover, we analyse the specific features of the optimization problems of the Hamiltonian model and take them into account in order to make this method the most efficient.

Mainly, this method aims at simplifying, accelerating and generalising the process of reproducing the different physical properties of different MCMs so that the scientists can easily study them and use in the AMR models.

The applicability of the proposed method is validated by experiments on three alloys from the following families: $LaFe_{13-x}Si_x$, $GdSiGe$ and $LaFeCoSi$. It is shown that the results, obtained by the proposed method for these three alloys, are in good qualitative agreement with the available experimental data.

Due to the fact that this method can reproduce the physical properties of different MCMs, it can be directly applied for creating a new database of materials in the same way as it is presented through validation experiments.

Design Exploration of the Dual-Mode Operating AMR model

In order to find an innovative architecture of the Active Magnetic Regenerator (AMR) model, which can operates in the two application modes: i.e., a Magnetic Refrigeration System (MRS) and a Thermo-Magnetic energy Generators (TMG), we investigate the impact of different combinations of the control and design parameters of this model on its performance. This study is presented in Section 6.3.

For this purpose, we apply the developed multi-objective algorithm, FastEMO, through the EASEA-based tool. The EASEA Application Programming Interface (API) allows users to launch the Scilab code of the AMR model developed by the *Ubiblue* company. The presented study case is centred on four optimization objectives: the energy efficiency and the power density in both modes that allows users to obtain detailed information about the relationship between the values of the control and design parameters of the AMR model and its performance.

The main conclusion is that common optimal parameters can be found for a commercially applicable design in both modes (the MRS and the TMG).

1.5 Thesis Outline

After this introduction, the monograph starts by a background review (Part II) of the state-of-the-art aspects in each domain of this thesis.

Chapter 2 presents a brief overview of the state-of-the-art methods for studying, modeling and reproducing physical properties of different magneto caloric materials and for optimizing the active magnetic regenerator design. Also, this chapter presents the description of the theoretical simulation model of magneto caloric materials, provided by the *Crismat* laboratory and the multi-physical and multi-scales numerical simulation model of the active magnetic regenerator, provided by the *Ubiblue* company.

In Chapter 3, we review the state-of-the-art continuous optimization algorithms used in this thesis and the theoretical points related to the methods for solving computationally intensive problems. Then we present the evolutionary algorithm specification platform, called EASEA, which serves as a software support for this thesis.

The next part concerns the presented contributions (Part III), which is divided in three chapters. The first two chapters (Chapter 4 and Chapter 5) are devoted to the first three research challenges, while the last three challenges of the defined research strategy are presented in Chapter 6.

Chapter 4 describes an archive-based multi-objective algorithm, FastEMO, with its experimental validations on different test suites, focusing on the questions of runtime and computational complexity.

Chapter 5 presents the quantum-inspired algorithm based on the Diffusion Monte Carlo method (QAES) and the unified algorithm based on a fusion method (QIU-NSA). The efficiency of QAES for computationally intensive problems is studied, as well as the scalability of QIU-NSA w.r.t. the number of objectives and w.r.t. the number of decision variables are explored. Both algorithms are experimentally validated on Black-Box Optimization Benchmarking (BBOB) test suites.

Chapter 6, first, provides the optimization tool based on the updated version of the EASEA platform. Then, a method for modeling and reproducing physical properties of different magneto caloric materials is introduced and validated on three different alloys. Last, the configuration design of the dual-mode operating model of the active magnetic regenerator is explored.

Finally, Chapter 7 summarizes the fundamental contributions of this thesis and offers future perspectives.

Part II

Background Review

Chapter 2

Simulation Models

In this chapter, we present a theoretical background, which is required for further understanding the research problems of the thesis.

First, in Section 2.1, we give a brief explanation of the Magneto Caloric Effect (MCE), which is the main working principle of magnetic devices. Then, in Section 2.2, we review the state-of-the-art non-measurement methods for modeling and studying the physical properties of Magneto Caloric Materials (MCMs). We discuss their main drawbacks that motivate us to develop a new method based on the optimization of the simulation model of MCMs for reproducing the physical properties of these materials. Then, we describe the theoretical phenomenological simulation model of MCMs provided by the *Crismat* laboratory, which is used as a part of optimization function(s) in the method proposed in this thesis. Next, in Section 2.3, we give the definition of an Active Magnetic Regenerator (AMR) and present the recent trends in the optimization of models of AMR. Finally, we give a brief presentation of the complex multi-physical and multi-scales numerical simulation model of AMR provided by the *Ubiblue* company, which is employed in the present thesis.

2.1 Basic Theory

2.1.1 Magneto Caloric Effect (MCE)

The **Magneto Caloric Effect (MCE)** is a magneto-thermodynamic phenomenon, which displays itself in the emission/absorption of heat by a magnetic material under applying/removing an external magnetic field.

A **Magneto Caloric Material (MCM)** is a material in which the MCE occurs. The cooling capacity of a Magnetic Refrigeration System (MRS) depends on the quantity and on the properties of the used MCM.

The schematic diagram of the MCE is presented in Figure 2.1: an increase or decrease in the strength of an external magnetic field modifies the ordering of the magnetic moments of the atoms that form the MCM. When the magnetic ordering coincides with a structural change affected by the field, additional heat is released or absorbed, thus strongly enhancing the MCE that can be used for refrigerating.

A larger MCE in the refrigerant, i.e. MCM, results in a greater variation of internal energy and provides more cooling needed for a Magnetic Refrigeration System (MRS).

The main parameters that define the MCE are:

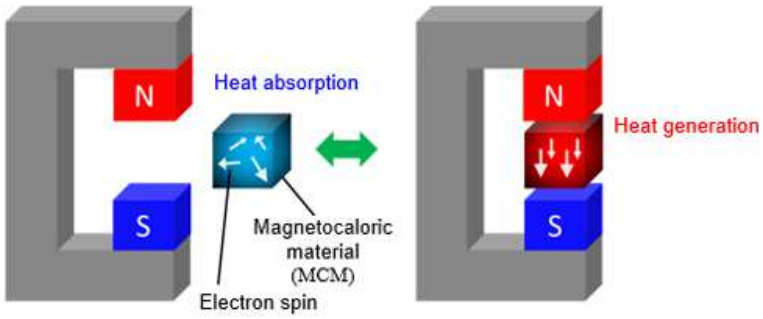


Figure 2.1: Schematic diagram of the MCE. The illustration is adapted from the site of Fujikura Ltd. (`https://www.fujikura.co.jp`).

1. **The Curie temperature (T_c):**
   T_c is the temperature of the critical point at which a material's intrinsic magnetic moments change direction (ordering/disordering). A MCM exhibits its maximum MCE at the material's Curie temperature.

2. **The magnitude of the MCE:**
   The magnitude of the MCE is quantified by two following parameters:

   (a) **Magnetic entropy change (ΔS_m):**
   ΔS_m is the reversible change in entropy produced during magnetization or demagnetization of the material under isothermal conditions (i.e., the temperature of material remains constant). It is dependent on the temperature of the material (T) and the magnitude of the magnetic field (H).

   The total entropy (S) of any MCM consists of the sum of the lattice entropy (S_{lat}), the electronic entropy (S_{el}), and the magnetic entropy (S_m) [Vasile et al. (2008)]:

   $$S(H,T) = S_m(H,T) + S_{el}(T) + S_{lat}(T) \tag{2.1}$$

   The lattice and electronic entropy can be considered independent from the magnetic field and only depend on temperature. However, the magnetic entropy is highly dependent on both the magnetic field and the temperature [Vasile et al. (2008)].

   (b) **Adiabatic temperature change (ΔT_{ad}):**
   ΔT_{ad} is the temperature change produced during magnetization or demagnetization of the material under adiabatic conditions (i.e., the entropy remains constant). It is also dependent on the temperature of the material (T) and the magnitude of the magnetic field (H).

In order to explain the thermodynamics of the MCE, in Figure 2.2, a representative entropy-temperature diagram demonstrates the magnetic field and temperature dependence of a MCM, where the total entropy is displayed with an applied external field (H_1, where $H_1 > H_0$), and without magnetic field ($H_0 = 0$). The magnetic part of the total entropy is also shown as a dash line in both cases (H_0 and H_1).

In this diagram, two relevant processes are shown, where the magnetic field is applied under two conditions (adiabatic and isothermal [Vasile et al. (2008)]):

1. **When the magnetic field (H) is applied adiabatically**, the magnetic field aligns the magnetic moments of the atoms of the MCM and decreasing the magnetic entropy (S_m), but as the total entropy does not change: $S(T_0, H_0) = S(T_1, H_1)$. In order to compensate the decrease in the magnetic contribution, the lattice entropy increases. For this reason, the application of a magnetic field results in a heating of the material.

   The adiabatic temperature rises and ΔT_{ad} can be illustrated as the difference between the corresponding $S(T, H)$ functions and it is a measurement of the MCE in the studied MCM: $\Delta T_{ad} = T_1 - T_0$ (see the horizontal arrow in Figure 2.2).

   When the applied magnetic field (H) is removed (an adiabatic demagnetization): the total entropy (S) remains constant and the temperature decreases in order to compensate for the magnetic entropy, which increases: the magnetic spin system returns to its original alignment together with the temperature.

2. **When the magnetic field (H) is applied isothermally**, the total entropy decreases due to the decrease in its magnetic contribution: $\Delta S_m = S(T_0, H_0) - S(T_0, H_1)$, since the lattice and electronic entropy do not vary as a result of keeping the temperature constant. The respective entropy change ΔS_m is shown as the vertical arrow in Figure 2.2).

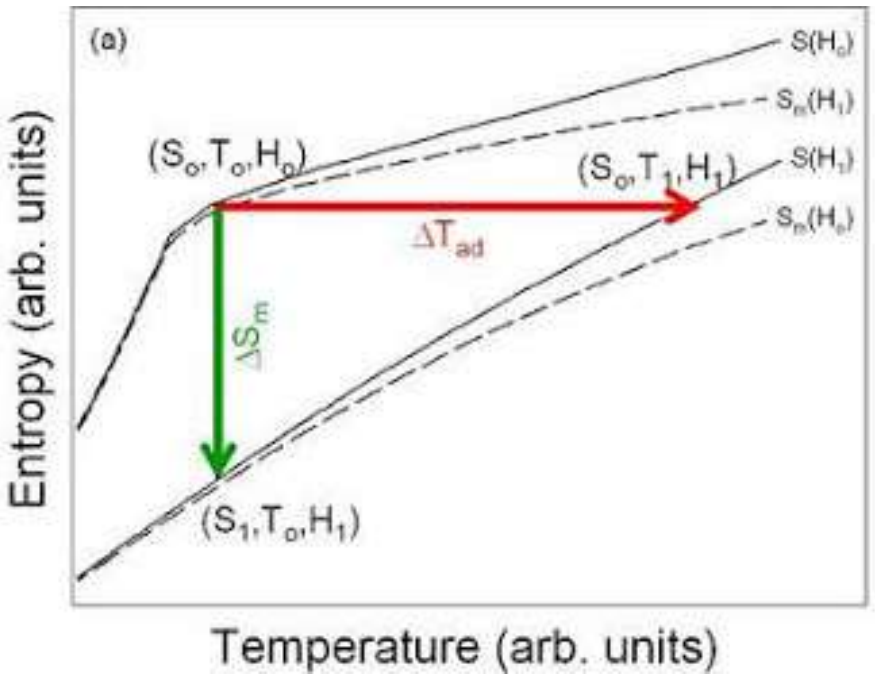


Figure 2.2: Representation of the MCE: $S-T$ diagram. The total entropy (S) is the solid lines and the magnetic entropy (S_m) is the dashed lines. ΔT_{ad} and ΔS_m are indicated by the horizontal and vertical arrows, respectively. The illustration is adapted from the site of the Department of Physics of University of South Florida (`http://labs.cas.usf.edu`).

2.1.2 First and Second Order Phase Transitions

Near the Curie temperature, the magnitude of ΔT_{ad} and ΔS_m is strongly dependent on the temperature of the material. The Curie temperature is also called the transition temperature, where the transition is the transformation of a MCM from one phase to another (ordered phase/disordered phase). The main characteristic of the transition is an abrupt change in one or more properties of a MCM.

With regard to the phase transition, there are two groups of MCMs divided by the nature of the magnetic phase transition that occurs at the Curie temperature: (i) First Order Transition (FOT) and (ii) Second Order Transition (SOT) [Chaudhary et al. (2019)], [Gómez et al. (2013)], [Gombi & Sahu (2020)], [Lyubina (2017)].

In order to present the distinction between the materials with different phase transition, in Figure 2.3, we show the schematic representation of the magnetization and total entropy of materials with SOT (a, b) and FOT (c, d) with respect to temperature and magnetic field.

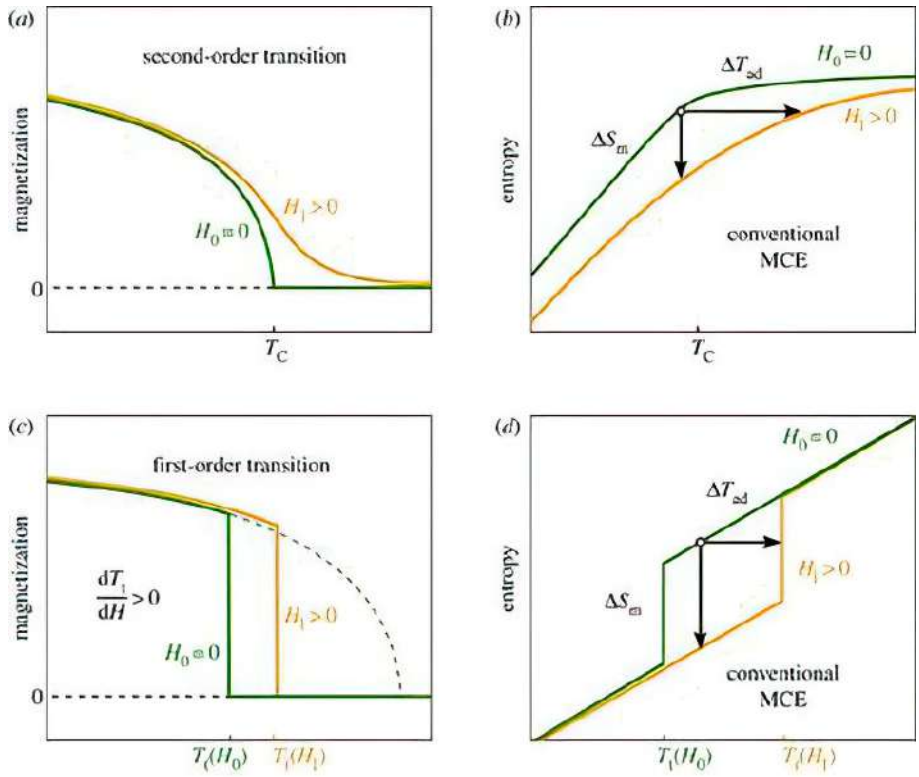


Figure 2.3: Schematic of the temperature-dependence of magnetization and the total entropy with ($H_1 > 0$) and without ($H_0 = 0$) a magnetic field of a second-order transition in (a) and (b), of a first-order transition in (c) and (d). The illustration is adapted from [Gutfleisch et al. (2016)].

Second Order magnetic phase Transition (SOT)

In Figure 2.3 (a), the green curve represents the magnetization ($M(T)$) change in zero magnetic field (H_0): around the Curie temperature (T_c), the magnetization vanishes and the ferromagnet turns into a paramagnet. In the magnetic field (H_1), a certain

magnetization is also observed in the paramagnetic phase due to a partial alignment of the spins (see yellow curve in Figure 2.3 (a)) [Gutfleisch et al. (2016)].

The temperature dependence of the total entropy with and without magnetic field is schematically shown as a yellow and green curve respectively in Figure 2.3 (b). By applying a magnetic field, the magnetic moments align to some extent. As explained in Section 2.1.1, the decrease in entropy is observed under isothermal conditions. Under adiabatic conditions, in order to compensate for the decrease in the magnetic contribution, the lattice entropy increases and for this reason, the application of a magnetic field results in a heating of the material by increasing ΔT_{ad}.

From Figure 2.3 (a, b) we can observe that the materials with SOT show a continuity in the curves of the magnetization and the total entropy. They have have no latent heat at the transition, no hysteresis, no crystalline lattice change and the MCE is almost instantaneous, in the order of microseconds.

First Order magnetic phase Transition (FOT)

In contrast with the Second Order magnetic phase Transition (SOT), materials with the First Order magnetic phase Transition (FOT) experience a simultaneous ordering of magnetic moments and a crystalline structure change associated with the transition. That is why the materials with FOT commonly show a giant magnitude of the MCE: the observed giant MCE in the FOT materials is the sum of the magnetic entropy change and the difference in the entropy of the two crystallographic modifications (structural entropy change). The latter contribution accounts for the larger entropy change of FOT respect to the SOT materials.

From Figure 2.3 (c, d) we can observe that in contrast with SOT, the FOT materials show a discontinuity in magnetization and total entropy curves, due to the transformation between two phases with equal thermodynamic potential. This discontinuity is related to the latent heat [Gutfleisch et al. (2016)].

The application of a magnetic field (yellow curves in Figure 2.3 (c, d)) results in a shift of the transition temperature T_t, which can be understood as the driving force of the MCE in such materials. This is happening because the magnetic field stabilizes the phase with a higher magnetization, being in the low temperature phase [Gutfleisch et al. (2016)]. Due to this shift of the transition temperature in magnetic fields, the entropy change diagram has the shape of a parallelogram (green and yellow curve in Figure 2.3 (d)).

In spite of the giant MCE, such kind of MCMs have a significant source of efficiency losses of the cooling cycle: i.e., thermal and magnetic hysteresis [Mozharivskyj (2016)], [Gómez et al. (2013)], [Hess et al. (2020)]. Also, as the FOT materials experience a change in structure, atoms are displaced during the change in crystal structure and therefore, the required time can be larger, with a magnitude in the order of seconds. This could be a problem because magnetic refrigerators operate between 0.5 and 10 Hz and much of the giant MCE may not be used during the rapid increase and decrease of the magnetic field.

2.1.3 Measurement of the MCE

The magnitude of the MCE of a material can be estimated by measurement and non-measurement methods.

In order to better explain the relation between the MCE and the physical properties of MCMs: i.e., the temperature and applied magnetic field dependency of magnetization ($M(T,H)$) and the temperature and applied magnetic field dependency of heat capacity ($Cv(T,H)$), we present the measurement methods.

The measurement methods are divided into two groups: direct and indirect.

Direct Measurement:

The direct measurement techniques of the MCE (in terms of ΔT_{ad}) involve the measurement of the initial (T_0) and final (T_F) temperatures of the MCM, when the external magnetic field is changed from an initial (H_0) to a final value (H_F) [Vasile et al. (2008)]. Thus, the value of the adiabatic temperature change is defined by Equation 2.2:

$$\Delta T_{ad}(T_0, H_F - H_0) = T_F - T_0 \tag{2.2}$$

However, the direct measurement method has some limitations, because it requires special adiabatic conditions with a good thermal contact between the sample of MCM and the thermal sensor, which is quite challenging to provide [Blázquez et al. (2017)].

Indirect Measurement:

Unlike direct measurements, which usually only yield an adiabatic temperature change, indirect experiments allow the calculation of both values: the adiabatic temperature change (ΔT_{ad}) and the magnetic entropy change (ΔS_m) in the case of heat capacity (Cv) measurements, or just (ΔS_m) in the case of magnetisation (M) measurements.

For materials with SOT, the relation between magnetic field (H), the magnetisation (M) of the material, and the temperature (T), to the MCE value in terms of ΔS_m is presented in Equation 2.3, which is derived from one of the Maxwell relations [Morrish (2001)]:

$$\Delta S_m(T, \Delta H) = \int_{H_1}^{H2} \left(\frac{\partial M(H,T)}{\partial T} \right) dH \tag{2.3}$$

The accuracy of ΔS_m value calculated from magnetisation data by Equation 2.3, depends on the accuracy of the measurements of the magnetic moment, T and H.

The value of the MCE in terms of ΔT_{ad} is presented in Equation 2.4:

$$\Delta T_{ad}(T, \Delta H) = -\int_{H_1}^{H2} \left(\frac{T}{Cv(T,H)} \right) \left(\frac{\partial M(H,T)}{\partial T} \right) dH \tag{2.4}$$

For materials with FOT, the the isothermal magnetic entropy change (ΔS_m) can be evaluated with the following equation based on the Clasius-Clapeyron equation:

$$\Delta S_m(T, \Delta H) = \left(\frac{dH}{dT_c} \right) \Delta M \tag{2.5}$$

The adiabatic temperature change (ΔT_{ad}) of FOT materials can be evaluated by Equation 2.6:

$$\Delta T_{ad}(T, \Delta H) = -\left(\frac{T}{Cv(T,H)} \right) \left(\frac{dH}{dT_c} \right) \Delta M \tag{2.6}$$

From Equations 2.3- 2.6 we can observe that the MCE of a MCM can be derived from magnetization measurements as a function of the temperature and the applied magnetic field ($M(T, H)$) and heat capacity measurements as a function of the temperature and the applied magnetic field ($Cv(T, H)$) [Vasile et al. (2008)]. It explains the importance of the accuracy of these physical properties (magnetization and heat capacity) of MCMs for the magnitude of the MCE.

One more important detail is that at least two physical quantities (magnetization and heat capacity) are required for the simulation of magnetic cooling systems. Consequently, two measurements by different devices (a magnetometer and an adiabatic calorimeter respectively) are needed, which is quite complicated [Blázquez et al. (2017)].

To summarize, the measurement methods are time-consuming and could be challenging to perform, especially in the case of direct measurements. This fact is an issue for modelling the Active Magnetic Regenerator (AMR) [Maiorino et al. (2019)], because it requires a high data density.

Thus, a non-measurement method for reproducing/studying the physical properties of MCMs can be useful to overcome the difficulties of measurement methods. Moreover, such kind of reliable method is required in order to guarantee the thermodynamic consistency and the perfect energy conservation in the model of AMR [Maiorino et al. (2019)].

2.2 Model of Magneto Caloric Materials (MCMs)

First, in Section 2.2.1, we discuss the state-of-the-art in non-measurement methods and their pros and cons, which motivate us to provide a new method for modeling the physical properties of Magneto Caloric Materials (MCMs), i.e., magnetization and heat capacity, based on the optimization of the simulation model of MCMs. Then, in Section 2.2.2, we present a description of the model of MCMs provided by the *Crismat* laboratory, which is used as a basis of the method proposed in this thesis.

2.2.1 State-of-the-art Methods for Reproducing Physical Properties of MCMs

Despite promising recent researches, magnetic devices development still faces several obstacles related with MCMs that prevent its commercial production. First, the earth compounds of MCMs are rare and costly [Gombi & Sahu (2020)]. Second, the MCMs temperature change produced by their MCE, is often too small, which leads to insufficient operating ranges for commercial applications [Gschneidner Jr & Pecharsky (2006)], [Aprea et al. (2015)].

In order to discover a suitable MCM for its potential application in commercial magnetic device, it is necessary to investigate and reproduce the physical properties that control the magnitude of the MCE: i.e., magnetization and heat capacity. It is a matter of a non-measurement method, that allows scientists to automate the process of reproducing the temperature dependence of magnetization and the temperature dependence of heat capacity for materials with First Order magnetic phase Transition (FOT) and Second Order magnetic phase Transition (SOT).

The non-measurement methods, like the measurement ones, aim at evaluating the adiabatic temperature change (ΔT_{ad}) of MCMs. But instead of the direct or indirect

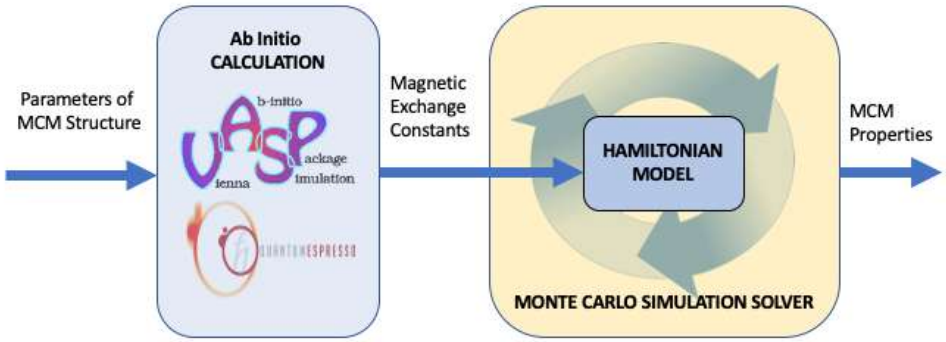


Figure 2.4: Combination of *ab initio* and Monte Carlo simulation.

measurements, the non-measurement methods are reproducing, i.e. modeling or predicting, the physical properties of MCMs: the temperature and the magnetic field dependency of magnetization and of heat capacity.

The non-measurement methods for reproducing physical properties of MCMs can be roughly divided into 2 groups:

1. **Based on the theoretical models:** They also can be divided into two groups:

   (a) **Based on the Mean Field Theory (MFT):** the Wiess MFT is the most used theoretical model to describe the entropy state of SOT materials in the Magnetic Refrigeration System (MRS). MFT is performed by the calculation of the value of magnetization as a function of the applied magnetic field, the temperature and the exchange mean field [Morrish (2001)], [Amaral et al. (2011)]. The main disadvantage of the MFT model is that it over-estimates the MCE [Benford & Brown (1981)]. Furthermore, it has difficulties in estimating the magnetic entropy for FOT materials Amaral et al. (2011).

   (b) **Based on combination of *ab initio* calculations and classical Monte Carlo simulation:** this method is shown in Figure 2.4. In order to theoretically investigate the temperature dependence of the physical properties of MCMs, a finite-temperature Monte Carlo simulation is applied to a theoretical model of Hamiltonian, for which the exchange coupling constants, magnetic moments, and anisotropy energy are taken from *ab initio* (first-principles) calculations [Sokolovskiy et al. (2009)], [Sokolovskiy et al. (2010)]. In the research presented in [Sokolovskiy et al. (2010)], the Monte Carlo simulation model is applied to theoretically investigate the MCE at the coupled magneto-structural transition in Heusler NiMnIn compound.

   The strong point of this approach is the thermodynamic coherence of obtained data, due to the fundamental Hamiltonian model. But, as seen, this approach requires some *ab initio* calculations of magnetic exchange constants of Hamiltonian. Thereby, the main disadvantage of this method is a lack of generalization: the Hamiltonian model depends on the approximately calculated constants, that reduce the scope of the study process to one chosen MCM. Moreover, *ab initio* calculations is somewhat challenging, because they need a lot of time

and an application of special program-packages for *ab initio* calculations and materials modeling, like *VASP*[1]or *Quantum Espresso*[2].

2. **Based on artificial intelligence:** machine-learning algorithms were applied in several recent researches of the MCE [Maiorino et al. (2019)], [de Castro et al. (2020)]. One collects the experimental data as a training set and applies an artificial neural network for time-efficient and accurate prediction of the magnitude of the MCE (ΔT_{ad}) for any magnetic fields and temperatures, ensuring a high density of the data needed for accurate numerical simulations of Active Magnetic Regenerator (AMR) [Maiorino et al. (2019)]. In [de Castro et al. (2020)], a gradient boosted tree algorithm is trained on the datasets from the literature [Gschneidner Jr et al. (2005)], [Franco et al. (2018)], and used to find the large MCE for a MCM near $T_c = 20K$. The machine-learning approach has some strong points: it can be generalized to different MCMs and it can be easily integrated within a simulation model of AMR. But this approach has a serious limitation due to the fact that it cannot guarantee the thermodynamic coherence of the obtained data, which requires reference to a more fundamental model. Moreover, it has a strong dependence on experimental database availability.

To summarize, the presented brief analysis of the methods for reproducing the physical properties of MCMs showed that each group of methods has its disadvantages: (i) the model-based methods are time-consuming, challenging to perform, require an active human participation in the modeling process and they have a lack of generalization; (ii) the artificial intelligence-based methods have strong dependency on the qualities and density of the databases $(M(T,H), Cv(T,H))$ for different families of MCMs. Moreover, both groups of methods have some problems with the accuracy of the MCE magnitude.

In the light of these disadvantages, the question of using an optimization algorithm as basis for a new more efficient method is raised. Furthermore, in the future, the optimization problem of the MCMs is considered as a sub-problem of optimization of the AMR model. In such a configuration, the state-of-the-art non-measurement methods cannot be considered as candidates.

The idea of our method is to use the theoretical model of Hamiltonian with the Monte Carlo solver as a part of the optimization function, due to its good thermodynamic coherence and to improve it by an optimization process. The goal of the improvement is to generalize this method and to automate it (i.e., to exclude an active human participation).

This goal can be achieved by applying an appropriate optimization algorithm for computationally intensive problems to the theoretical model of MCMs solved by the Monte Carlo method. Instead of the parameters evaluated by *ab initio* calculations, we use the free parameters introduced in the theoretical Hamiltonian model. In our method, we optimize these free parameters by minimizing the differences between the required/desirable and current post-processing parameters evaluated from the outputs of the theoretical Hamiltonian model, where the outputs are the simulated physical properties of MCMs $(M(T,H), Cv(T,H))$.

In the proposed method presented in Section 6.2, *ab initio* calculations are not required, since the combination of the free parameters will be carried out by the optimiza-

[1] `https://www.vasp.at`

[2] `https://www.quantum-espresso.org`

tion algorithm. Since the proposed method is automatic and does not require human participation, it can be included in the AMR model.

2.2.2 Theoretical MCMs Model

In this section, we present the theoretical model of Magneto Caloric Materials (MCMs) used inside our non-measurement method for reproducing physical properties of MCMs, aiming at evaluating the magnitude of the Magneto Caloric Effect (MCE).

This model is a theoretical phenomenological generalized Blume–Emery–Griffiths–Ising Hamiltonian model with free parameters, referred to as BEG-I, which allows us to model the physical properties of materials with First Order magnetic phase Transition (FOT) and Second Order magnetic phase Transition (SOT).

This model is developed and provided as a *C++*-based code by the *Crismat* laboratory.

We give an approximate description of the BEG-I model below, because an official representation of the model has not been published yet.

2.2.2.1 Blume–Emery–Griffiths–Ising (BEG-I) Hamiltonian Model

Since the official paper has not yet been published, the current description of the theoretical model of Hamiltonian is mainly based on the similar three-component Hamiltonian model presented in [Sokolovskiy et al. (2009)].

A Hamiltonian is an operator corresponding to the total energy of the system, which is used to take into account the exchange magnetic interaction and the crystalline deformation in a Magneto Caloric Material (MCM).

The Blume–Emery–Griffiths–Ising (BEG-I) model consists of two parts: the theoretical Hamiltonian model and a Monte-Carlo simulation solver, as shown in the right part of Figure 2.4. But comparatively with *ab initio*-based models, the BEG-I model is generalized, because it works with the free parameters of the Hamiltonian instead of the constants obtained by *ab initio* calculations.

In order to reproduce the curves of magnetization and heat capacity, and evaluate the magnitude of the Magneto Caloric Effect (MCE), while taking into account the order of transition (FOT/SOT) and hysteresis, a Hamiltonian model with several degrees of freedom is required, which includes different thermodynamic contributions from the crystal lattice and its impact on the magnetic behaviour in the material.

Thus, as seen from the name of the model, the Hamiltonian of the Blume–Emery–Griffiths–Ising (BEG-I) model can be represented by two interacting contributions: the first one describes the magnetic interactions by the Ising model [Kotze (2008)] and the second one takes care of the structural transformations, described by the degenerated Blume–Emery–Griffiths model [Vives et al. (1996)].

In the model, the 3-dimensional cubic lattice has $N = L^3$ number of sites, where $L = 14$ is the lateral size of the lattice. In order to take into account the magnetic and the crystalline contributions, two degrees of freedom are associated with each site: the spin and constraints, respectively.

The magnetic contribution is described by the Ising model using spin variables $S_i = \pm 1$ defined on the cubic lattice site $(i = 1, ..., N)$.

The crystal lattice contribution is described by 2-states of constraints variables $\sigma_i = \pm 1$ defined on the cubic lattice site $(i = 1, ..., N)$. It is considered to be a second degree of

freedom to introduce the effect of crystal entropy on the magnetic transition, which is physically described by vibrations and deformations of the crystal lattice.

The generalized Hamiltonian of the BEG-I model consists of three parts (Equation 2.7): the magnetic H_m (Equation 2.8), elastic H_e (Equation 2.9) and magneto-elastic H_{me} (Equation 2.10) components:

$$H = H_m + H_e + H_{me} \tag{2.7}$$

where:

$$H_m = -J_{exc} \sum_{\langle i,j \rangle}^{N,N} J_{ij} S_i S_j - g\mu_B H_{field} \sum_{i}^{N} S_i \tag{2.8}$$

$$H_e = -J \sum_{\langle i,j \rangle}^{N,N} \sigma_i \sigma_j - K \sum_{\langle i,j \rangle}^{N,N} \sigma_i^2 \sigma_j^2 + A_{temp} \sum_{i}^{N} \sigma_i^2 \tag{2.9}$$

$$H_{me} = -U_1 \sum_{\langle i,j \rangle}^{N,N} \sigma_i \sigma_j S_i S_j + \frac{U_2}{2} \sum_{\langle i,j \rangle}^{N,N} S_i S_j \tag{2.10}$$

Where J_{exc} is the coefficient of magnetic exchange energy of the nearest neighbors; J is the coefficient of the elastic exchange interaction of the nearest neighbors; μ_B is Bohr's magneton constant; $S_i = \pm 1$ is the spin variable; H_{field} is the external magnetic field; g is the Lande factor; variable $\sigma_i = \pm 1$ represents the deformation state on each site of the lattice. $K-$ is a stiffness factor; T_c is the Curie temperature; k_B is the Boltzmann constant; p is the degeneracy factor; U_1 and U_2 are the magneto-elastic interaction coefficients; $A_{temp} = k_B T_c \ln(p)$ is a magnetic entropy stabilization at high temperature; $\sum_{\langle i,j \rangle}^{N,N}$ is the sum over all the nearest-neighbor pair of spins, which means that the spin at site ij interacts with at $i(j \pm 1)$ and $(i \pm 1)j$.

The magnetic part (H_m) is presented in Equation 2.8, where the first and the second terms of the equation describes the effective magnetic interaction and the Zeeman interaction respectively.

The elastic part (H_e) is presented Equation 2.9, where the first term represents the elastic exchange interaction of the nearest neighbors, controlled by J. The second term represents the elasticity notion, controlled by the coefficient K. The third term describes a high entropy at high temperature (in our case, it is considered as an isothermal compressibility factor, which quantifies the relative change of the volume).

The magneto-elastic part, H_{me}, (see Equation 2.10) describes the work, which is associated with a force of displacement in the presence of a magnetic work and a magneto-static interaction. The first term of H_{me} represents the magneto-elastic coupling of the nearest neighbors according to the first term in the elastic (H_e) and the magnetic (H_m) part of Hamiltonian, controlled by U_1 and U_2. The second term is used to normalize and to remove a double counting of spin-spin interaction.

As seen from Equation 2.8 - 2.10, the free parameters presented in Table 2.1, are introduced in each part of Hamiltonian to account the intensity of the magneto-structural interactions.

In order to reproduce the theoretical data (the curves of magnetization and heat capacity) in good agreement with measured magnitudes by using the BEG-I model, an appropriate combination of the free parameters for each material must be found. These

free parameters control the energy barrier of paramagnetic and ferromagnetic phases and have an impact on the field dependence of the Curie temperature and on shape of the magnetization curve.

Table 2.1: Free parameters of Hamiltonian model (BEG-I).

Parameter	Hamiltonian part	Description
H_{field}	H_m	External magnetic field
U_1,U_2	H_{ms}	Magneto-elastic interaction coefficients
A_{temp}	H_s	Magnetic entropy
		stabilization at high temperature
K	H_s	Stiffness factor

To find the required combination of free parameters by hand is impossible due to the following reasons:

1. it is a very time-consuming process;
2. the model is very sensitive to changes of the values of the free parameters.

As seen from the Equation 2.8 - 2.10, depending on the number of the used parameters, the BEG-I model can be considered as a separable or partially-separable optimization problem. Consequently, an optimization algorithm is required to solve different computationally intensive separable/partially-separable problems with a small number of input decision variables (the number of free parameters).

2.2.2.2 Monte Carlo Solver of BEG-I Model

In order to obtain the MCE of MCMs, we have to compute the temperature and magnetic field dependence of magnetization and of heat capacity.

The results of the BEG-I model is described by the Hamiltonian in Equation 2.7, solved by Monte Carlo simulation described in detail in [Kotze (2008)], [Rubinstein & Kroese (2016)], which is using random numbers to estimate statistical averages.

The model lattice includes around 2800 atoms with periodic boundary conditions. The simulation with 10^5 steps is performed, using the Metropolis algorithm. As a result, the Monte Carlo solver provides the magnetization ($M(H,T)$) and the heat capacity ($Cv(H,T)$) curves as its outputs:

- $M_{H1}(T)$, $M_{H2}(T)$: arrays of magnetization values *versus* temperature under applying magnetic fields H_1, H_2 upon cooling and warming processes respectively;
- $Cv_{H1}(T)$, $Cv_{H2}(T)$: arrays of heat capacity values *versus* temperature under applying magnetic fields H_1, H_2 upon cooling and warming processes respectively.

The exchange constant J_{exc} is taken as unit of measurement of energies.

For each Monte Carlo (MC) step, the average of the magnetization per spin is calculated as follows:

$$< S >= \frac{1}{N} \sum_{i=1}^{N} S_i \tag{2.11}$$

For a given temperature, the mean energy for all MC steps is evaluated:

$$< E >= \frac{1}{MC} \sum_{i=1}^{MC} E_i \tag{2.12}$$

2.2.3 Connection with the Contribution

The presented generalized Blume–Emery–Griffiths–Ising (BEG-I) Hamiltonian model solved by the Monte Carlo solver can theoretically reproduce the physical properties of different Magneto Caloric Materials (MCMs) with the control of order transition: i.e., First Order Transition (FOT), Second Order Transition (SOT), if the combination of the free parameters corresponds to the properties of the required material. But to find such kind of combination is somewhat challenging, because of the model sensitivity to different magneto-thermal interactions, which have an impact on the temperature and magnetic field dependence of magnetization and of heat capacity of MCMs. Thus, in order to generalize the BEG-I model for different MCMs in an automatic and time-efficient way, we propose to employ an approach based on evolutionary optimization for finding a suitable combination of the free parameters of BEG-I model (see Section 6.2). This approach applies the following algorithms:

1. an archive-based multi-objective optimization algorithm for the parallel implementation (Section 4);
2. two quantum-inspired optimization algorithms (Section 5.1, Section 5.2).

In this optimization-based method, the outputs from the Monte Carlo solver contribute to the input parameters of the objective functions though a post-processing unit for each solution candidates. The objective functions are presented in details in Section 6.2.

From an optimization point of view, the proposed method can be considered as:

1. a computationally intensive problem, taking into account that the average execution time of one run of the Monte Carlo solver of BEG-I model is 8h on Intel(R) Pentium(R) CPU 4405U @ 2.10GHz 4 processors laptop;
2. a continuous optimization problem;
3. a separable/partially separable problem, because its Hamiltonian presents the sum of three parts, where each of these part is calculated with one or two independent decision variables;
4. a small scale problem, due to the small number of free parameters of the Hamiltonian.

2.3 Model of an Active Magnetic Regenerator (AMR)

One of the problems of existing Magneto Caloric Materials (MCMs) is that their adiabatic temperature change is relatively small. E.g., the adiabatic temperature change of gadolinium in a 1T magnetic field is around 3K [Lionte et al. (2015)]. The strength of the magnetic fields of permanent magnets in room temperature domestic devices is limited to

around 2T [Bouchekara & Nahas (2012)]. To achieve a temperature span comparable to conventional refrigeration, an amplification is required. This amplification can be achieved by using the Active Magnetic Regenerator (AMR), where several thermodynamic cycles are cascaded. In the AMR cycle, the magnetic material serves not only as a refrigerant, providing the temperature change as a result of adiabatic magnetization or demagnetization, but also as a regenerator for the heat transfer fluid. It allows the temperature span to increase over the adiabatic temperature change and consequently, makes the magnetic refrigeration process effective.

This section is organized as follows: first, in Section 2.3.1, we briefly explain the working principle of AMR. Then, in Section 2.3.2 we discuss the state-of-the-art methods for optimizing the performance of AMR. Finally, in Section 2.3.3, we present the numerical AMR model provided by the *Ubiblue* company, which is used in this thesis.

2.3.1 Active Magnetic Regenerator

The Active Magnetic Regenerator (AMR) consists of a porous matrix of Magneto Caloric Materials (MCMs) traversed by an alternating fluid flow, synchronized with a magnetic field variation. In Figure 2.5, the complete 3-dimensional geometry of the AMR is schematically presented, which is considered in the development of the numerical model. The heat exchangers (hot and cold) are modeled as plates, which are placed at both ends of the regenerator. The MCM (regenerator) is in the form of several long and thin sheets. In this form, the MCM has a longitudinal thermal resistance much more important than its thickness thermal resistance. The fluid flows back and forth through the sheets in the direction of their length. The space between the regenerator and the heat exchangers shows that the heat transfer to and from the regenerator occurs only through the fluid. The pistons are omitted from the model geometry because the heat conduction through the pistons is negligible [Lionte et al. (2015)].

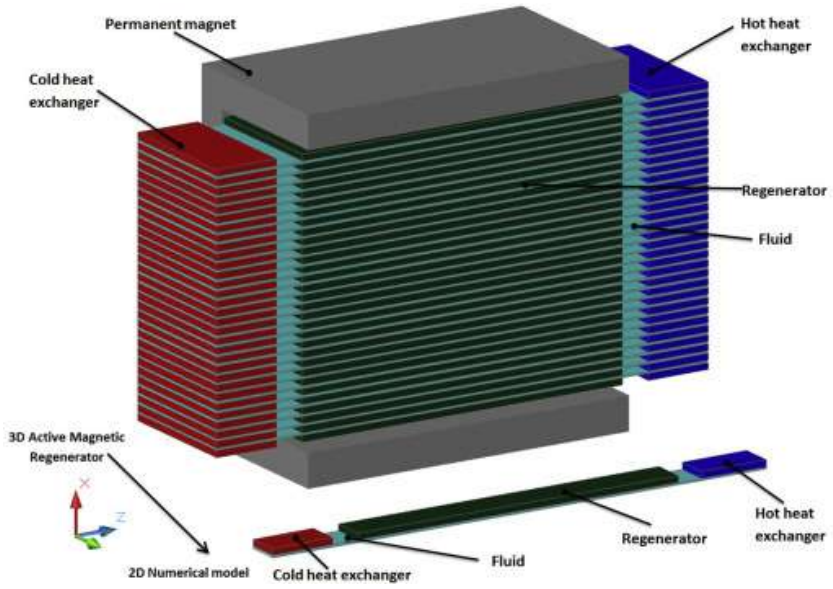


Figure 2.5: The geometry of an active magnetic regenerator. The illustration is adapted from [Lionte et al. (2015)].

The AMR is a special kind of regenerator for the active magnetic regenerative cycle

(AMR), in which the matrix of MCMs works both as a refrigerating medium and as a heat regenerating medium, while the fluid flowing in the porous matrix works as a heat transfer medium [Aprea, Greco, Maiorino & Masselli (2017)].

2.3.1.1 Active Magnetic Regenerative Cycle

An Active Magnetic Regenerative (AMR) cycle is required for operation over a temperature span larger than the adiabatic temperature change of a MCM.

The AMR cycle consists of two adiabatic stages: magnetization–demagnetization of the matrix and two isofield stages, corresponding to the heat transfer fluid flowing through the regenerator. Thus, the AMR cycle is divided on four processes related to an AMR regenerator kept in contact with a cold and a hot heat exchanger. The working principle of this cycle is schematically presented in Figure 2.6 and explained below [Aprea, Greco, Maiorino & Masselli (2017)]:

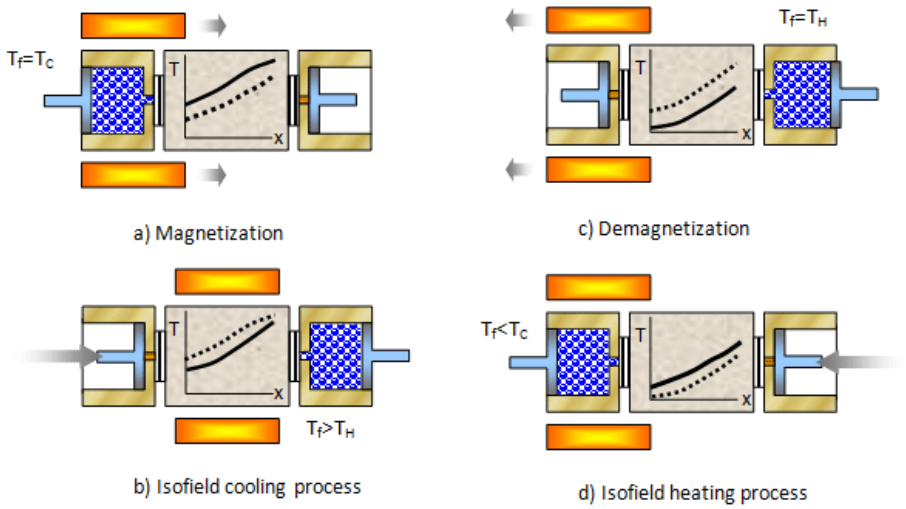


Figure 2.6: The four processes of an AMR cycle. The illustration is adapted from [Aprea, Greco, Maiorino & Masselli (2017)].

1. Adiabatic magnetization: at the beginning of the cycle, the fluid is in the cold heat exchanger at the heat absorption temperature T_c. In Figure 2.6 (a), the initial temperature profile is for the MCM matrix in its demagnetized state in the zero magnetic field (see dashed line). Then, the MCM matrix is magnetized with no fluid flow. Because of the progressive increase of the intensity of the magnetic field applied under adiabatic conditions, all the particles of the MCM, which constitutes the regenerator, warm up and the temperature of the material increases (the magnetic dipoles of the atoms align and the material's magnetic entropy and heat capacity decreases). The amount by which each particle warms up is equal to the adiabatic temperature change upon magnetization at the initial temperature of the particle, reduced by the effect of the heat capacity of the fluid in the pores between the particles. It means that the temperature of the MCM rises due the Magneto Caloric Effect (MCE) to form the final magnetized bed temperature profile (see the solid line in Figure 2.6 (a)).

2. Isofield cooling: immediately after the first step, maintaining the applied field, the fluid is circulated through the material matrix towards the hot heat exchanger (see Figure 2.6 (b)). The fluid absorbs heat from the material and transports it towards the hot heat exchanger and drives it out as long as its temperature is above the heat rejection temperature T_H.

3. Adiabatic demagnetization: after the fluid flow is stopped, the intensity of the magnetic applied field is decreased under adiabatic conditions from a maximum to a minimum value, which is cooling the matrix by the MCE (see Figure 2.6 (c)).

4. Isofield heating; finally, while the magnetic field is kept to its minimum value, the fluid is circulated from the hot to the cold heat exchanger through the regenerator, transferring heat to the MCM. The fluid cools itself down by crossing the regenerator and reaching a temperature lower than T_c (Figure 2.6 (d)).The fluid absorbs the heat in the cold heat exchanger, completing the cycle.

2.3.2 State-of-the-art Methods for Optimizing the Performance of AMR

The need for energy-efficient and environmentally friendly refrigeration, heat pumping, air conditioning, and thermal energy harvesting systems is currently more urgent than ever due to global warming [Kitanovski (2020)]. Magneto caloric energy conversion has been the subject of substantial basic and applied research over the last two decades for achieving the above mentioned goals. The subject is strongly interdisciplinary, requiring proper understanding and efficient integration of knowledge in different specialized fields [Kitanovski (2020)]. In order to achieve this understanding and consequently, to provide a new design for a magnetic device, which can work as a refrigeration system and as an energy generator system, new investigation methods are required. In this context, applying an optimization method seems promising, taking into account the large number of different parameters of such systems.

Various research works have been focused on the performance optimization of Magnetic Refrigeration System (MRS), generally by using numerical models for the simulation of the behaviour of Active Magnetic Regenerator (AMR), that consider the underlying modeled physical phenomena [Tušek et al. (2011)], [Risser et al. (2013)], [Lionte et al. (2015)]. Some research works have applied single [Teyber et al. (2018)] and multi-objective [Ganjehsarabi et al. (2016)], [Roy et al. (2017)] evolutionary algorithms to the AMR model, taking into account the different physical aspects. Another method uses an artificial neural network [Aprea, Greco & Maiorino (2017)], which is trained on an experimental data set in order to predict the energy performance.

Together, these works conducted to some improvements and helped to reduce the uncertainty of the performance for the recent prototypes [Lionte et al. (2021)]. However, commercially relevant performance for real world applications has not been demonstrated yet. It can be explained by the difficulties of conducting optimization experiments: the optimization process of the AMR model is computationally intensive, time consuming, requires special knowledge about optimization for selecting appropriate techniques and reliable optimization tools. In order to investigate the combined effects of different control and design parameters of the AMR model on its performance, various different experiments should be run.

Moreover, there is no research, which investigates the design of an AMR model for two operating modes simultaneously: a Magnetic Refrigeration System (MRS) and a Thermo-Magnetic energy Generator (TMG).

In this thesis, we propose an optimization tool for investigating the design of the dual-mode operating AMR of the *Ubiblue* company, which allows scientists to easily set up optimization experiments independently from any modifications of the model and any configurations of the optimization problems.

2.3.3 The Active Magnetic Regenerator Model

In this section, we present a description of a numerical model of the Active Magnetic Regenerative (AMR) device, which is used in optimization functions for finding the common design for two operating modes of the AMR: a Magnetic Refrigeration System (MRS) and a Thermo-Magnetic energy Generator (TMG).

This model is provided by the *Ubiblue* company and is under active development since 2013. The presented description is approximate, due to commercial confidentiality. It is mainly based on the first version of the model published by *Ubiblue* in [Risser et al. (2013)].

The numerical model of *Ubiblue* takes into account five main components of the basic AMR device: the regenerator (the matrix of Magneto Caloric Materials (MCMs)), the hot and cold heat exchangers, the displacer (or pump) and the magnet. It consists of the following parts: (i) a 3-dimensional model for magnetic circuits; (ii) a 2-dimensional model of thermal and magneto caloric behavior of the regenerator and (iii) a 1-dimensional model of the advection inside regular micro-fluid channels [Risser et al. (2013)].

This model is multi-physic and multi-scale, where the magnetic model is represented at micro scale by the electron's spins alignment under an applied magnetic field. The fluid flow is represented at mini scale by the heat transfer fluid. And the heat transfer is represented at macro scale by the thermal exchange via the heat transfer fluid.

The thermal and magnetic models are coupled via the magnetic field, which is at the origin of the Magneto Caloric Effect (MCE), and through temperature and its impact on the magnetization of the MCM [Risser et al. (2013)]. Figure 2.7 presents a scheme of the basic model of the magneto caloric device [Risser et al. (2013)], where the 3-dimensional magnetic model and the 1-dimensional model of heat advection in the coolant are schematized.

The 1-dimensional model of heat advection in the coolant fluid is shown in the right side of the Figure 2.7 and is described by Equation 2.13:

$$\rho_f C_f \left(\frac{\partial T_f}{\partial t} + u \frac{\partial T_f}{\partial x} \right) = \frac{\partial}{\partial x} \left(\lambda_f \frac{\partial T_f}{\partial x} \right) + (\dot{Q}_{visco}) + (\dot{Q}_{HT}) \tag{2.13}$$

where ρ_f is the density of the coolant, C_f is its heat capacity, T_f is its temperature, λ_f is its thermal conductivity and u is a speed. $\dot{Q}_{visco}$ is the heat dissipation due to the viscosity. $\dot{Q}_{HT}$ is the heat transfer between the coolant and the MCM.

Equation 2.14 presents the 2-dimensional model of heat in an MCM:

$$\rho_s C_{H_i,p} \frac{\partial T_s}{\partial t} = \lambda_s \left(\frac{\partial^2 T_s}{\partial x^2} + \frac{\partial^2 T_s}{\partial y^2} \right) + (\dot{Q}_{MC}) + (\dot{Q}_{leak}) - (\dot{Q}_{HT}) \tag{2.14}$$

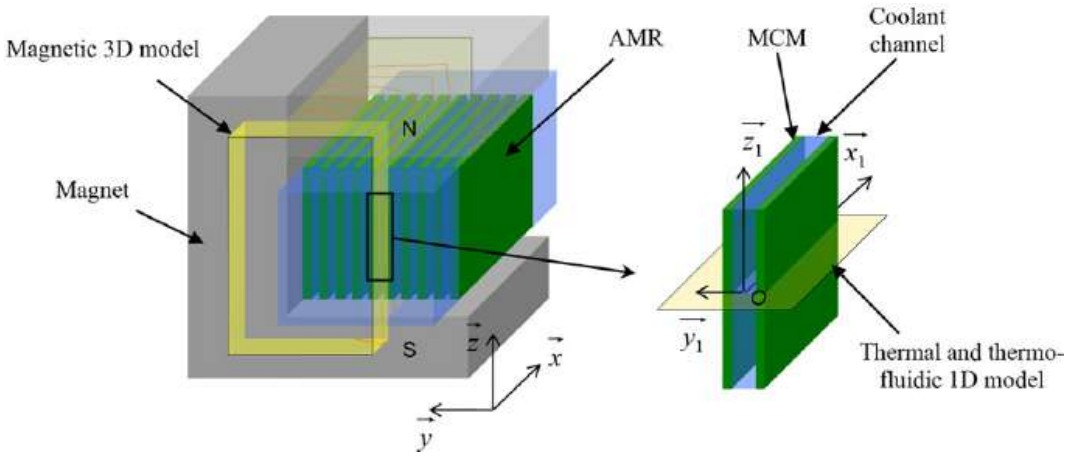


Figure 2.7: Scheme of the modelled magneto caloric device. The illustration is adapted from [Risser et al. (2013)].

where ρ_s is the density of the material, $C_{H_i,p}$ is the heat capacity of MCM at constant magnetic field and pressure, T_s its temperature and λ_s is the thermal conductivity. $\dot{Q}_{leak}$ represents the heat leakages due to the imperfect thermal insulation around the AMR. $\dot{Q}_{MC}$ is a magneto caloric coupling, which corresponds to the generation of heat or cold from the MCE and defined in Equation 2.15:

$$\dot{Q}_{MC} = \frac{\partial T_{ad}(T_s, H_i)}{\partial H_i} \cdot \frac{\partial H_i}{\partial t} \rho_s C_{H_i,p}(T_s, H_i) \tag{2.15}$$

As seen from Equation 2.15, $\dot{Q}_{MC}$ can be calculated from the adiabatic temperature change ($\partial T_{ad}/\partial H_i$), due to the varying of internal magnetic field H_i given by the magnetic model. This term creates the first connection between the magnetic model and the thermal model.

The Induction, B (Equation 2.16), in the volume of magnetic material is related to the external magnetic field (H_e) applied on the body, to the spontaneous magnetization (M) and to the geometry of the body. Because M depends on the temperature of the material and of the internal magnetic field H_i, this term creates a second interconnection between the magnetic model and the thermal model. By increasing or decreasing the temperature, the MCM will vary its magnetization and will switch from the ferromagnetic state to the paramagnetic state and vice versa at the vicinity of the Curie temperature T_c.

The convergence of field lines towards the body is restricted by the geometric configuration of the device. To guarantee the conservation of the magnetic flux in a heterogeneous area, a term of demagnetizing field (Hd) is needed. Hence, geometric restrictions are modelled as follows:

$$\vec{B} = \mu_0 \cdot (\vec{H_e} + \vec{M}(T_s, H_i) + \vec{H_d}) \tag{2.16}$$

where μ_0 is the magnetic permeability of vacuum, $H_d = f(r, T(r))$ is the demagnetizing field, r is the position in the space, H_i is the internal magnetic field, H_e is the external magnetic field, M is the magnetization.

The equations for the magnetic model, the thermal model and the fluid model are solved iteratively with a Gauss-Seidel method.

To evaluate the performance of an AMR system, the average power equations over an AMR cycle in steady state, provided by the hot and cold sources, are as follows:

$$\dot{Q}_{Cold} = \frac{1}{\Pi}\int_0^t \dot{M} \cdot C_{p,f}(T_{cold} - T_f(t))dt \tag{2.17}$$

$$\dot{Q}_{Hot} = \frac{1}{\Pi}\int_0^t \dot{M} \cdot C_{p,f}(T_f - T_{Hot}(t))dt \tag{2.18}$$

where Π is the AMR cycle period.

The Coefficient Of Performance (COP) is evaluated as follows:

$$COP = \frac{\dot{Q}_{Cold}}{(W_{mag} + W_{pump})} \tag{2.19}$$

where W_{mag} is the magnetic work and W_{pump} is the pump work.

In this model, the AMR cycle is modelled as the four sequential steps explained above. The same time step t has been chosen for the resolution during all the four periods of the cycle. The cycle is repeated several times with a constant operating frequency until the regenerator reaches a steady state operation.

In this thesis, we consider the presented model of the AMR device as a function with the following operations:

1. an initialization of the input parameters;
2. a solver of the above defined equations;
3. a post processing unit, which provides the output parameters.

Input parameters of the model

The input parameters of the model can be mainly divided into two groups:

1. The design parameters: e.g., the length of the MCM plate, the plate thickness (height of the MCM plate), the fluid channel thickness (the height of the fluid microchannel), the number of blades of the MCM, the number of fluid channels, the porosity of regenerator, etc.
2. The control parameters: e.g., the fluid velocity, the operating frequency, the initial system temperature, the magnetic field change value, the ratio of coolant volume transferred at each half AMR cycle on the AMR fluid volume, etc.

Output parameters of the model

In the post processing unit, the following output parameters can be evaluated:

1. The energy efficiency of the Magnetic Refrigeration System (MRS): $\eta = COP/COP_{Carnot}$, where COP is the coefficient of performance (Equation 2.19) and COP_{Carnot} is the Carnot coefficient of performance;
2. The energy efficiency of the Thermo-Magnetic energy Generator (TMG): η/η_{Carnot}, where η_{Carnot} is the Carnot yield;

3. The thermal power density of the MRS: $\dot{Q}_{cold}/V_{AMR}$, where V_{AMR} is the AMR volume ratio and $\dot{Q}_{cold}$ is the refrigeration power;

4. The mechanical power density of the TMG: $\dot{W}_r/V_{AMR}$, where $\dot{W}_r$ is the recovery power.

From an optimization point of view, the input parameters of the model are the decision variables of an optimization problem, different combinations of which have to be investigated in order to ensure a better performance of the model. The output values of the post processing unit are assumed as being the values of objective functions.

Taking into account the number of different optimization problems (e.g., different combinations of input parameters and different number of objectives taken into consideration) and different modifications of the model (e.g., different magnetic materials are used) an efficient optimization tool must be provided in order to rapidly set up experiments and to comprehensively investigate multiple parameters in the search space and their effect on the system performance.

2.3.4 Connection with the Contribution

Aiming at achieving the best technical-economic compromise for the industrial applications, we are looking for innovative architectures for the presented model of the Active Magnetic Regenerative (AMR) system, which will be compact, lightweight, powerful and energy efficient for the Magnetic Refrigeration System (MRS) and the Thermo-Magnetic energy Generator (TMG) modes. For this purpose, we propose a many-objective optimization tool in Section 6.3. This tool is a flexible instrument, which allows users to explore the AMR model design through various optimization experiments by taking into account different control and design parameters of the AMR. In order to do this, we analyse the AMR model as an optimization problem and employ appropriate optimization techniques through the EASEA platform w.r.t. the results of this analysis.

From an optimization point of view, the AMR model of *Ubiblue* is considered as:

1. a computationally intensive problem, taking into account that the execution time of one simulation run of single-mode operating AMR model has high variance and takes up to 15h on an AMD EPYC 7371 16-Core Processor;

2. a continuous optimization problem;

3. a black-box optimization problem;

4. a small, medium or large scale problem, where the number of decision variables depends on the number control and design parameters, which are taken under consideration;

5. a multi or many objective problems.

Moreover, during our preliminary experiments, it was revealed that the AMR model can provide:

1. several non-dominated solutions, which can be identical in the objective space and different in the search space;

2. dominance resistant solutions, which means that some solutions can have extremely good values for some objectives and extremely bad values for other objectives.

The description of optimization problems of the AMR model will be presented in Section 6.3.

Chapter 3

Algorithms for Continuous Optimization

In this chapter, we review continuous optimization algorithms for solving computationally intensive problems on different objective (target) space dimensions.

First, we recall the fundamentals of single-objective and multi-objective optimization in Section 3.1.1 and Section 3.1.2 respectively. Next, we briefly review the state-of-the-art methods for efficiently solving computationally intensive problems (Section 3.2). Then, taking into account the quantum nature of the model of magneto caloric materials, we turn to the quantum-inspired techniques for continuous optimization, which we overview in Section 3.3. Next, in Section 3.4, we give the descriptions of the state-of-the-art algorithms and methods, which are used in this thesis. Further, we provide the description of the benchmark suites (Section 3.5), which are used in this thesis. Finally, in Section 3.6, we present EASEA (*EAsy Specification of Evolutionary Algorithms*) [Collet et al. (2000)], an evolutionary algorithms specification framework which is applied as a support for this thesis.

3.1 Evolutionary Optimization

Evolutionary Algorithms (EAs) are stochastic derivative-free population-based optimization algorithms inspired from a simplified model of biological evolution.

In Figure 3.1, the conventional structure of an EA is schematically presented, which generally includes five main functions: **initialization**, **selection**, **crossover**, **mutation** and **evaluation**. Starting with a randomly initialized population of candidate solutions (called individuals), at each iteration/generation, candidate solutions are evolved by selection, crossover and mutation operators until a stopping criterion is reached with a possible satisfactory solution.

Before applying an EA to solve an optimization problem, a formulation of the problem is needed. There are three main components that are required to be specified for formulating any kind of optimization problem:

1. **Decision variables** (called genes), which encode candidate solutions to the problem. They are the values, which are manipulated in order to maximize or minimize the objective function(s). These can take on discrete (i.e., integer) or continuous (i.e., real) values, depending on the problem under consideration.

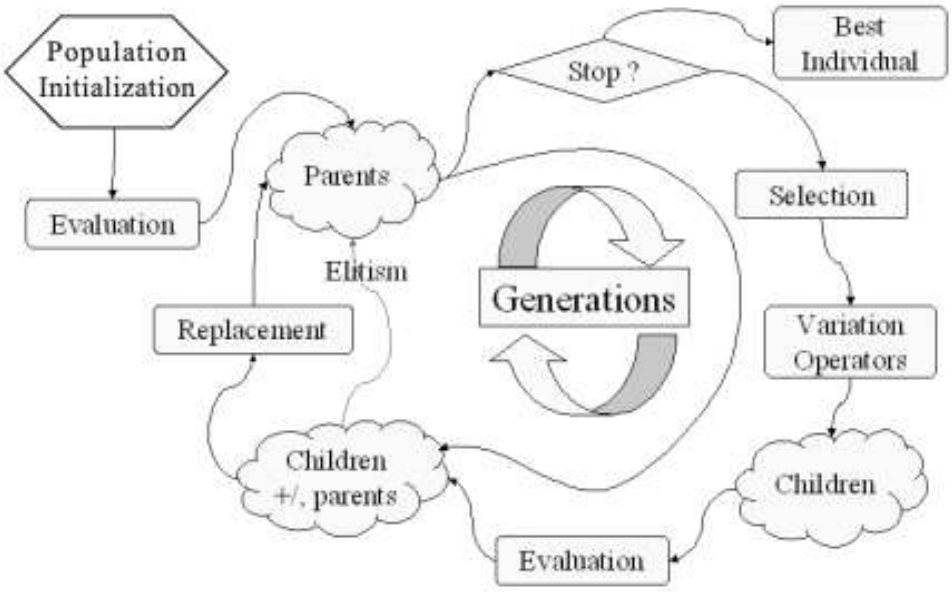


Figure 3.1: General structure of an evolutionary algorithm. The illustration is adapted from [Collet & Rennard (2008)].

In this thesis, for the both models, i.e., the model of Active Magnetic Regenerator (AMR) and the model of magneto caloric materials (MCMs), the values of the decision variables are considered as continuous.

2. **One or several objective function(s)**, which represent the value(s) that should be maximized (e.g., in case of the AMR model, it can be the energy efficiency of the magnetic refrigeration system and of the thermo-magnetic energy generator) or minimized (e.g., in case of the model of MCMs, it can be the difference between the expected and simulated values of different properties of magneto caloric materials). Note, that it is not sufficient to evaluate solutions as right or wrong, the objective value(s) needs to indicate the quality of a candidate solution: i.e., an objective function that returns either 0 or 1 is useless.

3. **Constraints**, which can generally be placed on the values that the decision variables can take, or used to avoid undesirable/infeasible system responses.

A solution is defined as a set of values of the decision variables, and a feasible solution is one that satisfies all problem constraints. The quality of different solutions is evaluated using the objective function.

Once the problem is formulated, an Evolutionary Algorithm (EA) can be applied to solve the optimization problem. In order to explain the working principle of an EA, we detail below its main steps:

1. `Initialization` randomly initializes the candidate solutions (each of them is presented as the set of the decision variables) of the initial population.

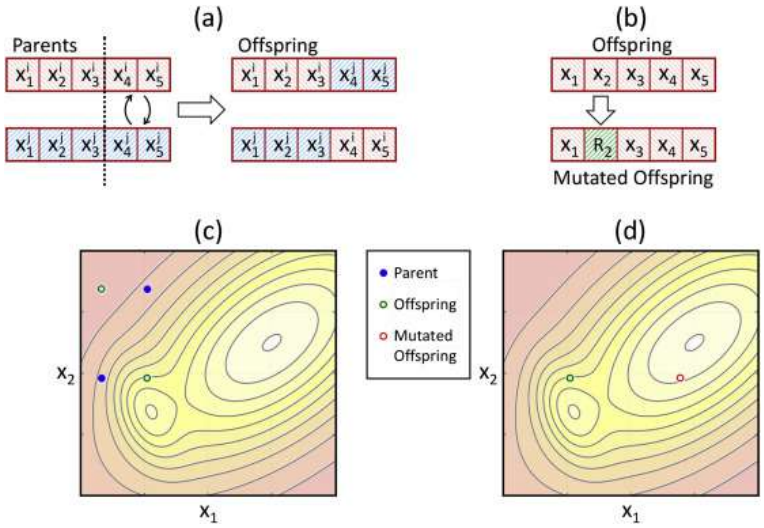


Figure 3.2: An example of crossover and mutation operators and their performance on a fitness landscape. (a) A single-point crossover that reproduces two offspring from two parents in a 5-dimensional space of decision variables. (b) A uniform mutation that replaces the value of a decision variable by a uniformly distributed random number between the variable lower and upper bounds. Example outcome of (c) single-point crossover and (d) uniform mutation in a 2-dimensional search space. The illustration is adapted from [Maier et al. (2019)].

2. `Evaluation` is a way to evaluate the quality of a candidate solution. The evaluation function determines the objective of each candidate solution of the population.

3. `Selection` pick the fittest (i.e., the best in terms of objective function) parent solutions in the current population with certain probability for producing the new candidate solutions (i.e., the offspring or children).

4. `Variation operator - Crossover` produces one or several new solutions given two or more parents by combining the decision variables that differ from those of either parents. There are many different types of crossovers, and depending on their type, crossover can work for exploitation (intensification) and for exploration (diversification) during the search process.

   As an example, in Figure 3.2 (a), a single point crossover is applied to two parent solutions i and j, which are presented on a 5-dimensional search space: i.e., each solution of problem consists of 5 decision variables. In this crossover, a single point on the solution is selected randomly, and all decision variable values beyond a selected point are swapped between the two parents to create one or two children. The result of this crossover is shown in Figure 3.2 (c) on a 2-dimensional search space, where the parent solutions are presented in blue and the children in green.

5. `Variation operator - Mutation` is usually applied after crossover and changes the values of some decision variables in the child population with a specified probability. E.g., in Figure 3.2 (b) an example of mutation is shown: one of the decision variables is randomly selected and replaced by a uniformly distributed random number in its feasible range. Figure 3.2 (d) illustrates the result of mutation in

a 2-dimensional search space, where the second variable x_2 is mutated. Mutation works to preserve and introduce diversity during the search, which allows EAs to escape local optimums.

6. `Replacement` selects the candidate solutions to create the parent population for the next generation.

To summarize, the selection and replacement operator reduce diversity and act as a force pushing for quality, whereas the variation operators, i.e., crossover and mutation, create the necessary diversity of the population and thereby facilitate novelty.

In this thesis, we consider to use EAs for different objective (target) space. For this reason, in the next Sections, we present some fundamentals of single- and multi- objective optimization.

3.1.1 Single-objective Optimization

Single-objective optimisation aims at finding the best solution, which corresponds to the minimum or maximum value of a single objective function (also called fitness function). Formally, for the minimization case, a single-objective problem can be represented by the following equation:

$$f(\boldsymbol{x}_{min}) = min\{f(\boldsymbol{x})\} \tag{3.1}$$

where $\boldsymbol{x} \in \mathbb{R}^{\mathrm{d}}$ denotes the vector of decision variables, i.e., candidate solutions, to be optimized, d is the number of decision variables, i.e., the dimension of the search space and f is the optimization function.

Vector $\boldsymbol{x}$ has to be found using a technically feasible number of function evaluations, where the number of function evaluations is used as a common search cost measure.

Engineering and scientific problems (e.g., the problems of the AMR or MCMs models) appear with various properties such as being low or high dimensional, separable or non-separable, computationally expensive, etc. Combinations of these properties lead to special requirements for an EA to solve such kinds of problems [Shan & Wang (2010)]. Below we list some of these properties:

Cost:

A function $f(\boldsymbol{x})$ is called computationally intensive/expensive or costly, if one evaluation of this function for one solution $\boldsymbol{x}$ takes more time than the time, which is needed for one cycle of the optimization process (excluding function evaluations). The optimization of computationally intensive function is challenging, because the optimization process is limited by the relatively small number of function evaluations, which can be used.

Separability, Partial Separability, Non-Separability:

Separability: A function $f(\boldsymbol{x}) : \boldsymbol{x} \in \mathbb{R}^{\mathrm{d}}$ is called separable if the optimum of this function can be achieved by performing d independent one-dimensional searches along each independent decision variables while keeping the other variables fixed. A separable function does not show any dependencies between the decision variables, and consequently, is much easier to solve than non-separable one: the optimization of separable functions can break the curse of dimensionality, because the problem complexity grows linearly with d.

Non-Separability: A function $f(\boldsymbol{x}) : \boldsymbol{x} \in \mathbb{R}^{\mathrm{d}}$ is called non-separable if the optimum of this function cannot be achieved by a sequence of one-dimensional searches, meaning that the decision variables of $\boldsymbol{x}$ are correlated.

Partial Separability: A function $f(\boldsymbol{x}) : \boldsymbol{x} \in \mathbb{R}^{\mathrm{d}}$ is partially separable, if f is non-separable and has sub-sets of decision variables which can be optimized separately.

Some algorithms explicitly or implicitly exploit separability and therefore, usually perform quite well on separable problems. The number of function evaluations to reach the optimum objective value then may scale almost linearly with d.

Uni-modality and Multi-modality:

The property of uni- and multi-modality is correlated with the aim of the optimization and the notion of fitness landscape, which graphically represents its optimization problem by depicting the shape of the objective (fitness) function w.r.t. the decision variables (e.g. Coefficient Of Performance (COP) of the model of active magnetic regenerator as a function of different values of model control and design parameters for model optimization problems).

In Figure 3.3 an example of fitness landscape for an single-objective function from two decision variables is presented. As we can see, a single objective optimization problem has a single fitness landscape and the aim of the optimization is to find the feasible combination of values of decision variables that results in the highest value of the objective function (for maximization problems). This value corresponds the highest peak in the fitness landscape for the problem under consideration, which is referred as the global optimum (see Figure 3.3).

The global optimum is the best solution among all possible candidate solutions.

The local optimum is one or multiple other optima in the fitness landscape with lower peaks (for maximization problems) than the global one. They are referred as local, because their respective candidate solutions are optimal only within their neighbourhood in the decision variable space.

The global and local optima of problem landscape are illustrated in Figure 3.3.

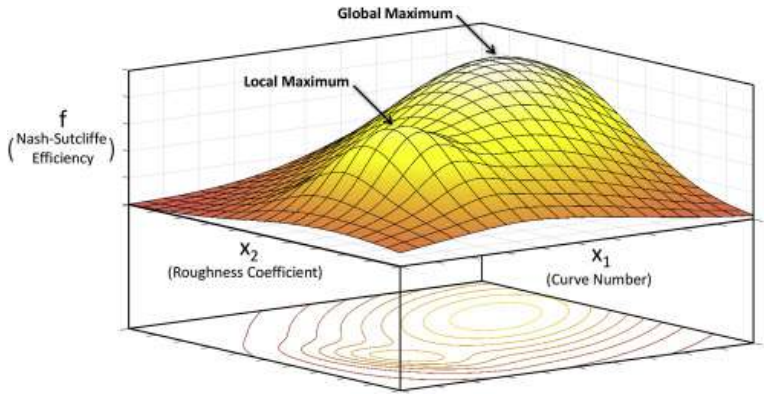


Figure 3.3: An example of a typical fitness landscape with local and global optima, where f is the objective function to be maximized and x_1 and x_2 are the decision variables. The illustration is adapted from [Maier et al. (2019)].

An objective function $f(\boldsymbol{x}) : \boldsymbol{x} \in \mathbb{R}^{\mathrm{d}}$ is called uni-modal, if it has only one local optimum which is at the same time also its global one.

A multi-modal function has at least two global/local optima: i.e., it can have non-unique global optima.

Multi-modal functions are highly common in practical optimization. If the fitness landscape is very rugged, i.e., it has a lot of local optima, finding the highest point in the landscape is very difficult, due to the difficulty to navigate the landscape and to know if an optimum that has been found is the global or not. Consequently, the optimization of multi-modal functions is difficult, because there are no guarantees that the optimum, which was found by an optimization algorithm is the global one. Moreover, because of the stochastic nature of EAs, they can converge towards one or another local optimum depending on the sampling of the search space.

Ill-Conditioning:

The conditioning of a problem is, roughly speaking, the difference in sensitivity of the objective function when varying different variables. On an ill-conditioned function, the length of search steps performed in different directions of the search space, may differ by orders of magnitude to produce the same improvements of the objective function. Thus, ill-conditioned problems are difficult for optimization, because before we learn an appropriate metric we often make too short or too long search steps.

High-Dimensionality of the Search Space:

In Figure 3.3, the fitness landscape of an optimization problem with two decision variables is presented. Obviously, in case of higher-dimensional problems with three or more decision variables, the fitness landscape cannot be visualised easily.

The volume of the search space increases exponentially with the number of decision variables d. Thus, some EAs, useful for small dimensions, become useless for large dimensions. This effect is called the curse of dimensionality [Bellman (1957)], because many important techniques such as distance, or neighborhood, become less meaningful with increasing dimension, due to a loss of contrast of distances [Houle et al. (2010)].

Dynamic:

An objective function $f(\boldsymbol{x}, t) : \boldsymbol{x} \in \mathbb{R}^{\mathrm{d}}$ is called dynamic, when the objective value for a given $\boldsymbol{x}$ depends on the time-step t. I.e., the objective function is changing in time: it may be a simple rotation of the search space or any other modifications of f. In dynamic optimization the goal is to find the global optimum of f for a time-step t and in the best case, predict its location for time-steps $t < T$.

In general, the relatively simple working mechanism of single-objective EAs allows them to find the global and local optima (see Figure 3.4). As shown in Figure 3.4 for an example with a 2-dimensional fitness landscape (the same landscape that is shown in Figure 3.3), EAs begin with a randomly distributed initial population (Figure 3.4 (a)) and an exploration-oriented search in the first iterations to locate the main regions of attraction (Figure 3.4 (b)). As the search continues, it becomes more exploitation-oriented in the regions of attraction, and identifies the best solution in the final iteration (Figure 3.4 (c)).

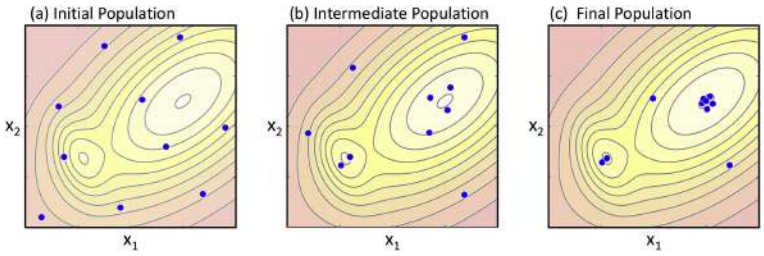


Figure 3.4: An example of the performance of an evolutionary algorithm with population size = 10. (a) An initial population of candidate solutions, which are randomly distributed on a 2-dimensional search space. (b) An intermediate population after several number of generations where local and global regions of attraction have been identified. (c) The final population, which has converged to local and global optima. The illustration is adapted from [Maier et al. (2019)].

3.1.2 Multi-objective Optimization

In this thesis, we consider to use optimization algorithms for solving different problems of the model of Active Magnetic Regenerator (AMR) and of the model of Magneto Caloric Materials (MCMs). In both cases, an optimization problem can be formulated with more than one objective functions. E.g., an optimization problem formulated for investigating the design of dual-mode operating model of AMR, can have at least four competing objectives: i.e., the energy efficiency of the Magnetic Refrigeration System (MRS), the energy efficiency of the Thermo-Magnetic energy Generator (TMG), thermal power density of magnetic refrigeration system and the mechanical power density of the thermo-magnetic energy generator. The mentioned objectives were explained earlier in Section 2.3.3.

Consequently, as seen in Figure 3.5 (a) and (b), when there are more than one objective, each objective has its own fitness landscape, as variations in objective values are likely to be different for different objectives with the same changes in the values of decision variables.

These objectives compete with each other, so that solutions that improve values of one objective might degrade values in another (antagonistic objectives).

Note that multi-objective optimization problems having more than three objectives are referred to as many-objective optimization problems.

Consequently, in this section, we recall the fundamentals of multi-/many-objective optimization and the indicators for their performance assessment.

The definitions, equations and terms presented in this section correspond to the mathematical formulations, commonly used in the multi-objective optimization literature [Knowles et al. (2006)]).

3.1.2.1 Fundamentals of Multi-objective Optimization

Formally, the fundamentals of multi-objective optimization can be defined as follows:

Definition 1 - Multi-objective Optimization Problem (MOP): A general MOP includes a set of d decision variables, a set of $k \geq 2$ objective functions, and optionally, a set of $m \geq 0$ constraints. Objective functions and constraints are functions of the decision variables. A MOP is mathematically formulated in Equation 3.2:

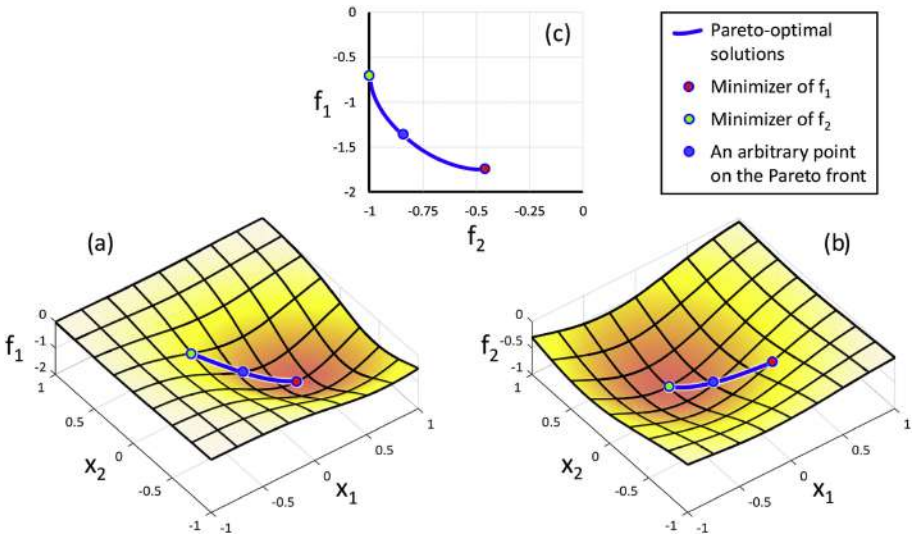


Figure 3.5: Illustration of the relationship between (a) the fitness landscape of objective 1, (b) the fitness landscape of objective 2 and (c) the Pareto-front of the two objective optimization problem. The illustration is adapted from [Maier et al. (2019)].

$$\begin{aligned} \textit{maximize } \boldsymbol{y} = \boldsymbol{f(x)} &= (f_1(\boldsymbol{x}), ..., f_k(\boldsymbol{x})) \\ \textit{subject to } \boldsymbol{e(x)} &= (e_1(\boldsymbol{x}), e_2(\boldsymbol{x}), ..., e_m(\boldsymbol{x})) \leq 0 \\ \textit{where } \boldsymbol{x} &= (x_1, x_2, ..., x_d) \in \boldsymbol{X}; \\ \boldsymbol{y} &= (y_1, y_2, ..., y_k) \in \boldsymbol{Y} \end{aligned} \tag{3.2}$$

where $\boldsymbol{x}$ is the vector of decision variables, $\boldsymbol{y}$ is the objective vector, $\boldsymbol{X}$ is the continuous search space $\mathbb{R}^{\mathrm{d}}$ and $\boldsymbol{Y}$ is the objective space $\mathbb{R}^{\mathrm{k}}$. If the MOP has constraints ($\boldsymbol{e(x)} \leq 0$), these constrains determine the set of feasible solutions.

Definition 2 - Pareto Dominance: The objective vector $\boldsymbol{y}^a$ dominates the objective vector $\boldsymbol{y}^b$ ($\boldsymbol{y}^a \succ \boldsymbol{y}^b$)$\Longleftrightarrow$($y_j^a \geq y_j^b$ for all $j \in \{1, ..., k\}$ and $y_n^a > y_n^b$ for at least one $n \in \{1, ..., k\}$).

Definition 3 - Weak Pareto dominance: The objective vector $\boldsymbol{y}^a$ weakly dominates the objective vector $\boldsymbol{y}^b$ ($\boldsymbol{y}^a \succeq \boldsymbol{y}^b$)$\Longleftrightarrow$($y_j^a \geq$y$_j^b$ for all $j \in \{1, ..., k\}$).

Definition 4 - Strict Pareto dominance: The objective vector $\boldsymbol{y}^a$ strictly dominates the objective vector $\boldsymbol{y}^b$ ($\boldsymbol{y}^a \succ\succ \boldsymbol{y}^b$)$\Longleftrightarrow$($y_j^a > y_j^b$ for all $j \in \{1, ..., k\}$).

Definition 5 - Incomparability: The objective vector $\boldsymbol{y}^a$ is indifferent to the objective vector $\boldsymbol{y}^b$ ($\boldsymbol{y}^a \parallel \boldsymbol{y}^b$)$\Longleftrightarrow$($\boldsymbol{y}^a \not\succeq \boldsymbol{y}^b \wedge \boldsymbol{y}^b \not\succeq \boldsymbol{y}^a$).

Definition 6 - Pareto Optimality: The solution vector $\boldsymbol{x}^{opt}$ and its corresponding objective vector $\boldsymbol{y}^{opt} = \boldsymbol{f(x}^{opt})$ are Pareto optimal$\Longleftrightarrow$if there exists no $\boldsymbol{y} \in \boldsymbol{Y}$ such that $\boldsymbol{y} \succ \boldsymbol{y}^{opt}$.

Definition 7 - Pareto Front Approximation: Let $\boldsymbol{P} \subseteq \boldsymbol{Y}$ be a set of objective vectors. $\boldsymbol{P}$ is called a Pareto front of non-dominated solutions $\Longleftrightarrow$ any vector of $\boldsymbol{P}$ does not weakly dominate any other vector in $\boldsymbol{P}$. Vectors of a Pareto front are called non-dominated vectors.

Definition 8 - Optimal Pareto front: An optimal Pareto front is a non-dominated front of optimal Pareto solutions $\boldsymbol{f(x}^{opt})$ and corresponding vectors $\boldsymbol{y}^{opt}$ (see Figure 3.5 (c)).

Definition 9 - Better front: An approximation set A is better than an approximation set B $(A \rhd B) \Longleftrightarrow$ every $y^b \in B$ is weakly dominated by at least one $y^a \in A$ and $A \neq B$.

Why are Multi-objective Optimization Problems (MOPs) are difficult to solve? The following reasons give the answer on this question:

1. They inherit properties of each objective and as with single-objective optimization, large and complex search space makes the search difficult and can preclude the use of certain optimization methods.

2. It is not obvious to find the trade-off between conflicting criteria and to select the most suitable solution, because it is not just one point, but a set of Pareto points, as shown in Figure 3.5 (c). This gets more difficult with growing dimensionality of the objective (target) space, especially when the solutions are incomparable (see Definition 5).

3. The algorithm computational complexity grows with increasing dimensionality of the objective space.

4. Having different multiple objectives makes it difficult to assess performance, because several conflicting goals must be considered: (i) to minimize the distance to the optimal Pareto front and (ii) to maximize the diversity of solutions in an approximation of the optimal Pareto front.

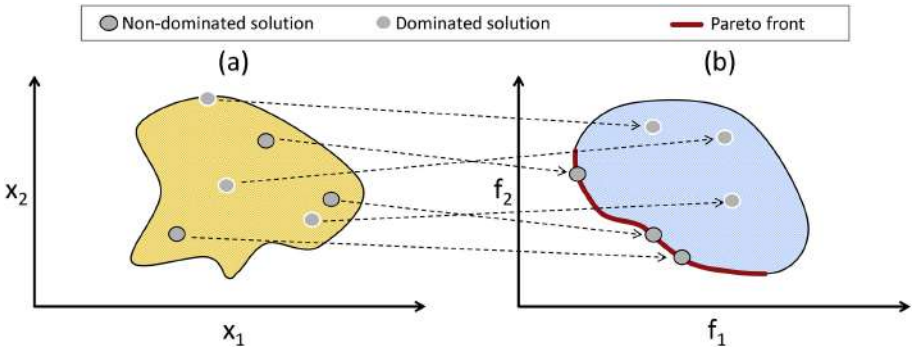


Figure 3.6: Mapping of (a) a space of decision variables onto (b) space of objectives, where the objectives are to be minimized. The illustration is adapted from [Maier et al. (2019)].

The relationship between decision variables and objective functions: in order to illustrate the relationship between decision variables and objective functions of a multi-objective optimization problem, in Figure 3.6 we present the mapping from the solution space to the objective space for a hypothetical multi-objective problem with two decision variables (i.e., 2-dimensional search space) and two objective functions (i.e., 2-dimensional objective space), both of which are to be minimized. As seen from Figure 3.6, the solutions that are non-dominated in the objective space lie on the Pareto front, whereas dominated solutions do not.

3.1.2.2 Performance Indicators

In order to estimate the comparative performance of multi-objective optimization algorithms, quality indicators are used. The use of quality indicators helps to quantify differences between Pareto fronts, in order to determine which approximation is better (in terms of Definition 9) even for incomparable solutions (see Definition 5).

The quality indicators can be Pareto-compliant or Pareto non-compliant:

Definition: (Pareto-compliant indicator) The indicator $I : \Omega \rightarrow \mathbb{R}$ is Pareto-compliant for every pair of approximation sets $\boldsymbol{A}$ and $\boldsymbol{B}$, for which $\boldsymbol{A} \succeq \boldsymbol{B}$, *I(A)* is not worse than *I(B)*.

Hypervolume

In this thesis, in order to comparatively study different multi-objective optimization algorithms, we use the Pareto-compliant hypervolume (HV) performance metric [Wagner et al. (2007)], as the most commonly used indicator (e.g., in Black-Box Optimization Benchmarking workshops[1]).

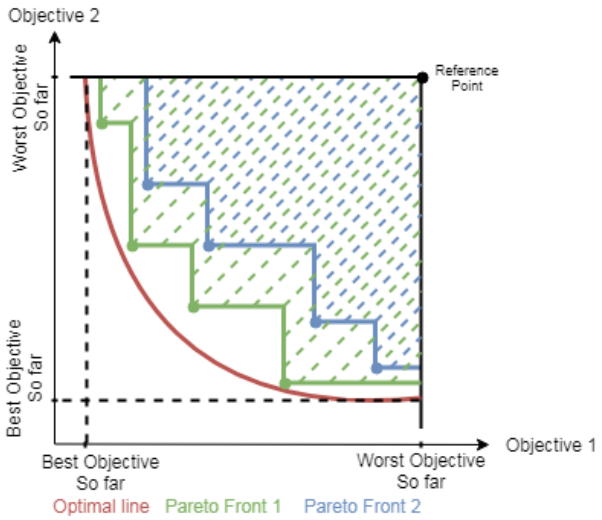


Figure 3.7: Illustration of the Hypervolume indicator (minimization problem assumed). The illustration is adapted from [Demir et al. (2019)].

[1] `https://coco.gforge.inria.fr/`

Let $\boldsymbol{p}_{ref}$ denotes a reference point (also called nadir point), dominated by all points in a set of points $\boldsymbol{P}$. The HV of $\boldsymbol{P}$ is the volume of the union of the hyper-cubes, defined by one point of the set $\boldsymbol{P}$ and $\boldsymbol{p}_{ref}$:

$$HV(P) = \text{Volume}(\bigcup_{i=1}^{k} Rect(\boldsymbol{p}_i, \boldsymbol{p}_{ref})) \tag{3.3}$$

where k is the number of points in the set $\boldsymbol{P}$ and $Rect(\boldsymbol{p}_i, \boldsymbol{p}_{ref})$ is the hyper-rectangle, whose diagonal is the segment $[\boldsymbol{p}_i, \boldsymbol{p}_{ref}]$. According to Equation 3.3, only the non-dominated points of $\boldsymbol{P}$ contribute to the HV.

Figure 3.7 illustrates the Hypervolume indicator. In fact, it indicates the volume between the reference point in the solution space and the Pareto front found by the solution approach (e.g., the green hyper-rectangle for Pareto front 1 in the Figure 3.7).

Pareto Ranking

The Pareto rank is an indicator of the solution quality based on Pareto dominance (see Definition 2 in Section 3.1.2.1). An iterative Pareto ranking procedure is presented in [Fonseca et al. (1993)]. All non-dominated points of set $\boldsymbol{P}$ are assigned rank 1 and temporarily removed from this set. Then, the next non-dominated points are assigned rank 2 and the process continues until all points of $\boldsymbol{P}$ have received a Pareto rank. Consequently, this indicator is related to the whole population. This indicator is widely used by the different MOEAs in Non-Dominated Sorting (NDS) procedure: a sorting of points w.r.t. the Pareto rank [Deb et al. (2000)], [Deb & Jain (2013)].

3.1.3 Pros and Cons of Evolutionary Algorithms

Now, when all necessary notions are introduced, we can explain our choice to employ Evolutionary Algorithms (EAs) for solving the defined research problems.

Main advantages of EAs:

1. EAs can be linked with simulation models to explore large solution spaces. In

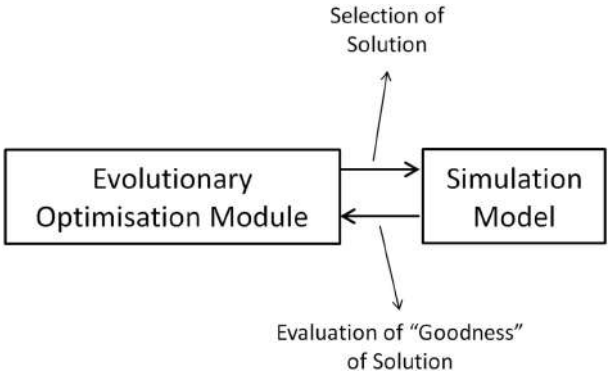


Figure 3.8: Illustration of coupling between an evolutionary optimization module and a simulation model. The illustration is adapted from [Maier et al. (2019)].

Figure 3.8, we show the coupling process between a simulation model and an optimization algorithm: (i) the optimization algorithm provides the values of decision variables that are transmitted to the simulation model; (ii) the simulation model

evaluates the corresponding objective function(s) and (iii) the objective function(s) are transmitted back to the optimization algorithm. Consequently, if the performance of a system (e.g., the active magnetic regenerator) is simulated using an existing model, it can be optimized using EAs.

2. In the case when the traditional analytical methods or other techniques cannot find a good enough solution, EAs can provide good approximate solutions for different problems, if they exist. For example, in Figure 3.9, we present the results of the

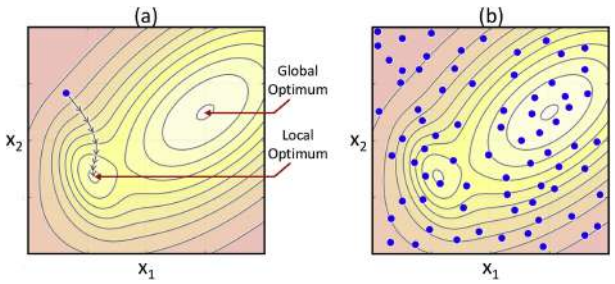


Figure 3.9: Examples of (a) a search using a gradient-based approach, and (b) a random search that randomly samples the search space. The illustration is adapted from [Maier et al. (2019)].

work of a gradient-based approach, which maximizes/minimizes a function using knowledge of its derivative, and a random search approach. As seen from Figure 3.9 (a), the gradient-based method, which is a classical optimization strategy, exhibits a high degree of exploitation and therefore, only searches locally in a certain region of the fitness landscape, but it does not have the ability to escape local optima. Random search strategy randomly samples the feasible space, as shown in Figure 3.9 (b), and exhibits a high degree of exploration, enabling larger areas of the fitness landscape to be searched, but does not have the ability to converge to good solutions.

In contrast, as shown in Figure 3.4 (c), EAs can converge to the global optimum on the same fitness landscape.

More generally, EAs are most appropriately applied when:

(a) the space being optimized defies closed-form mathematical descriptions such that more direct optimization techniques could be used;

(b) constraints or non-linearities in the solution space complicate the use of more direct optimization methods;

(c) the size of the search space to be explored is large and precluding the use of exhaustive (or random) search;

(d) exploration cost is high;

(e) the problem is multi-/many-objective;

A combination of any one or more of these factors may recommend the use of EAs techniques over more direct optimization methods such gradient-based methods [Sofge (2008)].

3. As an implication of previous point, EAs have a wider range of applicability compared with many more traditional optimization approaches [Maier et al. (2019)]. Consequently, EAs are successfully employed in different areas of real-world applications [Safi et al. (2018)].

4. EAs can satisfy to multiple competing objectives.

5. EAs are able to find solutions for problems in different dimensions of the search space.

6. EAs are useful to tackle problems that humans do not really know how to solve: i.e., the combination of control and design parameters of the model of Active Magnetic Regenerator (AMR), to ensure the best performance of the model. An EA, which is free of any human preconceptions or biases, can generate unanticipated solutions that can be comparable to, or better than, the best human-generated efforts.

As we can see, applying EAs for solving the optimization problems of simulation models provide a number of advantages over more conventional optimization methods. However, EAs have a number of challenges, especially related to computational efficiency of the models and the adjustment of their searching behaviour to the problems properties.

Main challenges of EAs:

1. EAs are potentially computationally intensive to find a global solution solution, although this primarily depends on the computational efficiency of the simulation model with which they are linked. Since EAs work with populations of solutions and generally, evolve better solutions over hundreds of generations, the number of calls of objective function(s) is mainly calculated as the product of the population size and the number of generations and is used as a metric to evaluate computing intensity. In this regard, the application of EAs to the problems of the AMR model is of particular concern, because of the following reasons:

   (a) the model run-time is in the order of hours (can be more than 24h for dual-operating model: the magnetic cooling system and the thermomagnetic energy generator);

   (b) the dimension of the search space can be large.

2. An EA efficient for finding the optimum of a single-objective optimization problem cannot be adequately applied to find multiple optimal solutions present in another optimization problem. To solve different kinds of problems, users need to know and/or to implement different algorithms, each specialized in solving a particular class of optimization problem [Deb & Tiwari (2008)].

   Since optimization problems of the AMR model and the model of Magneto Caloric Materials (MCMs) can appear with different numbers of objectives and of decision variables, this challenge is critical.

3. Generally, in accord with the No Free Lunch Theorem [Wolpert & Macready (1997)], EAs and all optimization algorithms need to be tuned to the properties of the optimization problem under consideration. It means that the searching behaviour

(i.e., the degree of exploration and exploitation) of EAs have to be adjusted to match the properties of the optimization problem. However, in practice, it is challenging, because the properties can be unknown.

4. Due to their stochastic nature, EAs never guarantee to find globally optimal solutions.

The presented advantages of EAs motivate us to use them for solving the optimization problems of the model of active magnetic regenerator and the model of the model of magneto caloric materials.

In the next section, we present and discuss several efficient optimization methods for overcoming the most important challenges.

3.2 Efficient Optimization Methods

In this section, we present several efficient optimization methods and discuss their applicability in the frame of our research objective. We focus on the methods for overcoming two mentioned challenges: (i) to reduce runtime for computationally intensive problems in Section 3.2.1 and (ii) to ensure scalability w.r.t. different dimensions of the objective (target) and search spaces in Section 3.2.2.

3.2.1 Methods for Computationally Intensive Problems

The challenge to solve computationally intensive problems, e.g, the model of Active Magnetic Regenerator (AMR) and the model of Magneto Caloric Materials (MCMs), can be overcome by using the following strategies:

1. **Simplification.** One of the most obvious way for reducing the computational intensity in terms of the number of function evaluations, is a re-formulation of a complex optimization problem to a simpler one: e.g. by reducing the search space, or by decreasing the number of objectives. Due to the fact that in this thesis, we aim at solving different problems w.r.t. the number of decision variables and the number of objectives, this method is not suitable.

2. **Surrogate-based approach.** Another commonly used method for reducing the search cost of computationally intensive objective function(s) is to employ a surrogate-based approach. The basic idea is to approximate a computationally intensive optimization function f by using one or several surrogate (approximate) models $\hat{f}$ of f that is faster to compute [Razavi et al. (2012)]. Such approach can reduce the number of calls of objective function(s) by one to two orders of magnitude [Timani & Peralta (2017)].

   In the context of computationally intensive optimization, $\hat{f}$ is considered as a computationally cheaper substitution of f. Therefore, during the optimization process the surrogate model $\hat{f}$ is learned according to a set of training points, i.e., pairs $(x, f(x))$, where f has been already evaluated. Then, $\hat{f}$ is applied to give hints about where promising candidate solutions are located or to replace the original f in order to be used for direct optimization.

Surrogate-based algorithms have to consider an exploration-exploitation trade-off: i.e., if a larger number of solutions are evaluated, using the original optimization function f, then a better quality of the surrogate model is implied, but, at the same time, the search cost is increased.

Various surrogate modeling techniques have been proposed to replace the expensive simulations or experiments by cheap surrogate models. The most popular techniques for surrogate model learning are: Radial Basis Functions [Orr et al. (1996)], Kriging also referred as Gaussian Processes regression [Kleijnen (2009)], Support Vector Machines [Herbrich et al. (1999)] and Artificial Neural Networks [Stanley et al. (2009)].

Although using a surrogate-based method seems to be an efficient way for reducing search cost, its application hides many traps and the following issues have to be taken under consideration:

(a) $\hat{f}$ has to be learned optimally for a surrogate-based algorithm.

(b) The computational complexity of the processes of learning and testing: e.g., the models of Radial Basis Functions and Gaussian Processes scale at least with $O(n^3)$, that limits their application only to low-dimensional optimization problems.

(c) Hyper-parameters, e.g., the number of training points, for surrogate learning procedure must be well chosen.

(d) Trade-off between accuracy and speedup. E.g., on multi-modal functions the advantages of using surrogate models are usually less significant due to difficult functions landscapes and limited number of training points together with high computational cost of surrogate learning.

(e) Different scenario for single- and multi- objective surrogate-based optimizations must be implemented and tested.

Taking the above mentioned issues into account, we conclude that an application of surrogate-based methods to reduce the search cost for solving the research problems of this thesis is intractable, because this method does not satisfy to the requirement of scalablility w.r.t. the dimensions of search and target spaces.

Thus, we assume that it is more advantageous to use an alternative surrogate-free method to reduce the search cost, which will lead to a comparable speedup, but with the required scalability.

3. **Parallel computing approach.** Alternatively, an approach to accelerate the optimization process is the use of parallel computing techniques for decreasing overall runtime [Falcón-Cardona et al. (2021)], [Talbi (2019)], [Sharma & Collet (2013)], [Tsutsui & Collet (2013)]. In the literature, is was demonstrated the relevance of parallel implementations of Multi-Objective Evolutionary Algorithms (MOEAs) in tackling computationally expensive real-world applications by providing considerable gains regarding speedup in terms of execution time and in some cases, performance in terms of accuracy [Falcón-Cardona et al. (2021)], [Talbi (2019)].

In the design of parallel MOEAs, three main parallel hierarchical models are identified [Talbi (2019)]:

(a) **Parallel algorithmic-level for MOEAs:** a parallel algorithm is composed of cooperating MOEAs, where the parallel cooperative work modify the behaviour of the MOEAs and allow to enhance not only search time, but also the quality of the obtained Pareto front. This model is not dependant on the Multi-objective Optimization Problem (MOP) to be solved.

(b) **Parallel iteration-level for MOEAs:** an algorithm handles in parallel a single iteration of evolutionary algorithms: e.g., the evaluation of population of solutions. The main goal is to speedup the search time of MOEAs manipulating large populations of solutions. This model is not dependant on the MOP to be solved and does not modify the behaviour of the MOEA.

(c) **Parallel solution-level for MOEAs:** an algorithm handles in parallel a single solution of the decision space. It consists in the parallel evaluation of the different objectives or constraints associated to the MOP. Consequently, this model is dependant on the MOP to be solved. This model does not modify the behaviour of the algorithm. It deals with improving the search time of the algorithm.

In this thesis, we apply the parallel iteration-level model, because of the following reasons:

(a) It is not dependant on the MOPs to be solved.

(b) It is recommended for the problems, which require a large population size [Falcón-Cardona et al. (2021)]: e.g., large scale optimization problems, many-objective optimization problems and difficult multi-modal problems.

(c) It does not modify the behaviour of the selected algorithm and ensures its theoretical computational complexity.

(d) It is the most popular, easy to be implemented and straightforward model [Talbi (2019)].

The basic idea of the parallel iteration-level model is to solve simultaneously a time consuming operation, i.e., the fitness assignment of the solutions composing the population, on different processing units. Once all of them have been completed, the results are employed for the next iteration (generation).

One of the most well known example is the master-slave model, which is also known as global parallelization: a single population is managed by a master and the evaluation of the objective functions is done simultaneously by the slaves. [Luna & Alba (2015)]. The master-slave model does not alter the search behaviour of the underlying sequential MOEAs, but instead, it makes them faster, especially when computationally expensive objective functions are tackled [Falcón-Cardona et al. (2021)].

In parallel iteration-level models, the solutions in population may be controlled in parallel according one from two strategies:

(a) **Synchronous:** i.e., all operations are handled in a synchronous way and then finalised before starting a new iteration. E.g., the main part of algorithm (i.e., master) has to wait for all results of executed in parallel evaluations of objective functions (i.e., slaves).

(b) **Asynchronous:** i.e., a new iteration can be handled by the master before all parallel parts finalize their process. Two queues of solutions are managed in parallel: the queue of solutions to be evaluated and the queue of already evaluated solutions. The solution of the first queue is handled by a free parallel part. Hence the main process handles the selection phase from the second queue without waiting the results from all parallel parts. This asynchronous model is not equivalent to the sequential one: there is no guarantee that the order of selecting and replacing the solutions is the same than in sequential algorithms.

Commonly, the parallel implemented MOEAs have a trade-off between achieving good execution times and producing high quality results. In [Falcón-Cardona et al. (2021)] it was shown that algorithms with asynchronous strategies which are faster than their synchronous counterparts, produce Pareto front approximations with less quality. There are some exceptions, e.g., in [Harada & Takadama (2017)], an adaptive semi-asynchronous communication strategy was proposed, which performs similar in terms of quality to synchronous algorithms, but maintaining high speedup.

However, in this thesis, we do not focus on high-performance parallel computing architectures, with which rise many questions about different topics: e.g., the switching mechanism between synchronous/asynchronous communication, shared or distributed memory, homogeneity or heterogeneity of machines, etc [Talbi (2019)], [Falcón-Cardona et al. (2021)]. Instead, applying the synchronous parallel iteration-level model for evaluating the objective functions, we concentrate our attention on the architecture of MOEA for parallel implementation itself, because of the following reasons:

(a) Since a MOEA with the parallel iteration-level model has to manipulate with large population size, it has to possess two following features [Falcón-Cardona et al. (2021)]:

   i. **Low computation complexity in a single generation** , which is not growing drastically with the increase of the number of objectives and the population size.

   ii. **Scalability w.r.t. the population size:** i.e., an ability of algorithms to approximate Pareto front with good accuracy (e.g., in terms of hypervolume) with increasing the population size and proportionally decreasing the number of generations.

   Unfortunately, many well-known state-of-the-art algorithms based on different techniques, e.g., domination-based, decomposition-based and performance indicator-based MOEAs, do not provide these two features. We discuss this topic in Section 4.1, where we explain our choice of an archive-based MOEA as a baseline technique for its implementation in parallel iteration level model.

(b) According to the most recent survey about parallel multi-objective evolutionary algorithms presented in [Falcón-Cardona et al. (2021)], most of the parallel MOEAs are based on a small number of baseline MOEAs such as Nondominated Sorting Genetic Algorithm II (NSGA-II) [Deb et al. (2000)] and Multi-objective Evolutionary Algorithm Based on Decomposition (MOEA-D) [Zhang & Li (2007)], which are limited by their properties: e.g., (i) NSGA-II is not suitable for many-objective optimization; (ii) the population size of

MOEA-D increases nonlinearly with the increase of the number of objectives and cannot be set arbitrarily, and it is difficult to choose a suitable decomposition method for different problems.

Consequently, a new multi- and many- objective optimization algorithm for parallel implementation with low computational complexity and the scalability w.r.t. the population size is required. In Chapter 4, we propose an archive-based MOEA for the parallel iteration-level model.

4. **Quantum-inspired approach.** Taking into account the quantum structure of the model of Magneto Caloric Materials (MCMs) presented in Section 2.2.2.1, we turn to quantum-inspired optimization algorithms for reducing the search cost. This can help not only to decrease the number of calls of the objective function comparatively to different EAs, but also to find the global optimum, due to its strong randomness and a high degree of intelligence. Quantum-inspired algorithms are discussed and presented in Section 3.3.

3.2.2 Methods for Unified Optimization

In this section we turn to the challenge of providing the scalability w.r.t. the dimensions of objective (target) space: i.e., the number of objectives.

As explained in the introduction of this thesis, we have to take into consideration the fact that in order to solve research problems, different optimization problems can be formulated for the model of Magneto Caloric Materials (MCMs) and the model of Active Magnetic Regenerator (AMR). It means that the problems can have different dimensions of search space (i.e., number of decision variables) and objective space (i.e., number of objectives). E.g., for reproducing the physical properties of MCMs, single- and multi-objective optimization problems can be formulated, depending on the studied material and its considered properties. Consequently, an optimization algorithm scalable w.r.t. the number of objectives is needed.

We apply the term "unified" borrowed from [Seada & Deb (2014)], to optimization algorithms, which are aimed at solving single-objective, multi-objective and many-objective problems with the relatively the same efficiency. The "No Free Lunch Theorem" [Wolpert & Macready (1997)] states that all algorithms have identically distributed performance across optimization functions, picked uniformly at random. It means that it is impossible to have a general algorithm to solve all optimization problems with the same efficiency. However, this theorem does not say that it is impossible to develop an algorithm scalable w.r.t. the number of objectives and efficient on a subcategory of problems, e.g., separable.

In the literature, there do not exist many studies, which propose an algorithm to unify single-, multi- and many-objective optimization together. An attempt to develop a unified algorithm is presented in [Deb & Tiwari (2008)], where the multi-objective Non-dominated Sorting Genetic Algorithm II (NSGA-II) [Deb et al. (2000)] is scaled down to solve single-objective problems. But it does not scale up to solve many-objective problems [Deb & Tiwari (2008)].

Thus, it seems logical to use a many-objective optimization algorithm as a baseline technique, e.g. NSGA-III, and to scale it down to solve single- and bi-objective problems. However, the scalable properties of NSGA-III present an issue and must be improved, because when the objective space becomes single-dimensional, the inherent guidance mech-

anism, which ensures diversity of population members in the objective space, disappeares. We will discuss the properties of NSGA-III in Section 3.4.5.

A possible method to improve the properties of an algorithm, is to apply hybridization. Hybridization is a method of combining two (or more) techniques such that the resulting algorithm contains the positive features of both (or all) the integrated algorithms [Thangaraj et al. (2011)]. Thus, the expectation from the hybridization is that the new algorithm combines the desirable properties of different algorithms such that it demonstrates the capability to solve a larger number of problems. Mainly, the existing hybrid algorithms aim at improving the exploration and exploitation capabilities of algorithms on different problems.

In practice, hybridization can be done in several ways [Thangaraj et al. (2011)]:

1. to alternate the processes of several techniques: e.g., initiate the algorithm with one technique and then apply the other technique on the final population obtained by the first technique;

2. to integrate the unique operators of a particular technique into the other technique: e.g., mutation and crossover operators of Genetic Algorithm can be used in Particle Swarm Optimization Algorithm;

3. to apply local search to improve the solution obtained by the global search;

4. to fuse the results of different algorithms, etc.

In the literature, many hybrid algorithms are largely presented. It should be noted that the Particle Swarm Algorithm (PSO) [Kennedy & Eberhart (1995)] algorithm is one of the most widely used algorithms in hybrid methods due to its simplicity and exploitation ability [Thangaraj et al. (2011)], [Mirjalili & Hashim (2010)]. In fact, the PSO algorithm is a strategy inspired on the social and cooperative behaviour shown by various species like flocks of bird or schools of fish; it is a population (swarm) based algorithm, where the potential solutions are called particles. These particles move through the search domain with a specified velocity in search of optimal solution. Each particle maintains a memory which helps it in keeping the track of its previous best position. The positions of the particles are distinguished as personal best and global best [Thangaraj et al. (2011)]. We do not use PSO algorithm in this thesis and thus, do not provide its description in details.

Below, we provide several examples for single- and multi-objective hybrid optimization algorithms:

1. Hybrid single-objective optimization algorithms:

   - In [Mirjalili & Hashim (2010)], two algorithms are combined: Particle Swarm Optimization (PSO) [Kennedy & Eberhart (1995)] and Gravitational Search Algorithms [Rashedi et al. (2009)]. It aims to integrate the exploitation ability of PSO with the exploration ability of Gravitational Search Algorithms.
   - In [Thangaraj et al. (2011)], many examples are presented, where PSO is employed.

2. Hybrid multi-objective optimization algorithms :

- In [Elhossini et al. (2010)], a new hybrid approach to multi-objective optimization of a swarm particles is proposed, where the fitness assignment technique of the Strength Pareto Evolutionary Algorithm II (SPEA-II) [Zitzler et al. (2001)] is applied to build a multi-objective PSO. The proposed approach presents a single framework that combines PSO and evolutionary operators: i.e., crossover and mutation. A single external archive is used to store non-dominant solutions for both algorithms. This approach allowed to build three hybrid algorithms, which alternate the processes of SPEA-II and PSO in different order.
- In [Tang & Wang (2012)], another multi-objective PSO-based hybrid approach is proposed to improve the robustness of evolutionary algorithms to solve different kinds of optimization problems by incorporating the definition of the position of particle of PSO with several crossover operators.
- In [Ibrahim, Rahnamayan, Martin & Deb (2017)] and [Ibrahim, Martin, Rahnamayan & Deb (2017)], the authors proposed the hybridization of population-based multi- and many-objective algorithms by fusing non-dominated fronts for extracting well-distributed solutions from a large set of non-dominated solutions collected during several runs of multiple algorithms. I.e., the fusion of solutions from multiple algorithms is employed to gain the combined benefits of several multi- and many-objective optimization algorithms, which helps to reduce the challenge of choosing one optimization.

  The main difference between the framework proposed in [Ibrahim, Martin, Rahnamayan & Deb (2017)] and in [Ibrahim, Rahnamayan, Martin & Deb (2017)] is that, in [Ibrahim, Rahnamayan, Martin & Deb (2017)], the fusion process occurs after the optimization process, whereas in the first one, the fusion of solutions is done during the optimization process.

Fusion-based algorithm for unified optimization

Despite the fact that the hybridization is used to solve different classes of problems, to the best of our knowledge, it has not been applied yet for unified optimization. However, we suppose that the challenge to provide the scalability w.r.t. the number of objectives and the number of decision variables can also be considered from the point of view of the hybridization. The algorithm described in [Ibrahim, Martin, Rahnamayan & Deb (2017)], presents for us the practical interest: it allows to solve many-objective problems by fusing the solutions from different techniques.

The working process of the fusion-based algorithm presented in [Ibrahim, Martin, Rahnamayan & Deb (2017)], includes three main steps:

1. the multiple algorithms are executed in parallel using the same population in order to determine best performing algorithms at every generation of the search process;
2. the fusion of solutions from different algorithms maintains the diversity of solutions;
3. the best performing algorithm is executed independently and continue generating improved candidate solutions.

The experimental results presented in [Ibrahim, Martin, Rahnamayan & Deb (2017)], proved that the fusion-based algorithm significantly outperforms all algorithms involved in the hybridization process in terms of diversity and convergence of obtained solutions.

The positive research results in [Ibrahim, Martin, Rahnamayan & Deb (2017)] motivate us to explore a fusion method for developing an optimization algorithm, which provides scalability w.r.t. the number of objectives and w.r.t. the number of decision variables: i.e. an algorithm capable to solve from single- to many-objective problems with different search space.

3.3 Quantum-inspired Optimization

Richard Feynman's proposal of a quantum computing system [Feynman (1982)], inspired by quantum mechanics in 1982, provided a way for future progress in the quantum computing area. Following up the progress, quantum-inspired optimization algorithms were proposed. Contrary to "true" quantum algorithms for real quantum hardware, e.g., Grover's search algorithm [Grover (1996)] and Shor's factorization algorithm [Shor (1994)], quantum-inspired algorithms do not require quantum hardware for their implementation and execution. However, they use quantum principles for improving performance: i.e., increasing speedup and accuracy. This approach allows developers to leverage the power of new quantum techniques today without waiting for quantum hardware, which is still an emerging industry.

In this section, we briefly describe the main directions in the development of quantum-inspired optimization algorithms with their advantages and disadvantages. Then, we evaluate their applicability to optimize the model of magneto caloric materials.

3.3.1 Quantum Theory

Given that the terminology from quantum physics is used in this thesis, we start by providing some essential terms of quantum mechanics on which quantum computing is based and then, we explain the general scheme of quantum algorithms. The provided theory, definitions and notations are based mainly on the following sources: [Greensite (2003)], [Abhijith et al. (2018)] and [DiVincenzo & Terhal (1998)].

3.3.1.1 Quantum Glossary

The current glossary consists of the notions required for a representation of the topic of this section.

Observable is something that one can measure about a system: e.g., position and momentum.

State is what will be found when a measurement of an observable is done. Quantum mechanics relies on a probabilistic notion of "state": knowing the state of a system is an equivalent to knowing the probability distribution for the measurements of all observables.

Ground state is the state of a quantum system with the lowest energy.

Stationary state is a quantum state with all observables independent of time.

Hamiltonian is an operator corresponding to the total energy of quantum system.

Wavefunction is a mathematical description of a quantum state of system.

Qubit is a unit of information in quantum computation, i.e., a quantum bit. Qubits have two basis states, 0 and 1, and can be in a simultaneous combination of these states.

Superposition is the ability of quantum particles to be simultaneously in all possible states. Consequently, a qubit represents a quantum particle in superposition of all possible

states. E.g., 4 bits are enough for a classical computer to represent any number between 0 and 15. However, with 4 qubits, a quantum computer can represent every number between 0 and 15 at the same time.

Quantum parallelism is a fundamental property of any quantum algorithm based on the quantum superposition principle. Computational quantum devices can perform multiple simultaneous evaluations of functions in a single time step.

Quantum measurement is the act of observing a quantum particle in superposition and resulting in one of the possible states.

Entanglement is the ability of several quantum particles in a group to correlate their measurement results with each other: i.e., the quantum state of each particle of the group cannot be described independently of the state of the others, including when the particles are separated by a large distance.

Interference is the intrinsic behaviour of a qubit due to superposition to influence the probability of it collapsing one way or another. In the same way as classical waves can reinforce or diminish each other, the wave functions of particle can reinforce or diminish itself.

Coherence is arising from quantum superposition and plays a central role in quantum mechanics, being a necessary condition for entanglement and other types of quantum correlations. It is the stability of the relative phase between quantum states: if the amplitude of the probability of the ground state 1 decreases, then the amplitude of the probability of the ground state 0 will increase. When a quantum system is measured, coherence disappears and the state of the system collapses to some ground state with a specific probability.

Coherence time, in context of quantum computing, is the time during which the computational process is carried out with controlled accuracy.

Decoherence is the loss of quantum coherence. It can be viewed as the loss of information from a system into the environment, since every system is loosely coupled with the energetic state of its surroundings. Another words, decoherence causes a quantum computer to lose two of its key properties: entanglement between the qubits and interference phenomena.

Quantum gate is an intrinsic quantum operation performed on qubits, that manipulates their quantum states and relate with the concept of classical logic gates.

Quantum circuit is a a computational routine, based on the similar concept of classical logic circuits: i.e., an ordered sequence of quantum operations (quantum gates) on quantum data (qubits), and measurements connected in a certain way to simulate and calculate a given function. The depth of a quantum circuit is defined as the number of gates connected sequentially.

3.3.1.2 General Scheme of a Quantum Algorithm

Now, since the most essential terms are presented, we can provide the general scheme of a quantum algorithm [Abhijith et al. (2018)]:

1. `Preparation phase:` A preparation of input qubits is performed, in which the initial values of the qubits are set in superposition.

2. `Calculation phase:` A sequence of quantum gates applied to this set of input qubit: i.e., the transformations are made on the qubits, trying to amplify the ex-

pected results and minimize the incorrect results, by taking advantage of quantum interference. In order to achieve this, usually, it is necessary to entangle the qubits. This phase can take several sequential steps.

3. `Measurement phase:` A measurement of the qubits is destroying the internal state of the measured qubit amplitudes and obtaining a classically interpretable result.

4. `Iterative phase:` To get the result statistically, the algorithm has to be iterated many times. Indeed, as the obtained result is probabilistic, it is necessary to run the algorithm again many times.

To summarize, the foundational core of quantum computing consists of three following rules: (i) to use wave-like properties of quantum particles to encode information, (ii) to store the information in quantum states of matter and (iii) to employ quantum gate operations to compute the information, by learning to "program" quantum interference.

3.3.2 Why Quantum-inspired Optimization Can Be Useful?

Now, when the essential terms are explained, we can discuss the reasons for the application of a quantum-inspired approach to solve the research problems of this thesis.

An application of a quantum physics approach for finding the optimal solution of an optimization problem, can result in the following potential benefits:

1. A speedup compared to classical sequential algorithms, because the computations take place in a highly parallel way: by using the qubit representation of information, many solutions are considered simultaneously as a superposition of the basis states.

2. Producing the global optimal solution: with interference, we can make transformations in the state so that the amplitudes of the incorrect solutions to the problem are subtracted and tend to zero and the amplitudes of the correct solutions are added and amplified so that the probability of obtaining such solutions are as high as possible.

However, quantum-inspired algorithms emulate/simulate quantum phenomena on classical hardware. For this reasons, quantum-inspired optimization algorithms, usually present a combination of quantum computing/physics and evolutionary optimization to achieve better heuristic optimization on classical hardware.

The research and development of quantum-inspired algorithms is carried out in three main directions: inspired by quantum computing, inspired by quantum physics and hybrid quantum-classical algorithms. In next subsections, we briefly discuss these algorithms.

3.3.3 Algorithms Inspired by Quantum Computing

The algorithms inspired by quantum computing are located at the intersection of two subareas of computer science: quantum computing and evolutionary computing.

The most well known algorithms for continuous optimization from this group are Quantum-inspired Evolutionary Algorithm [Li & Li (2008), da Cruz et al. (2010)] and Quantum-inspired Differential Evolution [Draa et al. (2004)].

The common significant features of these algorithms are:

1. A representation of solutions: i.e., a quantum population consists of probability distributions, instead of exact points in the search space. Uniform distributions or other families of distributions with random parameters can be used.

   E.g., a quantum inspired evolutionary algorithm described in [da Cruz et al. (2010)], represents each variable through a probability distribution function modeled by rectangular pulses. The centre of each pulse represents the mean of each variable, and the pulse height is the inverse of the domain length/N, where N is the number of pulses used to encode the variable. The quantum-inspiration comes from the constraint that the sum of the areas under the N pulses must equal 1: i.e., the pulses represent a superposition. Probability distributions are altered at each generation by updating the centres and widths of each pulse.

2. The genetic operators are adapted to the new representation. Two possible groups of new genetic operators are identified: (i) generalization of classical genetic operators to the new quantum representation and (ii) a new class of operators, modelling directly quantum mechanical phenomena: e.g., the rotations of quantum system state vectors.

Advantages:
Due to the probabilistic representation of each solution through a superposition of multiple states, such kind of algorithms have an additional element of randomness, which can allow them to achieve better population diversity and potentially, can be useful for many real-world problems with vague nature.

Disadvantages:
Our preliminary investigation have not revealed any advantages of the algorithms inspired by quantum computing over classical genetic algorithms for solving continuous optimization problems.

Furthermore, a number of important questions are ignored in all of these works:

1. How can the massive parallelism of quantum computing offer some benefits, especially, for evaluating objective function in a large population of solutions, which is the computational bottleneck for computationally intensive functions?

2. How can entanglement correlations between the solutions and their values of objective function (with the evaluation performed in between) be maintained?

3. How to implement mutation and crossover operators in quantum algorithms? It is not clear how to perform crossover operations such that entangled states will constructively interfere with one another.

Even though this approach is called quantum-inspired, the connection with quantum computing is minimal and actually is more akin to Estimation of Distribution Algorithms reviewed in [Larranaga (2002)].

3.3.4 Algorithms Inspired by Quantum Physics

The optimization algorithms of this group are based on mixed techniques of evolutionary optimization and quantum physics, where a quantum system behavior is simulated in

order to find the approximate global optimum. The main feature of these algorithms is that they are considered as a probabilistic system, where the probabilities, related to each state, describe the behaviour of the system.

The most well known algorithms from this group are Quantum-behaved Particle Swarm Optimization Algorithm (QPSO) [Sun, Feng & Xu (2004)], Multi-scale Quantum Harmonic Oscillator Algorithm (MQHOA) [Wang et al. (2018)], Quantum-inspired evolutionary Salp Swarm Algorithm [Chen et al. (2019)] and Adiabatic Quantum-inspired Optimization Algorithm [Pastorello & Blanzieri (2019)].

The MQHOA and QPSO algorithms transform a given optimization problem into the solution of the Schrödinger equation by describing the optimal distribution of the objective function through the wave function. The optimal solution of the problem corresponds the minimum energy of a quantum system, which is estimated by the objective function. Since the QPSO algorithm is used in this thesis, its detailed description is provided in Section 3.4.4.

The Adiabatic Quantum-inspired Optimization is the most "quantum" algorithm and emulates quantum tunneling effect by employing the adiabatic theorem. The adiabatic theorem states that, as long as a transformation of quantum mechanical system happens slowly enough, this system has time to adapt its functional form and will stay in that lowest energy configuration [Bornemann (1997)]. When the transformations are done, the optimization problem is solved.

Advantages:

Quantum physics inspired algorithms have a more "quantum" nature comparatively with quantum computing inspired algorithms. Thus, they can help to understand different quantum features in the optimization process in general way, without taking into account the features of quantum hardware.

These algorithms emulate a quantum process on classical hardware, which can perform better than state-of-the-art classical optimization techniques. E.g., in materials design, *OTI Lumionics*[2] has used the adiabatic-based optimization algorithm provided by *Azure Quantum*[3] to achieve more accurate simulations for computational chemistry.

Disadvantages:

Quantum mechanics is hard to simulate on classical hardware. Unlike classical probability theory, many configurations of the quantum state, which can be potentially observed, may interfere with each other like waves. This interference prevents the use of statistical sampling to obtain the quantum state configurations. Rather, every possible configuration of a quantum system, in which it could be, has to be tracked, in order to understand the quantum evolution.

3.3.5 Hybrid Quantum-Classical Algorithms

Before discussing the hybrid quantum-classical algorithms, it is necessary to briefly present the currently existing quantum hardware.

[2] `https://otilumionics.com/`

[3] `https://azure.microsoft.com/en-us/services/quantum/#product-overview`

3.3.5.1 Noisy Intermediate Scale Quantum Computers

We currently live in a world with Noisy Intermediate Scale Quantum (NISQ) computers. The term "intermediate scale" defines the size of quantum computers: i.e., the number of available qubits, which is around 50-120 qubits. The lower bound; i.e., 50 qubits, defines the maximum size of a quantum computer that can still be modeled on a classical supercomputer. The term "noisy" means that the key problem of current quantum computers that is limiting their use, is outside interference caused by the external thermal, magnetic and electromagnetic fields. The quantum computers of the NISQ era should be seen as a step towards more powerful quantum computing technology.

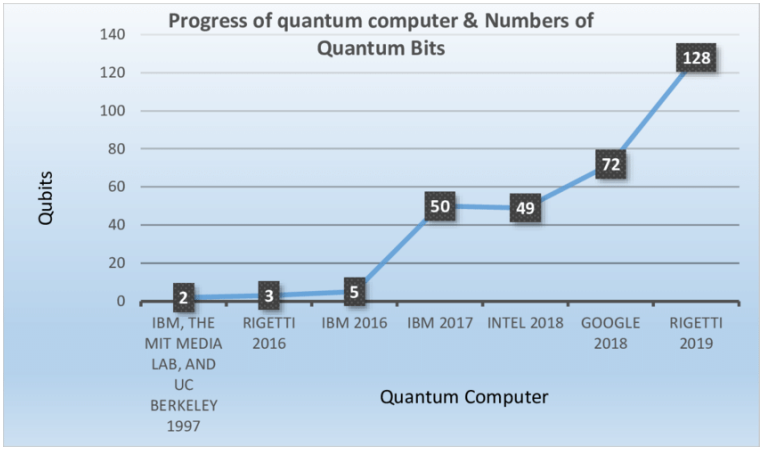


Figure 3.10: Quantum computer progress. The illustration is adapted from [Abohashima et al. (2020)].

Recent progress of quantum computers is illustrated in Figure 3.10. Many of NISQ devices are on the cloud and ready to use. They are different regarding to the number of qubits, coherence time and their set of physically implemented gates for quantum circuits. The most well-known open-source quantum software projects are the Microsoft Quantum Development Kit[4], the IBM Quantum Experience (with a software API called Qiskit)[5] and the Rigetti (Forest[6] and Cloud Computing Services[7]).

However, as mentioned above, current quantum hardware is noisy: i.e., non-durable to work with, because, by nature, qubits are fragile. They require a precise environment and state to operate correctly, and they are highly prone to outside interference. This interference is referred to as noise, which is a consistent challenge and a well-known reality of quantum computing: as the number of qubits increases, the noise level increases too. So there is a limitation on the number of qubits in one circuit and the size of the circuit imposes limits on the computational power of NISQ technology. As a result, error correction plays a significant role.

[4] https://azure.microsoft.com/ru-ru/resources/development-kit/quantum-computing/
[5] https://www.research.ibm.com/quantum-computing/
[6] https://www.rigetti.com/forest
[7] https://www.rigetti.com/qcs

3.3.5.2 Hybrid Quantum-Classical Optimization Algorithms

Along with the development and improvement of NISQ technology, hybrid quantum-classical algorithms are being developed. The aim of these algorithms is to find an approximate solution to an optimization problem on quantum hardware, as well as to provide error correction on classical hardware and, as a result, increase the computing power of NISQ devices. The most well known hybrid quantum-classical algorithms are the Variational Quantum Eigensolver (VQE) [Peruzzo et al. (2014)] and the Quantum Approximate Optimization Algorithm (QAOA) [Farhi et al. (2014)], [Verdon et al. (2019)], which have already been employed to solve optimization problems in the fields of quantum chemistry and materials [McArdle et al. (2020)], [Cao et al. (2019)].

VQE and QAOA are designed to solve optimization problems that can be cast in the form of finding the ground state energy of quantum system on quantum hardware. In order to do this, the objective function, which represents the quantity needed to be optimized, has to be mapped onto a relatively simple Ising-type Hamiltonian. Thus, the ground state energy is the smallest eigenvalue of the Hamiltonian. Then, the Hamiltonian has to be transformed in form of a parameterized quantum circuit in order to be run on quantum hardware. The circuit is executed at a specific set of parameters and the expectation value is computed. Based on the result, a classical optimization algorithm is employed to correct the errors of the parameters of this quantum circuit, which control the preparation of a quantum state, by iteratively changing them, to reduce the expectation value. Then a quantum computer prepares that state and calculates its properties. The application of a classical optimization algorithm makes it possible to increase the depth of the quantum circuit, and, consequently, to increase the computing power of a quantum computer without loss of stability and accuracy, which makes these algorithms attractive for quantum computers of the NISQ era. The general schema of the VQE algorithm is presented in Figure 3.11, where the quantum circuit is presented on the left side.

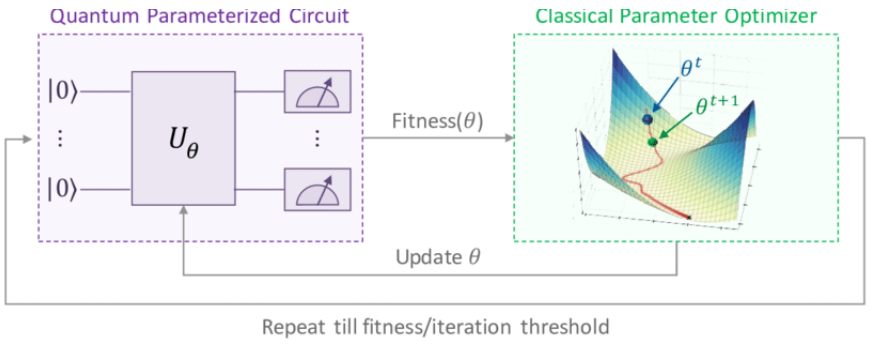


Figure 3.11: Variational Quantum Eigensolver algorithm.

Advantages:

The great advantage of hybrid quantum-classical algorithms is that they allow users to partially implement an optimization process on real quantum hardware by executing the quantum circuits, which implement the objective function and consequently, use real quantum features to obtain a quantum advantage: e.g., speedup from the physical superposition.

Disadvantages:

One of the main issues of the hybrid quantum-classical algorithms is the limited capability of current NISQ, due to decoherence and high error rate of quantum gates.

Next, the robustness of classical optimization algorithms is a critical importance for successful applications of hybrid algorithms on current NISQ devices: the quality of the solution produced by these algorithms for a given problem depends on the quality of the variational parameters of the circuit found by the classical optimizer. Consequently, developing such kind of algorithms can require a prohibitive amount of domain expertise: e.g., evolutionary optimization algorithms experts may find themselves unfamiliar with the concepts used in quantum physics and vice versa, while both aspects have a crucial role to play in developing algorithms for NISQ devices.

Finally, it is difficult to translate each optimization problem to a quantum circuit. Moreover, it is not obvious how well the hybrid quantum-classical algorithms will scale with the size of the different problems.

3.3.6 Discussion

Because all quantum-inspired optimization algorithms are heuristics, they not guarantee to find the optimal solution. Also, these algorithms do not always outperform other optimization techniques. In reality, it depends on the problem, and discovering what makes quantum-inspired optimization perform better than other methods in some situations and not others is still an active area of research.

Implementing optimization algorithms on a real quantum hardware seems to be a very promising field due to the stochastic nature of optimization algorithms. Furthermore, from a quantum computing point of view, the Hamiltonian model of Magneto Caloric Materials (MCMs) presented in Section 2.2.2.1, can be translated to a quantum circuit and executed on quantum hardware. However, this Hamiltonian model simulates around 4000 atoms and requires a too large number of qubits in order to be executed on the NISQ devices, because of the errors in the quantum operations.

If the qubits and their control were ideal, the computational power of quantum devices with a couple hundred qubits would already dwarf that of any classical computer and could show quantum advantage. Thus, in the future, the Hamiltonian model of MCMs should be implemented on real quantum hardware, when more universal quantum hardware will be available. Consequently, in this thesis, due to the existing hardware limitations, we currently turn to quantum-inspired algorithms, which are executed only on classical computers.

The algorithms inspired from quantum computing have been under active development between 2003 and 2014. But with the beginning of the NISQ era and since the first presentation of the hybrid quantum-classical optimization algorithm, the progress of research in the field of quantum computing inspired algorithms has drastically decreased. This fact can be explained by their inefficiency comparatively with hybrid algorithms and their disadvantages, which was discussed in the section about the algorithms inspired by quantum computing.

However, applying quantum-inspired optimization to real-world problems can offer new insights and solutions. Thus, in this thesis, we take into account the advantages of the algorithms based on quantum physics and investigate them due to their usefulness for theoretically understanding different quantum features in optimization process in general

way, without taking into account the features of quantum hardware.

3.4 State-of-the-art Algorithms and Methods Employed in the Thesis

In this section, we give the descriptions of the state-of-the-art algorithms and methods, which are used in this thesis.

First, in Section 3.4.1, we describe Archive-based Stochastic Ranking Evolutionary Algorithm (ASREA), which is selected as an efficient multi-objective optimization method for parallel implementation in order to solve time-consuming problems of the model of Active Magnetic Regenerator (AMR). Taking into consideration the features of the optimization tasks of the AMR model, ASREA is improved in Section 4.

Then, we provide the descriptions of two techniques, which are employed in the quantum-inspired optimization algorithm proposed in Section 5.1: the Diffusion quantum Monte Carlo (DMC) method and (1+1)-Evolution Strategy in Section 3.4.2 and Section 3.4.3 respectively.

Finally, we detail the descriptions of Quantum-inspired Particle Swarm Optimization algorithm (QPSO) and Non-dominated Sorting Genetic Algorithm III (NSGA-III) in Section 3.4.4 and Section 3.4.5, respectively, which are employed in the fusion-based algorithm presented in Section 5.2.

3.4.1 Archive-based Stochastic Ranking Evolutionary Algorithm

Archive-based Stochastic Ranking Evolutionary Algorithm (ASREA) is a Multi-Objective Evolutionary Algorithm (MOEA), which seems to be a good candidate for a massively parallel implementation because of its low computational complexity for the worst case for a single generation. This low complexity is explained by the procedure of stochastic ranking assignment, which ranks the population by comparing individuals with members of an archive.

In this thesis, we improve ASREA and present a new version, called FastEMO, in Chapter 4, for solving in parallel time-consuming continuous multi- and many-objective problems of the model of Active Magnetic Regenerator (AMR).

Below, we provide the summary of the original ASREA from the paper [Sharma & Collet (2010*a*)].

3.4.1.1 Algorithm

ASREA uses a regular population $\boldsymbol{P}$ of size n and an archive $\boldsymbol{A}$ of size a. The role of $\boldsymbol{A}$ is to store the best non-dominated solutions, i.e., the current Pareto front, and keep this front as wide as possible.

Starting with a random initial population $\boldsymbol{P}^{t=0}$ and an empty archive $\boldsymbol{A}^{t=0}$, ASREA evaluates each solution $\boldsymbol{p}^{t=0} \in \mathbb{R}^{\mathrm{d}}$ (where d is the dimension of search space). Then, the copies of all non-dominated members of $\boldsymbol{P}^{t=0}$ are duplicated into $\boldsymbol{A}^{t=0}$. If the number of non-dominated solutions is larger than the size of archive a, then the diverse non-dominated solutions are reduced according to the crowding distance operator from NSGA-II [Deb et al. (2000)].

Then, the steps presented in Algorithm 1 are performed for each generation t. The notations used in Algorithm 1 are summed up in Table 3.1.

Table 3.1: Notations of ASREA Algorithm.

Notation	Explication	Value
t	current generation	$t \in \mathbb{N}_+$
T	maximum number of generations	$T \in \mathbb{N}_+$
m	dimension of target space	$m \in \mathbb{N}_+$
$\boldsymbol{p}^t$	solution	$\boldsymbol{p}^t \in \boldsymbol{P}^t, \boldsymbol{p}^t \in \mathbb{R}^{\mathrm{d}}$
$\boldsymbol{P}^t$	regular population	$(\boldsymbol{p}_1^t, \boldsymbol{p}_2^t, ..., \boldsymbol{p}_n^t)$
$\boldsymbol{x'}^t$	child solution after SBX operator	$\boldsymbol{x}^t \in \mathbb{R}^{\mathrm{d}}$
$\boldsymbol{x}^t$	child solution	$\boldsymbol{x}^t \in \boldsymbol{X}^t, \boldsymbol{x}^t \in \mathbb{R}^{\mathrm{d}}$
d	dimension of search space ($size(\boldsymbol{p}^t) = size(\boldsymbol{x}^t)$)	$d \in \mathbb{N}_+$
$\boldsymbol{X}^t$	child population	$(\boldsymbol{x}_1^t, \boldsymbol{x}_2^t, ..., \boldsymbol{x}_n^t)$
$\boldsymbol{A}^t$	external archive of non-dominated solutions	$(\boldsymbol{a}_1^t, \boldsymbol{a}_2^t, ..., \boldsymbol{a}_m^t)$
n	population size ($size(\boldsymbol{P}^t) = size(\boldsymbol{X}^t)$)	$n \in \mathbb{N}_+$
a	archive size ($size(\boldsymbol{A}^t)$)	$a \in \mathbb{N}_+$

Each new generation starts by checking whether the stopping criterion $t \geq T$ (where T- the maximum number of generations) is satisfied. When it is the case then ASREA finishes its work by returning the Pareto Front.

3.4.1.2 Stochastic Ranking Assignment

As mentioned above, the feature of ASREA is not only to try to propagate the good solutions to the next generation, but also to reduce the ranking complexity of NDS, which is done by stochastic ranking procedure. The rank of a new solution $\boldsymbol{x}_k^t$ from the child population $\boldsymbol{X}^t$ is assigned while comparing it with the archive population by the following rule of dominance rank:

$$rank(\boldsymbol{x}_k^t) = 1 + number\ of\ \boldsymbol{a}^t\ from\ \boldsymbol{A}^t\ that\ dominate\ \boldsymbol{x}_k^t \tag{3.4}$$

According to Equation 3.4, a lower rank is better. Comparatively with ranking operator of such algorithms like NSGA-II and SPEA-2, the ranking operator of ASREA is not deterministic, where the rank of an individual is assigned while comparing it with only a small number of solutions of the archive population. These feature not only allows to propagate the good solutions to the next generation, but also reduces the ranking complexity of algorithm.

3.4.1.3 Archive update operator

The archive gets updated during this ranking process. The archive size is bounded by the interval $[0, a]$. Although the archive update operator is detailed in [Sharma & Collet (2010a)], we briefly describe it below, due to its key role in ASREA. If solution $\boldsymbol{x}_k^t$ gets rank 1, it will join to $\boldsymbol{A}^{t+1}$ depending on two cases:

Algorithm 1: ASREA: Pseudo-code of procedure at generation t.

Input: $\boldsymbol{P}^t$
Result: $\boldsymbol{P}^{t+1}$
for $k = 0; k < n; k = k + 1$ **do**
  Select randomly parent solutions $\boldsymbol{p}_i^t$ and $\boldsymbol{p}_j^t$ from $\boldsymbol{P}^t$
  $\boldsymbol{x'}_k^t = \text{SBX}(\boldsymbol{p}_i^t, \boldsymbol{p}_j^t)$ // Apply SBX Crossover
  $\boldsymbol{x}_k^t = \text{Mut}_{POLY}(\boldsymbol{x'}_k^t)$ // Apply Polynomial Mutation
  EVAL($\boldsymbol{x}_k^t$) //Evaluate $\boldsymbol{x}_k^t$
end
for $k = 0; k < n; k = k + 1$ **do**
  RANK($\boldsymbol{x}_k^t, \boldsymbol{A}^t$) // Pareto dominance based stochastic ranking of child $\boldsymbol{x}_k^t$ w.r.t. $\boldsymbol{A}^t$ (see Section 3.4.1.2)
  CD($\boldsymbol{x}_k^t, \boldsymbol{A}^t$) // Target-based crowding distance assignment of child $\boldsymbol{x}_k^t$
  $\boldsymbol{A}^t = \text{UPDATE}(\boldsymbol{x}_k^t, \boldsymbol{A}^t, a)$ // Archive update by rank and crowding distance (see Section 3.4.1.3)
end
$\boldsymbol{P}^{t+1} = \text{SELECT}(\boldsymbol{X}^t, \boldsymbol{A}^t)$// Selection strategy to new population (see Section 3.4.1.4)

1. If $\boldsymbol{x}_k^t$ dominates one or several members of $\boldsymbol{A}^t$ and is non-dominated w.r.t. the rest of them, then $\boldsymbol{x}_k^t$ replaces one of the dominated individuals of $\boldsymbol{A}^t$.

2. If $\boldsymbol{x}_k^t$ is non-dominated w.r.t. all the members of $\boldsymbol{A}^t$ and is distinct from them, then $\boldsymbol{x}_k^t$ goes in $\boldsymbol{A}^{t+1}$. If the size of $\boldsymbol{A}^{t+1} = a + 1$, then in order to reduce the current size value of $\boldsymbol{A}^{t+1}$, the objective-wise crowding distance operator [Deb et al. (2000)], [Sharma & Collet (2010*a*)] is applied for all members of $\boldsymbol{A}^{t+1}$ including $\boldsymbol{x}_k^t$. The extreme solutions in each objective are kept and the worst individual by crowding distance value is replaced by $\boldsymbol{x}_k^t$, so that the archive size stays in the interval $[0, a]$.

3.4.1.4 Selection strategy

The presented below selection strategy is used for the creation of the next generation. The selection strategy of ASREA works on the following way: 50% of $\boldsymbol{P}^{t+1}$ is filled from individuals of $\boldsymbol{A}^{t+1}$ and the rest from the current $\boldsymbol{X}^t$, using a binary tournament selection based on calculated rank and crowding distance operator.

3.4.1.5 Variation Operators

At each iteration, ASREA creates a child population through two operators: Simulated Binary crossover (SBX) [Deb et al. (1995)] and Polynomial mutation [Deb & Deb (2014)]. Here, we do not provide their descriptions, because they are well-known and commonly used variation operators and they do not present the key role in this work.

3.4.1.6 Summary of ASREA

Baseline Techniques of ASREA:

1. **stochastic ranking assignment** based on Pareto-dominance principle: Non-Dominated Sorting (NDS) procedure w.r.t. solutions of the archive (see Section 3.4.1.2);

2. **classical objective space-based crowding distance** procedure for preserving diversity [Deb et al. (2000)];

3. **archive update operator** w.r.t. the Pareto dominance-based ranks and the classical crowding distance values of solutions (see Section 3.4.1.3);

4. **selection strategy** to fill the parent population for next generation. It based on the Pareto dominance-based ranks and the classical crowding distance values of solutions and incorporates strong elitism (see Section 3.4.1.4).

Main Feature: ASREA uses an external set of non-dominated solutions not only to store good solutions and propagate them to the next generation (as used in many archive-based methods: e.g., SPEA-II [Zitzler et al. (2001)]), but also to reduce the computational complexity of the algorithm for a single generation. ASREA is a really "cheap" Multi-Objective Evolutionary Algorithm (MOEA), due to the ranking assignment procedure w.r.t. the solutions of small-sized archive.

Computational Complexity for the Worst Case at Single Generation:
$\mathcal{O}(man)+\mathcal{O}(mn\log(n))$, where: (i) $\mathcal{O}(man)$ is the computation complexity of ranking assignment; (ii) $\mathcal{O}(mn\log(n))$ is classical target space-based crowding distance; (iii) m is the number of objectives, a is the size of the archive and n is the population size.

Archive Size: $a = 10 \cdot m$ is the recommended size in the original paper [Sharma & Collet (2010*a*)], which is proved there on bi- and three-objectives problems with a small population size $n = 100$. However, the impact of the archive size on the population scalability of ASREA was not studied in [Sharma & Collet (2010*a*)].

Advantages:

1. low computational complexity for the worst case at a single generation;

2. simple and easily modified structure.

Limitations: In this thesis, we investigate the limitations of ASREA w.r.t. possible particularities of optimization problems of the AMR model. Their analysis is presented and discussed in Section 4.1.3.

3.4.2 The Diffusion Monte Carlo Method

Monte Carlo methods are statistical techniques, which were developed for estimating integrals that could not be evaluated analytically. The term "quantum Monte Carlo" covers a class of methods based on random sampling, which are able to simulate quantum systems and compute the electronic ground state of atoms, molecules and solids. Among quantum Monte Carlo methods presented in the literature [Toulouse et al. (2016)], the Diffusion quantum Monte Carlo (DMC) method is one of the most valuable computational tools

to accurately predict the ground state properties of a quantum many-body Hamiltonian that describe various quantum systems: e.g., electron gases, electrons in atoms, molecules and solids, quantum fluids, nuclear matter and ultra-cold atoms [Foulkes et al. (2001)].

3.4.2.1 Overview of the DMC Method

The presented description of the Diffusion quantum Monte Carlo (DMC) method is a compilation from the following references: [Foulkes et al. (2001)], [Kent (1999)], [Kosztin et al. (1996)], [Toulouse et al. (2016)], [Zen et al. (2016)]. The equations and terms presented in this section, correspond to the mathematical formulations used in the mentioned literature.

The DMC method is a stochastic projector method for solving the many-body Schrödinger equation in imaginary time. The time-dependent Schrödinger equation in imaginary time $\tau = it$ for a system on N particles is defined as follows (units are dimensionless) [Kent (1999)]:

$$\frac{\partial|\Psi\rangle}{\partial\tau} = -\hat{H}|\Psi\rangle \tag{3.5}$$

where $\hat{H}$ is a given Hamiltonian: i.e., a sum of kinetic and potential energy of the considered quantum system;
$|\Psi\rangle$ is a N-body wave function and is expanded in eigenstates of the Hamiltonian;
τ is the imaginary-time evolution operator ($t \to -i\tau$), which just replaces the real time t (so called Wick rotation of time) in order to transform the oscillatory behaviour of the wave function into exponential behaviour.

If the Hamiltonian is separated into kinetic energy and potential terms, the time-depended Schrödinger equation presented in Equation 3.5 takes on a form similar to a diffusion equation presented below [Zen et al. (2016)], [Kosztin et al. (1996)]:

$$-\frac{\partial\Psi(\mathbf{R},\tau)}{\partial\tau} = -\frac{1}{2}\nabla^2\Psi(\mathbf{R},\tau) - (E_T - V_P(\mathbf{R}))\Psi(\mathbf{R},\tau) \tag{3.6}$$

where $\mathbf{R} = (\mathbf{r}_1, \ldots, \mathbf{r}_N)$ specifies the positions of N particles;
$\mathbf{r}_{i,(i\in[1,N])}$ is the i-th particle, called a "walker", which presents a set of coordinates;
$\Psi(\mathbf{R},\tau)$ is a wave function, which depends on coordinates $\mathbf{R}$ and time τ;
V_P is the potential energy;
E_T is an energy offset: i.e., an introduced normalization factor through a shift of energy scale, which has to be adjusted to stabilize the simulation.

A formal solution of the imaginary time Schrödinger equation presented in Equation 3.6, is presented in Equation 3.7, in which the time dependence is given by a phase factor $e^{-(\hat{H}-E_\mathrm{T})\delta\tau}$ [Kent (1999)], [Kosztin et al. (1996)]:

$$|\Psi(\tau+\delta\tau)\rangle = e^{-(\hat{H}-E_\mathrm{T})\delta\tau}|\Psi(\tau)\rangle \tag{3.7}$$

where the $|\Psi\rangle$ state evolves from imaginary time τ to a later time $\tau + \delta\tau$;
the initial state ($|\Psi(\tau)\rangle$) is expanded in energy ordered eigenstates.

Hence, any initial state, $\Psi(\mathbf{R}, 0)$, that is not orthogonal to the ground state, will evolve to the ground state in a long time limit [Toulouse et al. (2016)], [Kent (1999)]:

$$\lim_{\tau\to\infty}|\Psi(\tau)\rangle = e^{-(E_0-E_\mathrm{T})\tau}\,|\Psi_0\rangle\,\langle\Psi_0|\Psi\rangle \tag{3.8}$$

where E_0 is the energy of ground state and $|\Psi_0\rangle$ is the wave function of the ground state.

According to Equation 3.8 the following asymptotic behaviour for $\tau \to \infty$ can be defined [Kosztin et al. (1996)] as follows:

1. if $E_\mathrm{T} > E_0$, $\lim_{\tau\to\infty} |\Psi(\tau)\rangle = \infty$, the wave function diverges exponentially fast;
2. if $E_\mathrm{T} < E_0$, $\lim_{\tau\to\infty} |\Psi(\tau)\rangle = 0$, the wave function vanishes exponentially fast;
3. if $E_\mathrm{T} = E_0$, the exponential $e^{-(E_0-E_\mathrm{T})\tau}$ can be eliminated and we then obtain that $\Psi(\tau)$ becomes proportional to the wave function of the ground state (Ψ_0) [Toulouse et al. (2016)]:

$$\lim_{\tau\to\infty} \big|\Psi(\tau)\big\rangle \propto \ |\Psi_0\rangle \tag{3.9}$$

The described behaviour provides the basis of the DMC method: i.e., for $E_\mathrm{T} = E_0$, the wave function in its position representation ($\Psi(\mathbf{R}, \tau)$) converges to the ground state wave function Ψ_0 regardless of the choice of the initial wave function $\Psi(\mathbf{R}, 0)$, as long as there is a numerically significant overlap between $\Psi(\mathbf{R}, \tau)$ and Ψ_0.

Now we have an equation, which allows us to find the ground state of any given Hamiltonian by propagating it forward in time and adjusting E_T appropriately. So, we need a way to integrate Equation 3.6 for an arbitrary energy offset (E_T) and initial wave function ($\Psi(\mathbf{R}, 0)$).

In position representation, Equation 3.7 is written as a convolution integral with a special function called the Green function and presented as follows [Zen et al. (2016)], [Toulouse et al. (2016)]:

$$\Psi(\mathbf{R}_\mathrm{f}, \tau + \delta\tau) = \int \mathrm{d}\mathbf{R}_\mathrm{i}\, G(\mathbf{R}_\mathrm{f} \leftarrow \mathbf{R}_\mathrm{i}; \delta\tau)\Psi(\mathbf{R}_\mathrm{i}, \tau) \tag{3.10}$$

where $\mathbf{R}_i$ is a starting points; $\mathbf{R}_\mathrm{f}$ is a new points;
$G(\mathbf{R}_\mathrm{f} \leftarrow \mathbf{R}_\mathrm{i}; \delta\tau)$ is the Green function: i.e., the imaginary-time propagator from $\mathbf{R}_\mathrm{i}$ to $\mathbf{R}_\mathrm{f}$ and defined in Equation 3.11:

$$G(\mathbf{R}_\mathrm{f} \leftarrow \mathbf{R}_\mathrm{i}; \delta\tau) = \langle \mathbf{R}_\mathrm{f}|\, e^{-(\hat{H}-E_\mathrm{T})\delta\tau}\, |\mathbf{R}_\mathrm{i}\rangle \tag{3.11}$$

The Green function prescribes how to propagate further in time the distribution of the wave function [Foulkes et al. (2001)]. More precisely, the exact Green function is a solution of the Schrödinger equation and is, therefore, unknown for realistic systems [Zen et al. (2016)], [Foulkes et al. (2001)], [Kent (1999)]. However, for a small enough time step $\delta\tau \to 0$, the Green function can be approximated by using the Trotter-Suzuki formula [Toulouse et al. (2016)]:

$$e^{-\left(\hat{T}+\hat{V}\right)\tau} = e^{-\hat{V}\tau/2} e^{-\hat{T}\tau} e^{-\hat{V}\tau/2} + O(\tau^3) \tag{3.12}$$

where $\hat{T}$ and $\hat{V}$ are the kinetic and potential energy operators.

Thus, by considering a small time steps $\delta\tau = \tau/n$ (where n is the number of steps), and using Equation 3.10 to write $\Psi(\mathbf{R}, \tau_i)$ in terms of $\Psi(\mathbf{R}, \tau_{i-1})$, with $i = 1, \ldots, n$ and $\tau_i = \tau_{i-1} + \delta\tau$, we obtain the following expression for the Green function [Zen et al. (2016)]:

$$G(\mathbf{R}_\mathrm{f} \leftarrow \mathbf{R}_\mathrm{i}; \tau) = \int G(\mathbf{R}_\mathrm{f} \leftarrow \mathbf{R}_1; \delta\tau) \ldots G(\mathbf{R}_{n-1} \leftarrow \mathbf{R}_\mathrm{i}, \delta\tau) d\mathbf{R}_1 \ldots d\mathbf{R}_{n-1} \tag{3.13}$$

In position representation, this approximation leads to the following expression [Toulouse et al. (2016)], [Zen et al. (2016)]:

$$G(\mathbf{R}_\mathrm{f} \leftarrow \mathbf{R}_\mathrm{i}; \tau) \approx P(\mathbf{R}_\mathrm{f} \leftarrow \mathbf{R}_\mathrm{i}) W(\mathbf{R}_\mathrm{f} \leftarrow \mathbf{R}_\mathrm{i}) \tag{3.14}$$

where $P(\mathbf{R}_\mathrm{f} \leftarrow \mathbf{R}_\mathrm{i})$ is a diffusion term related to the kinetic energy: i.e., a Gaussian probability distribution of particles centred around $\mathbf{R}_\mathrm{i}$ with standard deviation $\sqrt{\delta\tau}$;
$W(\mathbf{R}_\mathrm{f} \leftarrow \mathbf{R}_\mathrm{i})$ is a branching term: i.e., a weight function, which uses $(V_P(\mathbf{R}) - E_T)$ as a rate that describes a potential-dependent increase or decrease in particle density and determines the number of walkers that survive to the next step.

$$P(\mathbf{R}_\mathrm{f} \leftarrow \mathbf{R}_\mathrm{i}, \tau) = \frac{1}{(2\pi\delta\tau)^{3N/2}} e^{-\frac{(\mathbf{R}_\mathrm{f}-\mathbf{R}_\mathrm{i})^2}{2\delta\tau}} \tag{3.15}$$

where N is a current number of particles;
$\delta\tau$ is a time step.

$$W(\mathbf{R}_\mathrm{f} \leftarrow \mathbf{R}_\mathrm{i}, \tau) = e^{-\left(\frac{V_P(\mathbf{R}_\mathrm{f})+V_P(\mathbf{R}_\mathrm{i})}{2} - E_\mathrm{T}\right)\delta\tau} \tag{3.16}$$

where $V_P(\mathbf{R})$ is the potential energy.

As we can see from Equation 3.14, the Green function is now composed of a Gaussian probability distribution function, where the mean $\mathbf{R}_i$ spreads with time as $\sqrt{\delta\tau}$, and a rate term that grows or shrinks depending on the value of local energy relative to E_T. Hence, the Green function has the expected behaviour of a diffusion-like term multiplied by a branching term. In this case, the DMC method interprets the solving of the Schrödinger equation as a product of probabilities and weights to be modelled by a series of sequential stochastic processes. As the number of particles increases, the dimensionality of the integral increases as well, since we have to integrate over all coordinates $\mathbf{r}$ of all particles $\mathbf{R}$.

To make it clear, the process described by Equation 3.6 and solved in terms of the Green function, is shown schematically in Figure 3.12 for a one-dimension problem, where a single particle is confined by a potential well $V(x)$ [Foulkes et al. (2001)]. The initial walker distribution samples a uniform Ψ_{init}, which is represented by the circles in Figure 3.12. As the imaginary-time propagation proceeds, the distribution converges towards a distribution representative of the ground state wave function, Ψ_0: i.e., the distribution gradually evolves by a process of diffusion and branching to Ψ_0. Note that where the potential energy is low, walkers tend to branch giving a higher density of walkers, and where the potential energy is high, walkers tend to be removed.

3.4.2.2 Simulation

There are numerous methods to carry out the simulation, which are presented in the literature, e.g., in [Toulouse et al. (2016)], [Kent (1999)] and [Kosztin et al. (1996)]. In this thesis, we follow the method presented in [Kosztin et al. (1996)], and provide its description below, using the terms and notations from the original paper.

The simulation procedure presented in [Kosztin et al. (1996)], is selected because it is easy for implementation. As explained in [Kosztin et al. (1996)], this procedure provides the possibility to readjust the value of E_T after each time step and to follow the time evolution of the system for as many time steps as are needed to converge to the ground

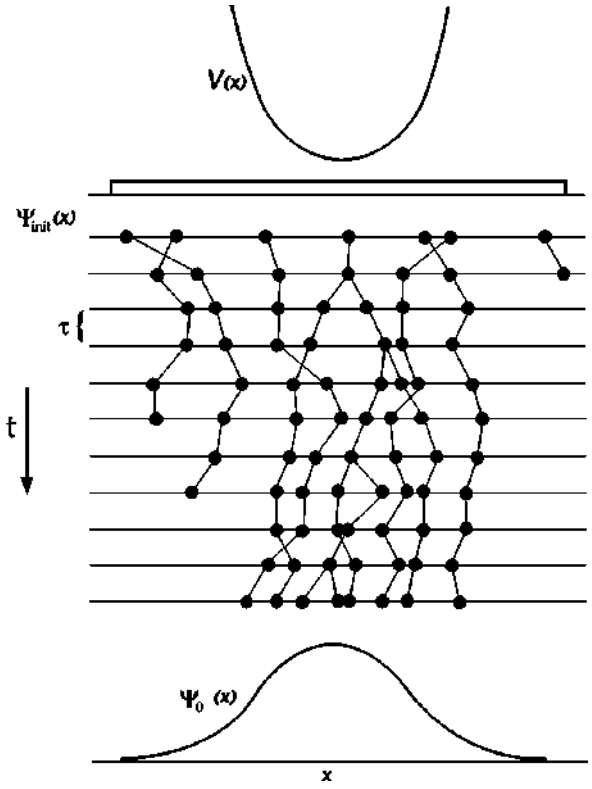


Figure 3.12: The evolution of walkers in the DMC method. The illustration is adapted from [Foulkes et al. (2001)].

state wave function and energy as a product of probabilities and weights to be modelled by a series of sequential stochastic processes.

Below, we summarize the steps taken in [Kosztin et al. (1996)] for simulation of the DMC method in order to obtain the ground state (the minimum value) energy of a given quantum system and its wave function. Schematically, these steps are depicted in Figure 3.13 and can be described as follow:

1. at the start of the simulation $\tau = 0$:

   (a) `initialization of particle positions (R):` $N^{(\tau=0)}$ particles, which in the originally paper are referred to as "replicas", are distributed in space to make up a initial wave function $\Psi(\mathbf{R}, 0)$;

   (b) `initialization of the offset energy (`E_T`):` the offset energy $E_T^{(\tau=0)}$ is set to be the average potential of the initial particles, which is calculated as follows:

$$E_T^{\tau} \;=\; \langle V_P \rangle^{\tau} \;=\; \frac{1}{\mathcal{N}^{\tau}} \sum_{j=1}^{\mathcal{N}^{\tau}} V_P(\mathbf{r}_j^{(\tau)}) \tag{3.17}$$

   where $\mathcal{N}^{\tau}$ is the current number of particles.

2. at each iteration of the simulation τ:

   (a) `increment of time:` the time τ is advanced by a small time step $\delta\tau$;

(b) `walk:` a diffusive displacement process, in which the positions of each particle $\mathbf{r}_i^{(\tau+1)}$ are determined according to Equation 3.18:

$$\mathbf{r}_i^{(\tau+1)} \;=\; \mathbf{r}_i^{(\tau)} \;+\; \sqrt{\delta\tau}\cdot\boldsymbol{\rho}^{(\tau+1)} \tag{3.18}$$

where $\boldsymbol{\rho}^{(\tau+1)}$ is a vector of Gaussian random numbers with zero mean and a variance equal to one.

Since each particle is moved, its potential ($V_P(\mathbf{r_i})$) is calculated and recorded.

(c) `branch:` it consists of two steps:

i. `birth/death process`, where the particles are replicated/removed from the distribution after each time step $\delta\tau$ with a probability proportional to $W(\mathbf{R})$ calculated by Equation 3.16. It means that each i-th particle is replaced by a number of particles m_i according to Equation 3.19 [Kosztin et al. (1996)]:

$$m_i \;=\; \min\left[\,\mathrm{int}[W(\mathbf{r}_i) \;+\; \mathcal{U}(0,1)\right],\, 3] \tag{3.19}$$

where $\mathrm{int}[W(\mathbf{r})]$ denotes the integer part of $W(\mathbf{r})$ and $\mathcal{U}(0,1)$ represents a random number uniformly distributed in the interval $[0,1]$.

Following Equation 3.19, the following three scenarios of birth/death process are possible:

A. if $m_i = 0$, the particle is deleted and its diffusion process is stopped; this is referred to as the "death" of a particle.

B. if $m_i = 1$, the particle is unaffected and it goes to the next iteration for further diffusion displacement;

C. if $m_i = 2, 3$, the particle also goes to the next iteration, but is cloned into new "replicas" (if $m_i = 3$ the two new "replicas" are created). This is referred to as the "birth" of a particle.

ii. `an adaptation of the offset energy` (E_T), where the value of E_T is redefined from the newly arranged particles through the following equation [Kosztin et al. (1996)]:

$$E_R^{(\tau+1)} = \langle V_P^{(\tau+1)}\rangle + \alpha\left(1 - \mathcal{N}^{(\tau)}/\mathcal{N}^{(\tau=0)}\right)\,, \tag{3.20}$$

where $\mathcal{N}^{(\tau)}$ is the current number of particles after the birth/death process;

$\langle V_P^{(\tau)}\rangle$ is the average potential, which is calculated by Equation 3.17;

$\alpha \in \mathbb{R}_+$ is a "feedback" parameter, which is chosen empirically for each individual problem so as to reduce as much as possible the statistical fluctuations in $\mathcal{N}^{(\tau=0)}$ and, at the same time, to diminish unwanted correlations between the successive generation of "replicas". The suggested value of α proposed in [Kosztin et al. (1996)] is $1/\delta\tau$.

According to [Kosztin et al. (1996)], Equation 3.20 should be regarded as an empirical result rather than an exact one.

(d) `count:` calculate the ground state energy and ground state wave function, when the system has already reached its stationary state identified through a converged $\langle V\rangle_\tau$.

The spatial interval $(\mathbf{r}_{min}, \mathbf{r}_{max})$ is divided equally into n_b "boxes" (sub-intervals) for each degree of freedom and the distribution of "replicas" among these "boxes" is counted by employing standard numerical methods [Kosztin et al. (1996)].

Because the counting process starts after time t_{max}, when the stationary state has been reached, the particles sample one and the same wave function at any subsequent time, and the cumulative counting of the particles in the "boxes" can be used to enhance the effective number of "replicas" by a factor of τ_{max} (number of time steps the counting is done).

Once the spatial distribution of replicas is known, the distribution can be normalized for obtaining the ground state wave function by the following equation:

$$\Psi_0(\mathbf{r}_i) \approx \frac{N_i}{\sqrt{\sum_{i=1}^{n_b} N_i^2}} , \quad i = 1, \ldots, n_b . \tag{3.21}$$

3. `output:` returns the results of the simulation:

   (a) the average value of the offset energy $\langle E_T \rangle \approx E_0$, which equals the ground state energy and is calculated during the second part of the simulation: i.e., when the system is already stabilized.

   The value of the average reference energy after n time steps is defined through the following equation [Kosztin et al. (1996)]:

$$\langle E_T(\tau = n\delta\tau) \rangle \; = \; \frac{1}{n} \sum_{i=1}^{n} E_T(i\delta\tau) \tag{3.22}$$

   (b) the ground state wave function, defined as a normalized spatial distribution of the "replicas".

Once the Schrödinger equation is solved, several properties of a given system may be calculated from the wave function. Errors or approximations made in obtaining the wave function will be manifest in any property derived from the wave function.

Remarks

The following details have to be precised, because they specify some features of the method proposed in [Kosztin et al. (1996)]:

1. there is a limitation on the birth rate of the particles equals 3 in the branch step: i.e., never more than two copies are created, which is seen from the description of the branch step.

   It is necessary in order to avoid numerical instabilities, especially at the beginning of the Monte Carlo simulation, when E_T may differ significantly from E_0. It means that the limited number of "replicas" helps to prevent the uncontrolled growth of the number of particles, which can occur when all the particles are located very close to each other and consequently, a large values m can arise. Such a situation can result in an overflow of the number of particles, which the wave function is be able to accommodate, due to its finite size;

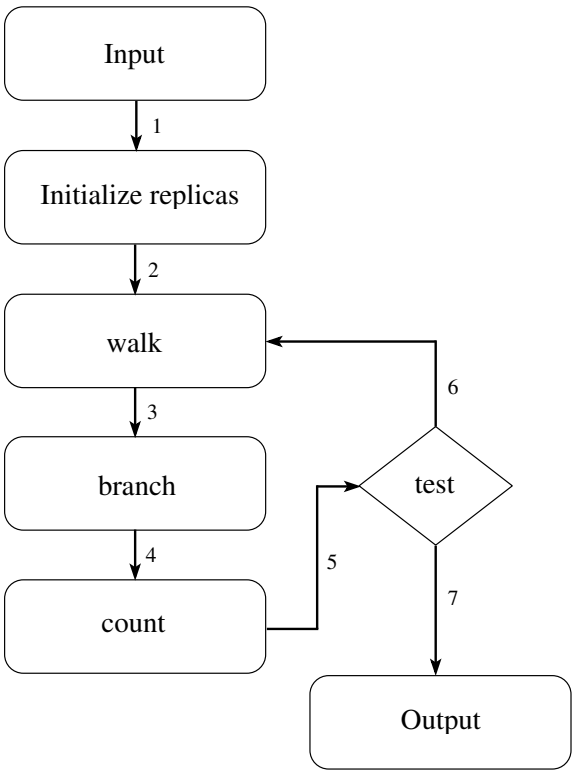


Figure 3.13: Diagram of the DMC simulation. The illustration is adapted from [Kosztin et al. (1996)].

2. the error resulting from the limitation $m \leq 3$ is expected to be small, because a sufficiently small time step is used;

3. the DMC simulation process terminates if the value m for all particles in the branch process equals zero. According to [Kosztin et al. (1996)], to avoid this possibility, one needs to choose the initial location of the replicas with care. In general, any point where the ground state wave function is large is a good choice.

4. the `walk` and `branch` steps are called $2\tau_{max}$ times while the `count` step is called only τ_{max} times: i.e., during the second half of the calculation, when the system has reached its stationary state.

Importance sampling

The DMC method can be significantly improved by resorting to a technique called importance sampling [Kosztin et al. (1996)], which is detailed in the following works: [Foulkes et al. (2001)], [Kent (1999)], [Kosztin et al. (1996)] and [Toulouse et al. (2016)].

The basic idea of this method is to change the probability distribution of the particles in a controlled way by replacing a given sampling distribution with a different distribution that is more efficient and provides a lower variance to the evaluation without changing the expectation value.

This can be achieved by reformulating Equation 3.6 such that the resulting equation has a solution $\Psi(\mathbf{R}, \tau)$ multiplied by an approximation of the ground state wave function ($\Psi_G(\mathbf{R})$), and the latter can be obtained by a variation method. This introduced

approximation of the ground state wave function is called a trial or guiding wave function $\Psi_G(\mathbf{R})$. Consequently, a new distribution is defined as follows:

$$f(\mathbf{R}, \tau) = \Psi_G(\mathbf{R})\Psi(\mathbf{R}, \tau) \tag{3.23}$$

The equation for non-importance sampled DMC presented in Equation 3.6, is consequently reformulated as follows [Kent (1999)], [Zen et al. (2016)]:

$$-\frac{\partial f(\mathbf{R}, \tau)}{\partial \tau} = -\frac{1}{2}\nabla^2 f(\mathbf{R}, \tau) + \nabla \cdot [\mathbf{V}(\mathbf{R}) f(\mathbf{R}, \tau)] - S(\mathbf{R}) f(\mathbf{R}, \tau) \tag{3.24}$$

where $\mathbf{V}(\mathbf{R}) \equiv \nabla \log \left|\psi_G(\mathbf{R})\right|$ is the additional drift velocity (so-called "quantum force");

$S(\mathbf{R}) \equiv E_T - E_L(\mathbf{R})$ is the branching term;

$E_L(\mathbf{R}) = \psi_G(\mathbf{R})^{-1}\hat{H}\psi_G(\mathbf{R})$ is the local energy computed w.r.t. the guiding wave function $\Psi_G(\mathbf{R})$;

E_T is a trial energy introduced to maintain the normalisation of the projected solution for a large τ: initially, E_T is chosen to be the energy of the guiding wave function obtained by a variational method (e.g., Variational Monte Carlo), and is updated during the simulation progresses. The right side of the equation of the importance sampled DMC (Equation 3.24) consists of the diffusion, the drift and the rate terms (from left to right) [Kent (1999)].

The application of the simulation method to this new equation (see Equation 3.24) makes the particles propagate ahead in the time branching-drift-diffusion process and yields "replicas" which spend more time in "important" regions of the configuration space where the wave function $\Psi(\mathbf{R}, \tau)$ is expected to be large.

The potential dependent rate term of the non-importance sampled method is replaced by a term, which depends on the difference between the local energy and the trial energy. Thus, according to [Kent (1999)], [Zen et al. (2016)], [Kosztin et al. (1996)], the DMC method using importance sampling should be substantially more efficient than unguided DMC, due to use of an optimised guiding function, which helps to minimise the difference between the local and trial energies, and hence to minimise fluctuations in the distribution f.

Advantages of importance sampling over simple sampling:

1. concentrating the sampling in the important part of the phase space [Zen et al. (2016)], [Kent (1999)], [Kosztin et al. (1996)];

2. the branching term depends on the local energy $E_L(\mathbf{R})$, and not on the potential energy $V_P(\mathbf{R})$. Since $E_L(\mathbf{R})$ is much smother than $V_P(\mathbf{R})$, the stability of the DMC simulation is greatly enhanced [Zen et al. (2016)].

3.4.2.3 Summary

The Diffusion quantum Monte Carlo (DMC) method can be summarized as follows:

1. a wave function Ψ is represented numerically as a distribution of particles (walkers), which can be thought of as points in the space on a definite position;

2. these particles are propagated through imaginary time, dying off or being created at each iteration depending on their position, their potential energy, and the average potential of all of the particles, which is interpreted as the energy of the system;

3. the offset energy E_T is adjusted at each iteration to control the number of particles;

4. after the defined number of iterations, the average of many successive measurements of E_T will approach the actual E_0 of the system, and the distribution of the superposition of the particles of many successive iterations will converge to the ground state wave function.

3.4.3 (1+1)-Evolution Strategy

In general, Evolution Strategies (ESs) are optimization methods inspired by the principles of biological evolution, which implement a repeated process of stochastic variations followed by selection [Hansen et al. (2015)]: in each generation, new candidate solutions (children) are generated from their parents, their fitness is evaluated, and the better candidate are selected to become the parents for the next generation. ESs most commonly address the problem of continuous black-box optimization [Hansen et al. (2015)]. According to [Hansen et al. (2015)], they have two main design principles: i.e., non-bias and adaptive control of parameters of the sample distribution.

(1+1)-ES originally proposed in [Rechenberg (1973)], is an elitist selection scheme with one parent and one child, which maybe is the simplest method, which belongs to the class of ESs.

In this thesis, we use the (1+1)-ES with the success-based step-size adaptation (1/5 rule) presented below, as a reinforcement technique for the Diffusion quantim Monte Carlo (DMC) method that is implemented in the quantum-inspired optimization algorithm proposed in Section 5.1.

Below, we briefly recall the original algorithm of (1+1)-Evolution Strategy (ES). The provided description is mainly based on the work of Hansen et al. (2015).

Algorithm

The simple variant of (1+1)-ES with 1/5 rule as a success-based step-size adaptation is given in Algorithm 2.

In Algorithm 2, we consider the search space $\mathbb{R}^{\mathrm{d}}$, where d is the dimension of the search space (the number of decision variables). In each generation $t \in \mathbb{N}_+$, the state of the (1+1)-ES is given by $(m^{(t)}, \sigma^{(t)})$, where $m \in \mathbb{R}^{\mathrm{d}}$ is a single parent individual and $\sigma \in \mathbb{R}_+$ is a step size. It samples one candidate offspring $x \in \mathbb{R}^{\mathrm{d}}$ in d-dimensional search space per generation t from the isotropic normal distribution $x^{(t)} \sim (m^{(t)}, \left(\sigma^{(t)}\right)^2 I)$ and applies (1+1)-selection, i.e., it keeps the better of the two points – an elitist selection scheme with one parent and one offspring. Thus the parent is replaced by the successful offspring, meaning that the offspring must perform at least as good as the parent to replace it. Here, $I \in \mathbb{R}^{\mathrm{d}\times\mathrm{d}}$ denotes the identity matrix. Using the identity matrix means that the variations of all variables are uncorrelated. The standard deviation $\sigma > 0$ of the sampling distribution, so called step size, has an impact on the performance [Hansen et al. (2015)].

In accordance with the original approach presented in [Rechenberg (1973)] and [Hansen et al. (2015)], we chose a simple mechanism of immediate step size adaptation based on

"success" or "failure" of each sample, so called the 1/5 success rule, which maintains a stable distribution of the success rate, concentrated around 1/5.

More precisely, the 1/5 success rule implements the following principle, which consists of two conditions:

1. the step size should be increased if many steps are successful, indicating that the search is too local;

2. the step size should be decreased if small number of steps are successful, indicating that the step size used for sampling solutions is too large.

The constants $c_- < 0$ and $c_+ > 0$ in Algorithm 2 control the change of $\ln(\sigma)$ in case of failure and success, respectively. They are given parameters of the rule. For $c_+ + 4 \cdot c_- = 0$ we obtain an implementation of the classic 1/5-rule [Rechenberg (1973)]. We call $\frac{c_-}{c_- - c_+}$ the target success probability of the algorithm, which is always assumed to be strictly less than 1/2. This is equivalent to $c_+ > -c_-$. A reasonable parameter setting is $c_-, c_+ \in \Omega\left(\frac{1}{d}\right)$. According to this scheme, σ increases if the success probability is larger than 1/5, and decreases if the success probability is smaller than 1/5.

Algorithm 2: (1+1)-ES.

input $m^{(0)} \in \mathrm{R}^d$, $\sigma^{(0)} > 0$
given parameters $c_+ > 0$, $c_- < 0$
$t \leftarrow 0$
repeat
  $\left(z^{(t)}\right) \sim \mathcal{N}(0, I)$
  $x^{(t)} \leftarrow m^{(t)} + \sigma^{(t)} \cdot z^{(t)}$
  if $f\left(x^{(t)}\right) \leq f\left(m^{(t)}\right)$ **then**
    $m^{(t+1)} \leftarrow x^{(t)}$
    $\sigma^{(t+1)} \leftarrow \sigma^{(t)} \cdot e^{c_+}$
  else
    $m^{(t+1)} \leftarrow m^{(t)}$
    $\sigma^{(t+1)} \leftarrow \sigma^{(t)} \cdot e^{c_-}$
  $t \leftarrow t + 1$
until *stopping criterion is met*;

Properties:

The (1+1)-ES with 1/5 success rule has two main properties:

1. it performs an elitist selection, ensuring that the best-so-far solution is never lost and the sequence $f(m(t))$ is monotonically decreasing;

2. the step-size control depends on the concept of the success probability, which is the probability of a sampled point outperforming the parent in the search distribution centre.

Advantages:

1. (1+1)-ES with the 1/5 success rule of step size adaptation has a very simple structure;
2. generally, (1+1)-ES can be applicable to solve black-box optimization problems;
3. the (1+1)-selection scheme tends to work well on uni-modal functions;
4. the uncorrelated mutation is efficient on separable functions.

Limitations:

1. applying uncorrelated mutations limits the success of (1+1)-ES on non-linear, non-separable functions;
2. preservation of the best solution (elitism) limits its application on multi-modal and dynamic functions, because of premature convergence.

3.4.4 Quantum-inspired Particle Swarm Optimization (QPSO)

The Quantum-inspired Particle Swarm Optimization (QPSO) algorithm firstly presented in [Sun, Xu & Feng (2004)], is a heuristic optimization technique that eploits the concept of quantum particle motion for reaching the optimal solution.

In this thesis, we use QPSO in the following way:

1. as a well-known state-of-the-art algorithm for a comparative study with the quantum-inspired optimization algorithm provided in Section 5.1;
2. as a part of the fusion-based algorithm presented in Section 5.2.1, which aims at ensuring an invariant property w.r.t. the number of objectives.

The description of QPSO provided in this section, is mainly based on the sources that provide the explanation and analyse of the standard QPSO algorithm: i.e., [Sun, Xu & Feng (2004)], [Sun, Feng & Xu (2004)] and [Sun et al. (2007)].

Working Principle of QPSO

Quantum-inspired Particle Swarm Optimization (QPSO) considers particles moving in a bounded potential field and endows them with a wave behaviour. It is assumed that particles are attracted to a quantum potential field which is centred on their local attractors $\mathbf{p}$. Under the given context, the quantum state of a particle with momentum and energy is depicted by its wave function $\Psi(\mathbf{R}, t)$, where $\mathbf{R} = (\mathbf{r}_1, \ldots, \mathbf{r}_M)$ specifies the coordinates of the M particles and a particular value of $\mathbf{R}$, $\mathbf{r}_i$ $(i \in [1, M])$, presents a set of N coordinates for each of these M particles. The wave function is used instead of the conventional representation of a state of a particle by its position and velocity, because in quantum mechanics, it is not possible to determine simultaneously both the position and velocity of a particle with arbitrary accurate precision (due to the uncertainty principle of Heisenberg). It means that we can only obtain the probability of a particle appearing in a certain position from the probability density function ($|\Psi(\mathbf{R}, t)|^2$), whose form depends on the potential field in which the particle is moving. Thus, the position of one particle can be measured by a Monte Carlo method by solving the time-dependent Schrödinger equation (presented in Equation 3.6).

Analogies of QPSO

The following analogies between quantum physics and classical optimization algorithms are used:

1. each candidate solution is represented by a particle position ($\mathbf{r}_i$), the quantum state of which is described by the wave function w.r.t. the selected potential field model;

2. the decision variables of a solution candidate are encoded into the position coordinates of a particle;

3. a performance index of each particle position is defined as an objective function ($f(x)$) of a given optimization problem;

4. the memory attribute of each particle refers to the ability to save the best personal position of the particle ($\mathbf{r}_{ibest}$) by comparing its actual position with the position afer the motion, as it is defined in Equation 3.25;

$$\mathbf{r}_{ibest}^{(t+1)} = \begin{cases} \mathbf{r}_{ibest}^{(t)}, & f\left(\mathbf{r}_i^{(t+1)}\right) \geq f\left(\mathbf{r}_{ibest}^{(t)}\right) \\ \mathbf{r}_i^{(t+1)}, & f\left(\mathbf{r}_i^{(t+1)}\right) \leq f\left(\mathbf{r}_{ibest}^{(t)}\right) \end{cases} \tag{3.25}$$

5. the communication attribute of each particle refers to the ability to save the particle with the best global position ($\mathbf{r}_{gbest}$) among the swarm, which is calculated according to Equation 3.26. Each particle $\mathbf{r}_i$ shares information with other particles through the best position $\mathbf{r}_{gbest}$ seen so far in the swarm history and follows its trajectory toward the global optimum based on quantum mechanics described above.

$$\mathbf{r}_{gbest}^{(t+1)} = \arg\min_{\mathbf{r}_{i,pbest}} f\left(\mathbf{r}_{ibest}^{(t+1)}\right) \tag{3.26}$$

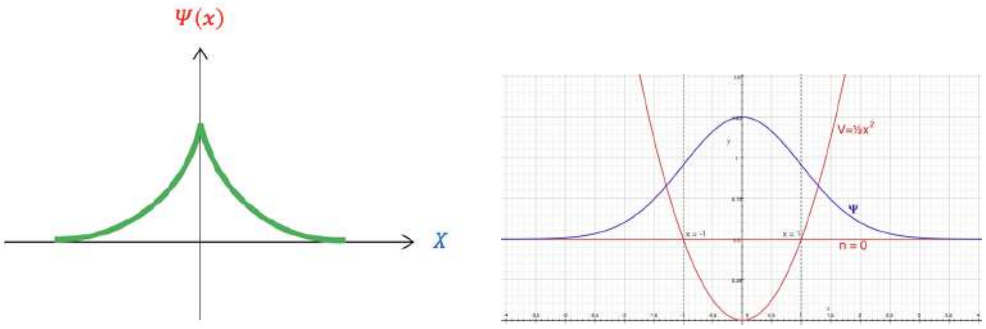
Potential Field Model

The choice of the potential field model is very important and has a strong impact on the performance of the algorithm. There are many potential field models in quantum mechanics. The delta potential well and quantum oscillator are two paradigms of such potential fields. In [Sun, Xu & Feng (2004)],[Sun, Feng & Xu (2004)] and [Sun et al. (2007)], the Delta potential well model is used, assuming to achieve a better convergence. The comparison of the models of quantum harmonic oscillator and Delta potential well are provided in [Sun, Feng & Xu (2004)] and [Sun et al. (2007)], where the differences between the probability density functions in both cases are shown.

More precisely, as seen from Figure 3.14, the curve of the quantum harmonic oscillator model is relatively narrower than that of the Delta potential well, which means that the particle converges much faster in the system of the quantum harmonic oscillator model than in the system of the Delta potential well model with the same parameter control method.

However, such convergence speed is acquired in sacrifice of the global search ability of QPSO: i.e., according to the results of the tests presented in [Sun, Feng & Xu (2004)], the oscillator model of QPSO is prone to premature convergence.

Some other potential field models have wave functions too complex to be simulated or can be inefficient [Sun, Feng & Xu (2004)]. In more recent work [Alvarez-Alvarado et al.



a) in Delta potential well model b) in Harmonic oscillator model

Figure 3.14: Probability density functions associated with the model of potential fields. The illustration is adapted from [Sun, Feng & Xu (2004)].

(2021)], some other potential field models are also employed in QPSO: Lorentz, Rosen-Morse, and Coulomblike Square Root. The experiments presented in [Alvarez-Alvarado et al. (2021)], show a significant increase in computation time, because of simulations of their complex wave functions. Moreover, it is demonstrated in [Alvarez-Alvarado et al. (2021)] that among these three models, no one shows the best performance in all the given attributes taken under consideration: i.e., exploration, exploitation, and simulation time.

Consequently, in our work, we follow [Sun et al. (2007)] and select the standard model of the Delta potential well.

Quantum State of a Particle

According to the selected model, the wave function is considered as the state of a particle in the Delta potential well and is defined as follows:

$$\Psi(\mathbf{r}_i, t+1) = \frac{1}{\sqrt{(\mathbf{L}_i{}^{(t)})}} \exp\left\{\left(-\frac{2 \cdot |\mathbf{p}_i^{(t)} - \mathbf{r}_i^{(t)}|}{\mathbf{L}_i{}^{(t)}}\right)\right\} \tag{3.27}$$

where $\mathbf{L}_i^{(t)}$ is the width of the Delta potential well in which all particles move, and its value determines the search space for each particle in each generation;

$\mathbf{p}_i^{(t)}$ is the local attractor, which is a vector of coordinates of a point in the centre of potential well;

$\mathbf{r}_i(t)$ is the vector of coordinates of the i-th particle.

The local attractor $\mathbf{p}_i$ is interpreted as the centre of gravity toward which the i-th particle careens while its kinetic energy declines. If the search space is stationary, i.e., which is the case in most of the practical applications, there will be no periodic orbits also known as unstable equilibrium in the search hyperspace. The local attractor of the i-th particle denoted by $\mathbf{p}_i$ is computed using Equation 3.28:

$$\mathbf{p}_i^{(t)} = \frac{\left(\mathbf{c}_p \times \mathbf{r}_{ibest}^{(t)} + \mathbf{c}_g \times \mathbf{r}_{gbest}^{(t)}\right)}{c_p + c_g} \tag{3.28}$$

where $\mathbf{c}_g$ and $\mathbf{c}_p$ are sequences of random numbers uniformly distributed on [0, 1];

$\mathbf{r}_{ibest}$ is the coordinate vector of the best personal position of i-th particle;

$\mathbf{r}_{gbest}$ is the coordinate vector of the best global position.

Knowing the wave function $\Psi(\mathbf{r}_i, t+1)$, we can get the probability density function that the particle $\mathbf{r}_i^{(t)}$ appears at position $\mathbf{r}_i^{(t+1)}$ relative to $\mathbf{p}^{(t)}$ according to the following equation [Sun, Xu & Feng (2004)]:

$$|\Psi(\mathbf{r}_i^{(t+1)})|^2 = \frac{1}{\mathbf{L}_i^{(t)}} \exp\left\{\left(-\frac{2 \cdot |\mathbf{p}_i^{(t)} - \mathbf{r}_i^{(t)}|}{\mathbf{L}_i^{(}t)}\right)\right\} \tag{3.29}$$

Measurement

The probability density function of a particle position defined in Equation 3.29, depicts the state of this particle in a "quantized" search space, not informing us of any certain information about the position of a particle that is vital to evaluate the fitness of a particle. Therefore, a state transformation between the two spaces: i.e., the quantum search space and the classical solution state, is absolutely necessary. In terms of quantum mechanics, the transformation from quantum state to classical state is called collapse, which is the measurement of the position of particle. It means that we have to measure the position of the particle, which is called collapsing the quantum state to the classical state. A Monte Carlo method can simulate the measurement process. By this simulation procedure, the coordinates of position of a particle $\mathbf{r}_i \in \mathbf{R}$ can be obtained as follows [Sun et al. (2007)]:

$$\mathbf{r}_i^{(t+1)} = \mathbf{p}_i^{(t)} \pm \frac{\mathbf{L}_i^{(t)}}{2} \cdot ln\Big(\frac{1}{\mathcal{U}[0,1]}\Big) \tag{3.30}$$

where $\mathcal{U}$ is a real value randomly generated following a uniform distribution in the range of $[0, 1]$;

The value of $\mathbf{L}_i^{(t)}$ is evaluated through Equation 3.31 [Sun et al. (2007)]:

$$\mathbf{L}_i^{(t+1)} = 2\alpha \cdot |\mathbf{p}_{mean}^{(t)} - \mathbf{r}_i^{(t)}| \tag{3.31}$$

where $\mathbf{r}_i^{(t)}$ is the coordinate vector of current particle;

$\mathbf{p}_{mean}^{(t)}$ is the coordinate vector, which consists of the mean of the personal best positions of all particles and denotes the local attractor of the particle i and can be calculated according to Equation 3.33 [Sun et al. (2007)];

α is a contraction-expansion coefficient, which can be tuned to control the convergence speed of the algorithm. In order to guarantee convergence of the particle and to lead the QPSO algorithm to a good performance in general, in most of the papers in the literature on QPSO, like [Sun, Xu & Feng (2004)], the recommended value of α is a linearly decreasing value from 1.0 to 0.5 according to Equation 3.32:

$$\alpha = 0.5 + 0.5 \cdot (T - t)/T) \tag{3.32}$$

where T is is the maximum number of generations;

$$\mathbf{p}_{mean}^{(t)} = \Big(\frac{1}{M}\sum_{i=1}^{M} \mathbf{r}_{ibest,1}^{(t)}, \frac{1}{M}\sum_{i=1}^{M} \mathbf{r}_{ibest,2}^{(t)}, \frac{1}{M}\sum_{i=1}^{M} \mathbf{r}_{ibest,N}^{(t)}\Big) \tag{3.33}$$

where N is the size of the search space (the number of decision variables) and M is the population size.

We have to notice that the introduction of the mean best position $\mathbf{p}_{mean}$ in the algorithm was made in the QPSO version presented in [Sun et al. (2007)]. In this thesis, we use and provide the description of QPSO with the mean best position $\mathbf{p}_{mean}$, because it better enhances the global search ability of the algorithm. This fact can be explained as follows: each particle cannot converge to the global best position without considering its colleagues, because the distance between the current position of particle $\mathbf{r}$ and the $\mathbf{p}_{mean}$ determines the position distribution of the particle for next iteration. If the personal best positions of several particles are far from the global best position, while those of the other particles are near the best global solution, the position $\mathbf{p}_{mean}$ can be pulled away from the global best solution by these "slow" particles. When the "slow" particles are chasing after their colleagues, the position $\mathbf{p}_{mean}$ will be approaching the best global solution slowly. The distances between position $\mathbf{p}_{mean}$ and the personal best positions $\mathbf{r}_{ibest}$ of a particle near the best global solution do not decrease quickly, decelerating the convergence of the particles near the best global solution, and making them explore globally around the best global solution until the "slow" particles are close to the global one. Therefore, the QPSO version with the mean best position seems to be a more cooperative social organism.

Algorithm

The QPSO algorithm is summarized in Algorithm 3, which depicts quantum behaviour of particles in the Delta potential well model with the introduced mean best personal position of particles.

In the pseudo-code of Algorithm 3, T and M denote the maximum number of generation and swarm population size respectively and $\mathcal{U}[0,1]$ is a uniform random number in the range of $[0,1]$.

At the beginning of the algorithm, in order to initialize the first population of solution candidates, we randomly assign the values of the coordinates of each particle $\mathbf{r}_i^{(t)} \in \mathbf{R}$ at $t = 0$, according to the boundary values of the given optimization problem.

Then, the steps presented in the pseudo-code of Algorithm 3 are executed.

Advantages comparatively with its classical counterpart (PSO):

1. it is theoretically guaranteed that QPSO converges to the global optimum, because it can sample a larger region of the solution space and demonstrate a better global search ability especially on non-linear separable optimization problems, due to the following reasons [Sun, Feng & Xu (2004)]:

   - a quantum system has far more states than a linear system: i.e., a quantum system is a complex non-linear system in which the principle of state superposition works on and thus, the number of possible states is larger;
   - before measurement, a particle of a given quantum system can appear anywhere in the search space $\in \mathbb{R}^{\mathrm{d}}$ with a certain probability distribution, because it has no determined trajectory (the uncertainty principle of Heisenberg). Thus, it can appear even at a position far from the best personal position and can be superior to the current best global position.

Algorithm 3: The standard QPSO algorithm.

Result: returns the position vector of the global best particle

begin
  Initialize the current positions randomly
  for $t = 1$ *to* T **do**
    Calculate α using Eq. 3.32
    Calculate $\mathbf{p}_{mean}$ using Eq. 3.33
    for $i = 1$ *to* M **do**
      Calculate fitness $f\left(\mathbf{r}_i^{(t)}\right)$
      Update personal best $\left(\mathbf{r}_{ibest}^{(t)}\right)$ using Eq. 3.25
      Update global best $\left(\mathbf{r}_{gbest}^{(t)}\right)$ using Eq. 3.26
      $c_p, c_g \sim \mathcal{U}[0, 1]$
      Compute the local attractor $\mathbf{p}_i^{(t)}$ using Eq. 3.28
      for $j = 1$ *to* N **do**
        $u \sim \mathcal{U}[0, 1]$
        $L_{i,j} = (2 \cdot \alpha) \cdot |p_{meani,j}^{(t)} - r_{i,j}^{(t)}|$
        $\mathbf{r}_{i,j}^{(t+1)} = \mathbf{p}_{i,j}^{(t)} - L_{i,j} \times ln\left(1/u\right)$ with probability 0.5
        otherwise $\mathbf{r}_{i,j}^{(t+1)} = \mathbf{p}_{i,j}^{(t)} + L_{i,j} \times ln\left(1/u\right)$
  return $\mathbf{r}_{best_{(g)}}$

2. it is also experimentally shown that standard QPSO outperforms conventional PSO variations in finding the optimal solutions and has stronger exploring capabilities [Sun, Xu & Feng (2004)];

3. QPSO has been successfully applied to a vast variety of engineering problems such as system identification [Fei et al. (2008)], power systems [Zhisheng (2010)], image processing [Lei & Fu (2008)], neural network training [Li et al. (2007)] and brain-computer interfacing [Hassani & Lee (2014)].

Limitations:

1. since QPSO is a probability-based searching technique, a larger population size can be required to obtain a better global search ability [Sun, Feng & Xu (2004)];

2. as already discussed earlier, the selected model of potential field has an impact on the global search ability, the convergence rate and the execution time of the algorithm, because the model of potential field defines the equation of the wave function and consequently, the probability distribution, with which a particle can appear at any position in the whole feasible search space [Sun, Feng & Xu (2004)]. Apparently, QPSO would be more efficient if the wave function was as close as possible to the objective function;

3. QPSO is sensitive to α parameter, which has an impact on its performance. In fact, it is connected with the previous limitation: i.e., since the shape of the selected potential well has an impact of the performance, its width also has an impact. According to Equation 3.31, it is seen that the value of $\mathbf{L}$, which determines a search space of each particle in each generation, depends on the value of α. It means that a smart technique is required in order to adjust the value of α so that it ensures a rapid global convergence;

4. the performance of QPSO depends on coordinate system rotations, which explains the lost of efficiency on the non-separable problems, because of the existing correlations between decision variables of such kind of problems. This fact is proved by the experiments presented in Section 5.1.

3.4.5 Non-dominated Sorting Genetic Algorithm III (NSGA-III)

Non-dominated Sorting Genetic Algorithm III (NSGA-III) presented in [Deb & Jain (2013)], is probably the most widely used many-objective evolutionary optimization algorithm, which is based on a reference point-based non-dominated sorting selection mechanism, and has a small number of hyper-parameters.

In this thesis, we use NSGA-III:

1. as a well-known state-of-the-art algorithm for several comparative studies provided in this thesis;

2. as a part of the fusion-based algorithm presented in Section 5.2.1, which mainly aims at ensuring the scalability w.r.t. the number of objectives and w.r.t. the number of decision variables especially, and consequently, to solve different problems of the model of Magneto Caloric Materials (MCMs) and Active Magnetic Regenerator (AMR).

Below, we provide the summary of the original NSGA-III mainly based on the following sources: [Deb & Jain (2013)], [Deb & Deb (2014)], [Seada & Deb (2014)] and [Seada & Deb (2015)]. Since NSGA-III applies the idea of well-spread reference points for maintaining the diversity among population members [Deb & Jain (2013)], we provide a brief description of the determination procedure of reference points, before providing the pseudo-code of the NSGA-III.

3.4.5.1 Determination of Reference Points

As presented in [Deb & Jain (2013)], NSGA-III uses a predefined set of reference points to ensure the diversity of the solution. Each solution (individual) in the population is associated to a reference point.

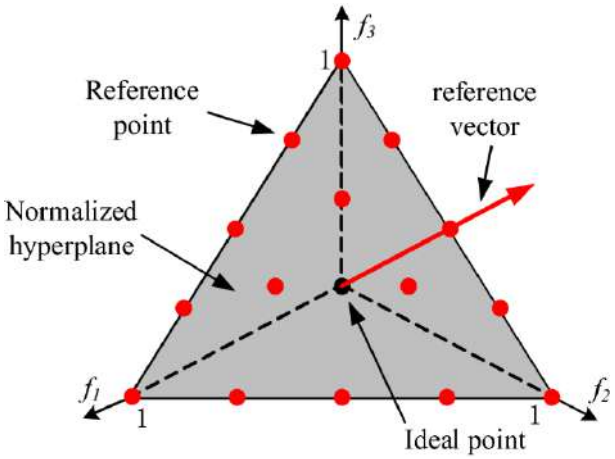


Figure 3.15: Reference points in NSGA-III. The illustration is adapted from [Li et al. (2020)].

In order to determine this set of reference points, NSGA-III uses an approach presented in [Das & Dennis (1998)], which defines the reference points on a normalized hyperplane that is equally inclined to all objective axes and has an intercept of one on each objective axis. The total number of reference points H in problems with the number of objectives m, is calculated by Equation 3.34, where p is a given integer value, which refers to the number of divisions considered along each objective axis.

$$H = \binom{m+p-1}{p} \tag{3.34}$$

For example, in problems with three objectives ($m = 3$), the reference points constitute a triangle according to Equation 3.34 with the vertex at (1, 0, 0), (0, 1, 0), and (0, 0, 1) as it is shown in Figure 3.15, where four divisions ($p = 4$) are considered for each objective axis and $H = 15$ reference points will be created.

Thus, the reference points created by this approach are widely and uniformly distributed on the entire normalized hyperplane. But when p is too small, the intermediate points will not be created; when p is bigger than m, it leads to an excessive number of points in the high-dimensional objective space. Therefore, when the number of objective become larger, a two layer reference-point generation method with small values of p is used, as it is suggested and explained in [Deb & Jain (2013)]. We do not provide more details about this method, because it is not studied in the frame of this thesis.

3.4.5.2 Algorithm

NSGA-III starts with generating a set of H reference points. Then, it randomly initialises a population of size N. If the termination criteria is not satisfied, the next steps presented in the pseudo-code of Algorithm 4 (originally provided in [Deb & Jain (2013)]), will be run repeatedly for each generation t.

Algorithm 4: NSGAIII: Pseudo-code for generation t.

Input: P^t - initial parent population;
Z_s - H structured reference points;
Result: P^{t+1}

$S^t = \emptyset$;
$i = 1$;
$Q^t = \texttt{MakeChildPopulation}(P^t)$;
$R^t = P^t \cup Q^t$;
$\{F_1, F_2, ...\} = \texttt{Non-dominated Sorting}(R^t)$;
repeat
  $S^t = S^t \cup F_i$; $i = i + 1$
until $|S^t| \geq N$;
$F_l = F_i$; *//Last front to be included*
if $|S^t| = N$ **then**
  $P^{t+1} = S^t; break$;
end
else
  $P^{t+1} = \cup_{j=1}^{l-1} F_j$;
  $K = N - |P^{t+1}|$; *// Number of points to be chosen from* F_l
  /**Normalize objectives and create reference set* Z_r:*/
  $\texttt{Normalize}(\boldsymbol{f}^n, S^t, Z_r)$;
  /**Associate each member* $\boldsymbol{s}$ *of* S^t *with a reference point:**/
  /*$\pi(\boldsymbol{s})$ *− closest reference point* * /
  /*$d(\boldsymbol{s})$ *- distance between* $\boldsymbol{s}$ *and* $\pi(\boldsymbol{s})$ * /
  $[\pi(\boldsymbol{s}), d(\boldsymbol{s})] = \texttt{Associate}(\mathrm{S}^t, Z_r)$;
  /* *Compute niche count of reference point* $j \in Z_r$ */
  $p_j = \Sigma_{\boldsymbol{s} \in S^t / F_l}((\pi(s) = j)?1 : 0)$;
  /* *Choose K members one at a time from* F_l *to construct* R_{t+1}*/
  $\texttt{Niching}(K, p_j, \pi(s), d(s), Z_r, F_l, P^{t+1})$;
end

For the sake of completeness, we briefly describe each step of Algorithm 4 below:

1. `MakeChildPopulation` (line 3): the procedure, which takes N members of parent population (P^t) to produce a new child population Q^t with size N by using the following operators:

   (a) a random selection: it seems to be reasonable, since only one population member is considered to be found for each reference point. Thus, there is no need for any tournament selection operator;

   (b) a Simulated Binary crossover (SBX) operator;

   (c) a polynomial mutation operator.

2. a combined population of size $2N$ is formed (line 4), which consists of the parent and children candidate solutions ($R^t = P^t \cup Q^t$);

3. `Non-dominated Sorting` (line 5): the Pareto-based non-dominated sorting procedure that classifies R^t into different non-domination ranks ($F_1, F_2, ...$) in order to select the best N members from R^t for the next generation;

4. a new empty population S^t is constructed by taking the candidate solutions from different non-dominated ranks, starting from F_1 until the size of S^t equals or exceeds N for the first time (line 6-8).

   The last level which is included in S^t is denoted as F_l. Note that all solutions from level $l + 1$ are rejected.

5. the solutions in $S^t \setminus \mathrm{F}_l$ are selected for P^{t+1} (line 14) and the number of candidates solutions of F_l is defined (line 15), which further will be selected for P^{t+1} by an environmental selection in order to maintain diversity in P^{t+1}.

6. `Normalize` (line 17): to prepare for environmental selection, a normalization operator is useded, where the objective values and supplied reference points are normalized so that they have an identical range. This normalization is required, because since we connect each candidate solution with its reference point to maintain diversity, the reference points are evenly distributed in the objective (target) space, and the scale of each objective function value of each solution is different, which leads to different bias of the solutions: e.g., if the range of the objective function f_1 is $[0, 1]$, and the range of f_2 is $[0, 100]$, when the solutions and the reference points are contacted, the roles played by f_1 and f_2 are not "fair";

7. `Associate` (line 21): the procedure, which associates each candidate solution in S^t with a reference point by calculating the perpendicular distance between a member in S^t and each of the reference lines, joining the ideal point with a reference point;

8. for the j-th reference point, the niche count p_j is defined as the number of candidate solutions in $S^t \setminus \mathrm{F}_l$ associated with the j-th reference point (line 23);

9. `Niching` (line 25): the niche-preserving operation, which is used to choose solutions from F_l instead of using a crowding distance, because the latter does not perform well for many-objective problems [Kukkonen & Deb (2006)].

   First, the set of reference points Z_{min} with the minimum p_j value is defined. If $|Z_{min}| > 1$, then $p_j \in Z_{min}$ is randomly chosen. The two scenarios below are then used:

(a) if some individuals in F_l are associated with the j-th reference point, then we consider two cases:

i. If $p_j = 0$, we choice the solution with the shortest perpendicular distance to the j-th reference line among candidate solutions associated with the j-th reference point in F_l, and add it to P^{t+1}. Then, the count of p_j is increased by 1;

ii. if $p_j \geq 1$, the candidate solution (individual) chosen randomly from F_l that is associated with the j-th reference point, is added to P^{t+1}. Then, the count of p_j is increased by 1;

(b) if no candidate solution (individual) in F_l is associated with the j-th reference point, the reference point is excluded from further consideration for the current generation.

This niche operation is repeated for a total of $K = N$ - $|S^t \backslash F_l|$ times to fill the remaining population slots of P^{t+1}.

3.4.5.3 Variation Operators

For each iteration, NSGA-III creates a child population through two operators: Simulated Binary crossover (SBX) [Deb et al. (1995)] and polynomial mutation [Deb & Deb (2014)]. Here, we do not provide their descriptions, because they are well-known and commonly used variation operators and they do not play a key role in this work.

3.4.5.4 Population size

According to [Deb & Jain (2013)], the recommended population size N is the smallest multiple of 4 greater than the number of reference directions with the aim that one candidate solution will be found for each reference point, i.e., $N \approx H$. Thus, the population size depends on the number of reference directions.

Below, we summarise some behavioural features of NSGA-III, depending on the values of N and H:

1. the population size N is small and $N \approx H$:

   (a) NSGA-III can solve efficiently separable and uni-modal problems. It was shown in [Deb & Jain (2013)], on the DTLZ2 problem from the Deb–Thiele–Laumanns–Zitzler (DTLZ) test suite [Deb et al. (2002)] that:

   i. for $m = 3$, $N = 12$ and $H = 10$, NSGA-III finds all 10 Pareto-optimal points after 250 generations;

   ii. for $m = 10$, $N = 68$ and $H = 65$, NSGA-III finds all 65 Pareto-optimal points after 1000 generations.

   (b) NSGA-III is not efficient on multi-modal problems and gets stuck in local optima;

   (c) NSGA-III is not efficient for solving single-objective problems [Seada & Deb (2014)] because of the following reasons:

i. If all the individuals are associated to only one reference direction, the population size should be $N = 4$, according to the recommended population size, which is the smallest multiple of 4 greater than the number of reference directions. This population size is too small for NSGA-III's recombination operator to find useful children candidate solutions (individuals) [Seada & Deb (2014)].

ii. The candidate solutions are picked randomly for the recombination and mutation operators, without any selection pressure [Seada & Deb (2014)].

iii. The `Niching` operation becomes defunct and only use execution time, as there is no concept of perpendicular distance of a function value from the reference direction [Seada & Deb (2014)].

2. the population size is larger than the number of reference points ($N > H$):

   (a) NSGA-III is expected to have a higher ability to escape local optima. As shown in [Seada & Deb (2015)], if $N > H$, NSGA-III becomes a less greedy algorithm, because of its random selection. Indeed, in this cases, some candidate solutions will be guided by the reference directions while the others will keep floating randomly in the search space due to the lack of any selection pressure for them to be focused anywhere in the search space. It provides excessive randomness to the additional candidate solutions and consequently, can help to maintain diversity in the population, which is useful for solving multi-modal function.

   (b) NSGA-III is less dependent on mutation operators. According to experiments presented in [Seada & Deb (2015)] on three highly multi-modal problems, if the population size is larger than the number of reference points, NSGA-III can avoid local optima by completely remove mutation (i.e., mutation probability equals zero).

   (c) The convergence of NSGA-III is expected to be slow on some problems, due to absence of selection in the reproduction procedure.

3.4.5.5 Summary of NSGA-III

Baseline Techniques of NSGA-III:

1. a reference points-based guidance mechanism to preserve diversity among solutions;
2. non-dominated sorting;
3. no selection pressure, i.e., there are two random selection: (i) NSGA-III randomly selects parents in reproduction procedure; (ii) NSGA-III randomly selects among the individuals in a cluster when the niche count is greater than zero in the niching procedure.

Computational Complexity for the Worst Case for a Single Generation:
According to the analysis presented in the original paper [Deb & Jain (2013)], the overall worst case complexity for one generation of NSGA-III is $\mathcal{O}(N^2 log^{m-2} N)$ or $\mathcal{O}(N^2 m)$, where N is the population size and m is the number of objectives.

Advantages:

1. NSGA-III is very efficient for solving many-objective problems. This efficiency is shown by analysis and experiments in [Seada & Deb (2014)], [Deb & Deb (2014)];

2. NSGA-III has a few number of tuning parameters [Deb & Deb (2014)]: i.e., population size, termination parameters, parameters associated with SBX crossover and polynomial mutation (e.g., their probabilities);

   In the original paper [Deb & Jain (2013)], the number of reference points H is not considered as another one algorithmic parameter because it is related to population size N ($N \approx H$);

Limitation:

1. NSGA-III is not efficient on single-objective problems [Seada & Deb (2014)];

2. It is difficult for NSGA-III to solve bi-objective optimization problems, due to the mild selection pressure that NSGA-III introduces to non-dominated solutions of a population [Seada & Deb (2014)];

3. NSGA-III does not ensure good scalability w.r.t. the population size (see Section 4.1.2.1).

3.5 Benchmarking

In this section, we briefly overview the test suites, which are used in this thesis for providing different experiments.

3.5.1 Black-Box Optimization Benchmarking (BBOB)

To reduce the risk of biased benchmarking, in this thesis, we extensively use the Black-Box Optimization Benchmarking (BBOB) framework [Hansen et al. (2012)] of the COmparing Continuous Optimizers (COCO)[8] platform [Hansen et al. (2021)] for single and bi-objective experiments. This framework is widely used in the community for comparing algorithms performance in continuous search spaces and for estimating a statistical significance of the results. The COCO platform provides both tools needed to run the simulations (experiment code) in different programming languages (C, C++, Java, Python, Matlab and Octave) and a Python code that allows users to post-process the data. The post-processing code is used to produce different useful plots and tables for visualization of the obtained results.

The BBOB framework provides 24 noiseless single-objective [Hansen et al. (2009)] test functions, their extended versions for large-scale optimization [Elhara et al. (2019)], and 55 bi-objective [Brockhoff et al. (2016)] problems, which present different combinations of the 24 single-objective functions. Each problem in all test suites has 15 differently parameterized instances. For a reliable statistical analysis of the results, 15 runs (trials) of the benchmarked algorithm should be conducted on different instances of each test problem f.

[8] `https://coco.gforge.inria.fr/`

The Single-objective Noiseless BBOB test suite:

The functions of the single-objective objective noiseless (*bbob*) test suite and their extension for large-scale optimization (*bbob-largescale*) [Elhara et al. (2019)] are grouped into the following five categories, where each category contains a set of functions that share some property/difficulty:

- **Separable functions** ($f1 - f5$): these functions are considered to be easy since they can be solved independently on each dimension. The Rastrigin functions ($f3$ and $f4$) present the additional difficulty of being highly multi-modal, which makes them harder for many algorithms.
- **Low/moderately conditioned functions** ($f6 - f9$): these functions have a condition number of about 100. The two first functions ($f6$ and $f7$) are uni-modal (single optimum), while the Rosenbrock functions ($f8$ and $f9$) have a second local optimum.
- **Uni-modal highly conditioned functions** ($f10 - f14$): these functions have a single optimum. They have a relatively high condition number in comparison to the functions of the previous category: 10^6 for $f10$, $f11$ and $f12$.
- **Multi-modal functions with adequate global structure** ($f15 - f19$): these functions are highly multi-modal, with a number of local optima that depends on the dimension. However, the adequate structure that might be exploited by algorithms means that the local optima are generally similarly shaped and distributed in a regular way. Thus, seen on a global scale, the landscape of the function contains repetitive, symmetric, patterns.
- **Multi-modal functions with weak global structure** ($f20 - f24$): similar to the previous category in multi-modality but the landscapes of the function has less structure and symmetries are broken. Most algorithms find these problems to be the hardest.

As an input, only the dimension of problem (d) and the number of function evaluations must be chosen. All functions are defined for dimensions $d \in [2, 3, 5, 10, 20]$ and extended up to d =640 in the large scale test suite (*bbob-largescale*). Each function can be evaluated over $\mathbb{R}^d$, while the actual search domain is given as $[-5, 5]^d$.

The goal of experiments on the *bbob* and *bbob-largescale* test suites is to reach the target function value $f_{target} = f_{opt} + \Delta f$, ($\Delta f = 10^{-7}$), using as few function evaluations as possible. The optimum function value f_{opt} is different for each function and for each instance of the function.

The Bi-objective BBOB test suite:

Bi-objective functions of the bi-objective BBOB test suite (*biobj-bbob*) are derived from combining functions of the *bbob* suite. On the *bbob-biobj* test suite, the performance of algorithms is assessed in terms of the target Quality Indicator (QI) value I_{HV} (the Hypervolume (HV), described in Section 3.1.2.2). More precisely, the goal is to minimize the difference between the reference HV of a reference Pareto Front (consisting of the best known set of objective vectors for each problem) and the achieved HV: $I_{HV} = I_{ref} + \Delta I$, ($\Delta I = 10^{-5}$), using as few function evaluations as possible. The I_{ref} is different for each problem and each instance of the problem.

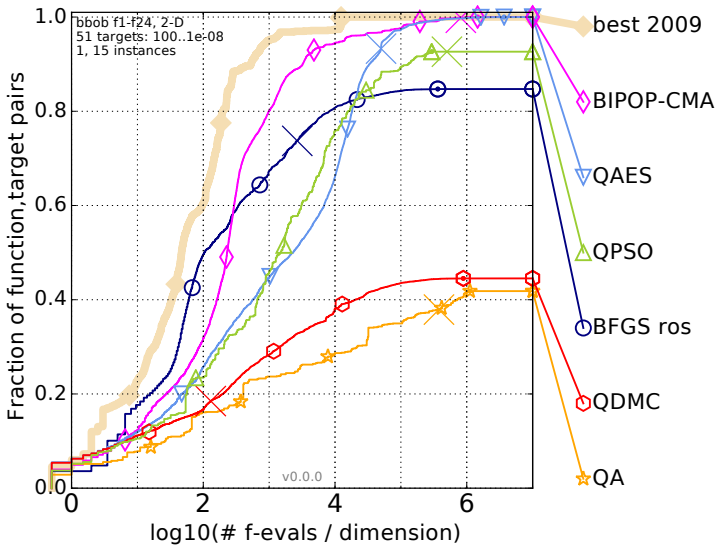


Figure 3.16: Example: Empirical Cumulative Distribution of runtime, summarized for all Functions.

Results Presentation

In the COCO platform, performance is measured in terms of Expected Run Time (ERT), which depends on a given target function value. The ERT is computed over all relevant runs on the given number of instances as the number of function evaluations executed during each run, until the best function value (f_{target}) is reached, summed over all runs and divided by the number of runs that actually reached the target value [Hansen et al. (2012)]. Mathematically, the ERT is formulated as follows:

$$ERT(f_{target}) = \frac{\#FEs(f_{best} \geq f_{target})}{\#succ} \tag{3.35}$$

where the $\#FEs(f_{best} \geq f_{target})$ is the number of function evaluations conducted in all trials, while the best function value is not smaller than f_{target} during the run; $\#succ$ is the number of successful runs.

In order to summarize and to compare the results of benchmarked algorithms, Empirical Cumulative Distribution Functions (ECDFs) of the running time are used. The ECDF is a monotonous function $F : \mathbb{R} \mapsto [0, 1]$ defined for a given set of real-valued data S, such that $F(x)$ equals the fraction of elements in S, which are smaller than x. The function F is a lossless representation of the set S [Hansen et al. (2012)].

For the sake of visualization simplicity, in the COCO platform, the ECDFs are aggregated by the presented above categories of problems and by dimensions. An example of the results presentation in form of ECDFs is illustrated in Figure 3.16, where any given value of the x-axis indicates a number of function evaluations in logarithmic scale, divided by the dimension and used to solve a given value of y-axis, which presents a proportion of the solving problems. Consequently, if the graph reaches the value 1 of the y-axis, it means that all runs reached the best target value: i.e., 100% of problems are solved with the highest precision.

The horizontal distance between graphs represents a difference in runtime for solving the same proportion of problems. The area between two graphs, up to a given y-value,

is the average runtime difference. The best algorithm covers the largest area under its graph.

The "best 2009" results shown as a thick transparent line, corresponds to the best solutions of multiple algorithms during the 2009 GECCO workshop. Consequently, they are not taken into account in the comparative experiments of this thesis, because it does not correspond to the performance of a single algorithms.

3.5.2 Scalable Multi-objective Benchmarking

Although COmparing Continuous Optimizers (COCO) is one of most reliable benchmarking platforms, it unfortunately provides only single- and bi-objective optimization test problems. Consequently, the performance of Multi-Objective Evolutionary Algorithms (MOEAs) on problems with dimension of objective (target) space $m \geq 3$ can not be evaluated using the current the COCO platform.

Thus, for an unbiased assessment of algorithms scalability w.r.t. the number of objectives, in this thesis, two representative scalable test suites are used: Deb–Thiele–Laumanns–Zitzler (DTLZ) [Deb et al. (2002)] and Walking Fish Group (WFG) [Huband et al. (2005)].

Deb–Thiele–Laumanns–Zitzler Test Suite

The Deb–Thiele–Laumanns–Zitzler (DTLZ) test suite [Deb et al. (2002)] includes nine test problems for comparing multi-objective algorithms, which are scalable to any number of decision variables and objectives. The majority of these problems are separable, including degenerated and multi-modal Pareto optimal fronts, of which the exact shape and location are known.

In this thesis, we selected the following tests configuration for our algorithms:

- **Selected problems:** the five unconstrained problems, which are the most frequently used tests in the literature, i.e., DTLZ1-DTLZ4 and DTLZ7.
- **Dimension of the objective space:** the number of objectives, i.e., m, are specified in the following range: $m \in [3-5]$.
- **Dimension of search space:** the number of decision variables, i.e., d, is given by the equation proposed in [Deb et al. (2002)]: $d = m + k - 1$, where m represents the number of objectives and k is the number of distance parameters. We applied the values of k suggested in [Deb et al. (2002)]: $k = 5$ for DTLZ1 and $k = 10$ for DTLZ2-DTLZ4 and DTLZ7.
- **Pareto-optimal front:** The DTLZ1 problem has the corresponding Pareto-optimal front in $f_i \in [0, 0.5]$, DTLZ2-4 problems in $f_i \in [0, 1]$ and DTLZ7 problems in $f_{i<m} \in [0, 1]$, $f_m \in [0, 7]$, where $i \in [1, m]$.

The characteristics of the selected problems of the DTLZ test suite are shown in Table 3.2.

Table 3.2: DTLZ functions

Function	Characteristics
DTLZ1	linear, separable, multi-modal
DTLZ2	concave, separable, uni-modal
DTLZ3	concave, separable, multi-modal
DTLZ4	concave, separable, uni-modal biased
DTLZ7	discontinuous

Walking Fish Group Test Suite

The Walking-Fish-Group (WFG) test suite [Huband et al. (2005)], suggests nine multi-objective test problems: WFG1-WFG9, that are scalable w.r.t. the number of objectives and decision variables, and have known Pareto optimal sets.

These problems include a wide variety of Pareto optimal geometries. Moreover, characteristics such as bias, multi-modality, and non-separability are defined by a set of transformations.

In this thesis, we selected the following tests configuration:

- **Selected problems:** WFG1-WFG9.
- **Dimension of target space:** the number of objectives, i.e., m, are specified in the following range: $m \in [3-6]$.
- **Dimension of search space:** the number of decision variables, i.e., d, is given by the equation suggested in [Huband et al. (2005)]: $d = k + l$, where the position parameter k is set to $k = 2(m-1)$, and the distance parameter l is set to $l = 20$.

Table 3.3 summarizes the properties of the WFG problems.

Table 3.3: WFG Functions

Function	Characteristics
WFG1	mixed, separable, uni-modal, biased
WFG2	discontinuous, non-separable, multi-modal
WFG3	partially degenerate, non-separable, uni-modal
WFG4	concave, separable, multi-modal
WFG5	concave, separable, uni-modal, deceptive
WFG6	concave, non-separable, uni-modal
WFG7	concave, separable, uni-modal,biased
WFG8	concave, non-separable,uni-modal,biased
WFG9	concave, non-separable, multi-modal, biased, deceptive

3.6 The EASEA platform

The EASEA[9] (*EAsy Specification of Evolutionary Algorithms*) [Collet et al. (2000)], [QUERRY et al. (2017)] platform is aimed at allowing scientists to apply evolutionary computation to solve their real-world problems through a user-friendly interface. Consequently, it suits perfectly as a software support for the optimization tool, developed in this thesis. By itself, EASEA is an open source cross-platform (Linux, MacOS and Windows) and cross-compiler (Clang *C++*, GNU *C++* and Visual *C++*) framework for evolutionary optimization, hosted on GitHub[10]. There are two main parts, distributed with the EASEA software: an internal *C++*-based library (*LibEASEA*), that is statically compiled with the produced executable file and a *lex/yacc*-based code generator/compiler (*easena*).

This compiler, called *easena*, couples one of the EASEA template files (**.tpl* - a skeleton of optimization algorithm, selected by the user) and a problem-specific part (**.ez* - the optimization problem, defined by the user).

The compiler goes through the template file of the selected algorithm and inserts the problem specifications into an instance of the template. At the end of the automatic compilation process, the ready-to-use source code of algorithm with the integrated optimization problem is provided as a set of the following files: **.cpp/*.cu*, **.h*, **.prm* to be compiled by $C++/nvcc$ compiler via automatically generated *Makefile*. This process is depicted in Figure 3.17.

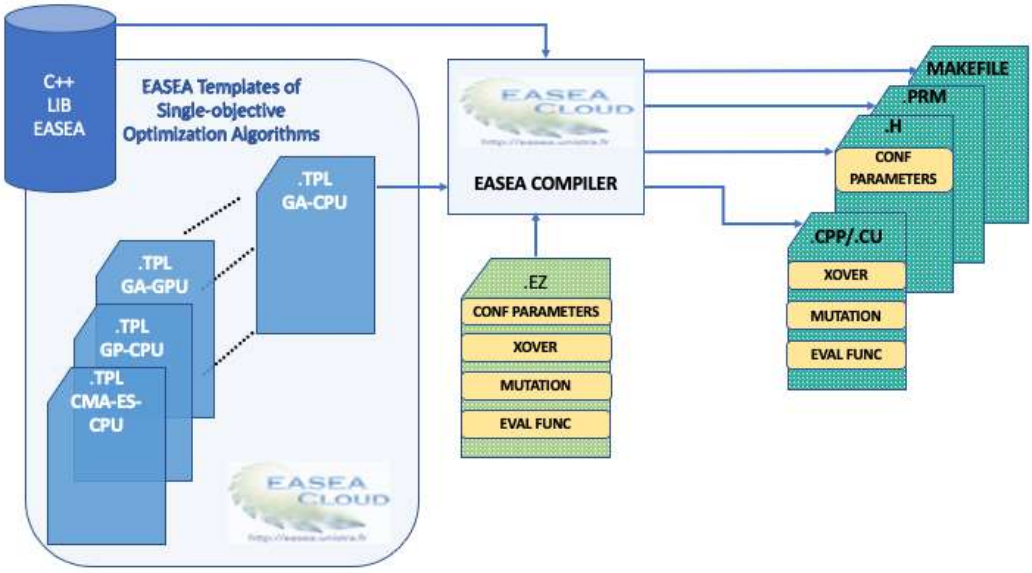


Figure 3.17: The EASEA compilation process.

The main features of EASEA are the following:

- An automatic coupling of an evolutionary algorithm template with an optimization problem-specific part into a *C++* source code, without human intervention. This reduces the possibility of making mistakes, and makes conducting experiments convenient.

[9] `http://easea.unistra.fr`
[10] `https://github.com/EASEA/easea`

- A parallel computing framework. The software not only creates code for multi-core CPUs but also supports GPUs [Collet et al. (2013)].
- An object-oriented architecture, which allows the programmer to use different modules and arbitrarily combine them with each other by polymorphism.
- A set of problem-independent optimization algorithms, which can be applied to various test suites and real-world applications.
- A set of experimental examples of single-objective optimization problems.
- The preservation of experimental data in simple **.csv* files.
- A simple visualisation tool.

Despite the above-mentioned advantages, four important features were not implemented in the EASEA platform:

1. The support of Multi-objective Optimization Problems (MOPs). Only standard single-objective evolutionary algorithms are implemented: Genetic Algorithm, Genetic Programming, CMA-ES, Differential Evolution.
2. A programming language independent interface. The problem-specific part (optimization problem) must be implemented in *C++*.
3. Variation operators (e.g., the crossover and mutation) are not implemented in *LibEASEA* and have to be defined by users directly in the problem-specific (**.ez* file), which requires some special knowledge and skills from users.
4. Customised and extended logging, which is necessary for reproducibility of research.

These missing features had to be implemented (especially 1, 2 and 3) in order to satisfy the requirements, formulated in Section 1.1. For this purpose, in this thesis, we implemented the updated version of EASEA presented in Section 6.1.1), where the missing features are integrated.

3.7 Connection with the Contribution

In this chapter, we reviewed the fundamentals of continuous optimization (i.e., the algorithms, methods, platforms and test suites) in order to clearly explain the convenience of the techniques, which are selected and used in the frame of this thesis.

Since we deal with computationally intensive simulation models, which are used as a core for various optimization problems, the challenge is to provide a special software tool, which provides a user-friendly interface along with the algorithms capable to solve different time-consuming single-, multi- and many-objective problems.

According to the analysis presented in Section 3.2.1, we can conclude that the surrogate-based methods can not be used for solving the various problems of the frequently modified models, because they are not flexible enough and can not provide the required scalability w.r.t. different dimensions of search and objective spaces. Instead, we propose a user-friendly software tool based on the upgraded version of the EASEA platform, which, besides several state-of-the-art algorithms, includes three optimization algorithms developed

specially to solve different time-consuming problems of the Magneto Caloric Materials and Active Magnetic Regenerator models. These algorithms presented aim at different objectives and are based on the techniques, which are presented in this chapter. The developed algorithms, together with the modified version of the platform EASEA, are presented in the next part of this thesis.

Part III

Contributions

Chapter 4

Archive-based Multi-objective Optimization Algorithm: FastEMO

According to the defined research strategy, we present in this chapter a parallel evolutionary optimization algorithm for solving time-consuming continuous multi- and many-objective problems of the model of Active Magnetic Regenerator (AMR).

The proposed algorithm, called FastEMO, is designed to take into consideration the following possible particularities of optimization problems of the AMR model presented in decreasing order of importance:

1. **Computationally intensiveness**: i.e., the objective functions are time-consuming to be evaluated because of the computationally expensiveness of the AMR model, which is a part of the objective functions. Moreover, the execution time of one simulation run of the single-mode operating AMR model has a high variance: i.e., 2h-15h on an AMD EPYC 7371 16-Core Processor.

2. **Scalablility w.r.t. the dimensions of the objective (target) space**: i.e., the ability of an algorithm to solve optimization problems for increasing the number of objectives (from 2 to 5 objectives).

3. **Control of solutions**: i.e., efficiency with non-dominated solutions, which are different on the search space, but very similar on the target (objective) space;

4. **Scalablility w.r.t. the dimensions of the search space**: i.e., the ability of an algorithm to scale w.r.t. the number of decision variables (from 3 to 150). The number of decision variables depends on the number control and design parameters, which are taken under consideration.

5. **Presence of Dominance Resistant Solutions (DRS)**: i.e., the solutions, which have extremely good values for some objectives and extremely bad values for other objectives.

As explained in Section 3.2.1, in order to enhance computational speed for solving time-consuming optimization problems, we turn to the advantage of parallel computing and focus on an approach based on the parallel iteration-level model for Multi-Objective Evolutionary Algorithms (MOEAs), where all solutions in the population are evaluated in parallel at each generation. Taking into account that the parallel iteration-level model

requires a large population size [Talbi (2019)], [Falcón-Cardona et al. (2021)], we concentrate our attention on the design of MOEA, which can ensure two following features for its parallel implementation:

1. **Low computation complexity**.
2. **Scalability w.r.t. the population size:** i.e., the ability of the algorithm to approximate the Pareto front with a good accuracy with increasing the population size while proportionally decreasing the number of generations.

This chapter is organized as follows. First, in Section 4.1, we discuss the importance of the two mentioned features for the parallel iteration-level model and explain our choice of the Archived-based Stochastic Ranking Evolutionary Algorithm (ASREA) [Sharma & Collet (2010*a*)] as an inspiriting technique. Second, in Section 3.4.1, we analytically and experimentally investigate the applicability of ASREA w.r.t. the mentioned particularities of the optimization problems of the AMR model and show the identified disadvantages of ASREA. Then, in Section 4.2, we provide an improved version of ASREA, FastEMO, which is capable to solve multi- and many-optimization problems with the mentioned particularities. The design components are explained and discussed. Next, FastEMO is validated in Section 4.3 by using benchmark problems of the following test suites: Deb–Thiele–Laumanns–Zitzler (DTLZ) [Deb et al. (2002)], Walking Fish Group (WFG) [Huband et al. (2005)] and Black-Box Optimization Benchmarking (BBOB) [Hansen et al. (2012)].

The preliminary content of this chapter was published in [Ouskova Leonteva et al. (2019)].

4.1 Multi-Objective Evolutionary Algorithms for Parallel Implementation

In this thesis, we use a classical master-slave model as a parallel iteration-level model for Multi-Objective Evolutionary Algorithms (MOEAs): i.e., the master handles the initialization of the population, the selection and the replacement operators, while the variation operators, i.e.., mutation, crossover, and the objective functions evaluations are managed in parallel by the slaves.

As it was discussed in Section 3.2.1, in this thesis, the master-slave model is implemented. Consequently, the computational complexity of the operators of the master has an impact on the total speedup [Talbi (2019)], [Falcón-Cardona et al. (2021)].

The computational complexity and scalability w.r.t. the population size define the gain obtained by the parallelization of MOEAs [Talbi (2019)]. In order to select the most suitable Multi-Objective Evolutionary Algorithm (MOEA) for the parallel iteration-level model for solving different problems of the Active Magnetic Regenerator (AMR), we start this section by discussing the computational complexity of different state-of-the-art MOEAs and an impact of scalability w.r.t. the population size on the speedup of MOEAs. Next, we explain the selection of the Archived-based Stochastic Ranking Evolutionary Algorithm (ASREA) as a baseline MOEA for parallel implementation w.r.t. the computational complexity and the population size scalability.

4.1.1 Computational Complexity

Computational complexity is usually defined as a function of the population size [Curry & Dagli (2014)]. Big-O notation ($\mathcal{O}$) is used to measure and compare the worst-case scenarios of different algorithms theoretically, denoting the asymptotic upper bound in a single generation. The reason why the computational complexity is stated w.r.t. a single generation is that it allows algorithms to be compared when the convergence rate of the overall algorithm is unknown [Curry & Dagli (2014)].

The computational complexity for a single generation depends on the technique on which the algorithm is based. Below, we provide the list of the baseline techniques with their computational complexity, which are commonly used in parallel implementation according to the most recent survey about parallel multi-objective evolutionary algorithms presented in [Falcón-Cardona et al. (2021)]:

1. **Domination-based MOEAs:** e.g., Non-dominated Sorting Genetic Algorithm II (NSGA-II) [Deb et al. (2000)] and Non-dominated Sorting Genetic Algorithm III (NSGA-III) [Deb & Jain (2013)], Controlling Dominance Area of Solutions (CDAS) [Sato et al. (2007)] and Multi-Objective Covariance Matrix Adaptation Evolution Strategy (MO-CMA-ES) [Igel et al. (2007)].

   Such kind of MOEAs consist of two operator [Huang et al. (2019)]:

   (a) **A Non-Dominated Sorting (NDS)**. It serves to achieve a good convergence, by assigning a rank based on Pareto-dominance principle to each solution and comparing them by the assigned rank [Deb & Jain (2013)]. This technique consumes the main part of the computational complexity of the algorithm for a single generation. The worst-case computational complexity for NDS is $\mathcal{O}(mn^2)$, where m is the number of objectives and n is the population size.

   (b) **The diversity preserving operators**. They are used as additional sorting criterion for solutions with the same ranks and are based on different techniques:

      i. **A crowding distance** [Deb et al. (2000)]. It is computed as the average distance between the two points on either side of each solution along each objective Its computational complexity is $\mathcal{O}(mnlog(n))$. The crowding distance is used in NSGA-II [Deb et al. (2000)] and CDAS [Deb et al. (2000)].

      ii. A **niching strategy** [Deb & Jain (2013)]. It uses a set of reference points, which are updated according to the extent of the population. Its computational complexity is $\mathcal{O}(n)$. The niching strategy is used in NSGA-III [Deb & Jain (2013)].

      iii. **A contributing hypervolume** [Igel et al. (2007)]. It is based on the calculation of hypervolume value and is used in MO-CME-ES.

   As seen, the computational complexity in terms of Big-O notation of such MOEAs increases with the population size and the number of objectives, which is critical for the parallel iteration-level model, where a large population size is required. Furthermore, it is critical as well for solving the many-objective and/or large-scale optimization problems of the model of Active Magnetic Regenerator (AMR).

2. **Decomposition-based MOEAs:** e.g., Multi-objective Evolutionary Algorithm Based on Decomposition (MOED-D) [Zhang & Li (2007)].

   These approaches adopt a transformation of multi-objective problem by using decomposition strategies.

   The **Decomposition Strategy** is a technique for decomposing a multi-objective problem into t single-objective sub-problems, which are represented by t weight vectors, and are solved simultaneously. The sub-problems are associated with weight vectors in order to obtain a wide range of solutions. Each weight vector is used as a direction to define a scalar function. Solutions of each sub-problem are optimized by performing evolutionary operations among its several neighboring sub-problems. Neighbor structures between sub-problems are defined based on the distance between their weight vectors [Zhang & Li (2007)], [Huang et al. (2019)].

   Commonly used decomposition techniques are **Weighted Sum Approach** [Miettinen (2000)] and **Tchebycheff Approach** [Miettinen (2000)]. The main advantage of the Tchebycheff approach is that it works regardless of the shape of the Pareto front, while other decomposition approaches (such as the weighted sum approach) only work for convex Pareto fronts.

   The computational complexity of decomposition-based MOEAs is $\mathcal{O}(mnt)$, where $t < n$ is the number of weight vectors. As we can see, this computational complexity is lower than that of NDS ($\mathcal{O}(mn^2)$) [Huang et al. (2019)]. Consequently, such kind of algorithm seem promising for a highly parallel implementation.

   However, decomposition-based MOEAs have the following disadvantages [Huang et al. (2019)], which are critical for a massively parallel MOEA aimed at solving different many-objective problems of the AMR model: (i) the population size cannot be set arbitrarily and increases nonlinearly with the increase of the number of objectives; (ii) the weight vector distribution is not very uniform for three or more objectives: (iii) it can be difficult to choose a suitable decomposition method for different problems.

3. **Performance indicator-based MOEAs:** e.g., Indicator Based Evolutionary Algorithm (IBEA) [Zitzler & Künzli (2004)].

   The **Performance Indicator** of a solution quality measurement, compliant with Pareto dominance, e.g., a hypervolume, is integrated into a MOEA as the criteria for environmental selection, in order to guide the search and continually optimize the expected attributes of the entire population. The indicator-based MOEAs, such as IBEA [Zitzler & Künzli (2004)], require a scaling factor tht is dependent on the problem and the used indicator.

   The computational complexity of IBEA is ($\mathcal{O}(n^2)$), which is less than those of NDS ($\mathcal{O}(mn^2)$) [Huang et al. (2019)]. However, the calculation of some indicators, e.g., hypervolume, is a time-consuming procedure by itself, which becomes very computationally intensive with an increasing number of objectives (see Equation 3.3). The fact that the indicator calculation procedure can be more time-consuming than a non-dominated sorting procedure for many-objective problems, makes its application unsuitable for a parallel iteration-level model.

The advantage of the indicator-based technique is that some indicators, like hypervolume, take into account convergence and diversity of solutions to enhance the selection pressure and guide the search to the optimal Pareto front [Huang et al. (2019)].

However, its higher selection pressure than Pareto domination guides solutions towards some specific regions of the Pareto front, causing some optimal solutions to disappear during the evolutionary process. Then final solutions may not be evenly distributed along the Pareto front [Huang et al. (2019)], which is critical for the optimization problems of the AMR model, where all possible design configuration are sought.

4. **Archive-based MOEAs:** e.g., Strength Pareto Evolutionary Algorithm II (SPEA-II) [Zitzler et al. (2001)], Archived-based Stochastic Ranking Evolutionary Algorithm (ASREA) [Sharma & Collet (2010*a*)].

   Commonly speaking, an archive is an external set of non-dominated solutions in order to store good solutions and propagate them to the next iteration. E.g., Strength Pareto Evolutionary Algorithm II (SPEA-II) [Zitzler et al. (2001)] uses an archive to introduce an elitism for better convergence, where the size of the archive has the same value that the main population. It was shown in [Zitzler et al. (2001)] that in higher dimensional target spaces, SPEA-II provides better distribution of solutions compared to NSGA-II, however, this better distributing ability comes with a larger computational complexity in its selection/truncation approach compared to that in the objective-wise crowding distance approach of NSGA-II. Consequently, since the size of the archive of SPEA-II has the same size as the main population, it does not provide advantages for reducing computational complexity [Sharma & Collet (2010*a*)], which is the same as NSGA-II ($O(mn^2)$).

   However, an archive can be used not only to introduce an elitism for better convergence, but also to reduce computational complexity of Non-Dominated Sorting (NDS) procedure. E.g., Archived-based Stochastic Ranking Evolutionary Algorithm (ASREA) [Sharma & Collet (2010*a*)].

   ASREA was developed to cut the usual $\mathcal{O}(mn^2)$ computational complexity into $\mathcal{O}(man)$, where m is the number of objectives, a is the size of archive and n is the population size, by using a stochastic ranking operator. This relatively cheap complexity gives ASREA a potential ability to work with very large population sizes, which makes this algorithm very suitable for a massive parallel implementation.

To summarize, for a highly parallel iteration-level model, a MOEA requires the following features: (i) the computational complexity for a single generation should to be lower than $\mathcal{O}(n^2)$, where n is the size of population; (ii) the scalability w.r.t. the population size; (iii) it should be possible to set the population size arbitrarily.

The catch is that the computational complexity discussed here is not the aggregated complexity of algorithm, but the computational complexity for a single generation, because it allows algorithms to be compared when the convergence rate of the overall algorithm is unknown [Curry & Dagli (2014)]. In order to calculate the actual computational complexity of a MOEA, it is necessary to know both the complexity for a single generation and the number of generations. The number of generations required to achieve a

good approximation of the Pareto front depends not only on the problem to solve, but also on the population size. It is strongly correlated with the question of the scalability of algorithms w.r.t. the population size.

4.1.2 Population Scalability

In order to guarantee a low value of total computational complexity, a parallel implemented MOEA has to ensure the convergence of an algorithm to well approximated Pareto front in the smallest number of generation, where a source of speedup is to increase the population size and to run the objective function computations in parallel.

The population size is one of the significant parameters of an evolutionary algorithm since it has direct influence on its dynamics and search abilities [Roeva et al. (2015)], [He et al. (2011)]. According to [Roeva et al. (2015)], [Rylander & Gotshall (2002)], [Falcón-Cardona et al. (2021)] in general, increasing the population size improves the accuracy of algorithms, but it also increases the number of generations to converge. Consequently, the scalability w.r.t. the population size is a very important property of algorithms for an efficient parallel implementation. The notion of population scalability is similar to that of the speedup widely used when analysing parallel algorithms [He et al. (2011)].

The **population scalability** describes the relationship between the performance of an optimization algorithm and its population size. It can be defined by the following equation [He et al. (2011)]:

$$S_{pop} = \frac{P_{best}}{P} \tag{4.1}$$

where S_{pop} is the population scalability, P_{best} is the mean best value of the performance obtained by the tested algorithm with its most efficient combination of the population size and the number of generations and P is the mean value of the performance obtained by tested algorithm with an increased population size.

The performance of an evolutionary algorithm may be evaluated by different measures: e.g., the Expected Run Time (ERT) presented in Section 3.5.1 or the Hypervolume (HV) metric presented in Section 3.1.2.2. For parallel implementations, the required scalability w.r.t. the population size is that an algorithm can find good a approximated Pareto front with a large population size in the same total number number of function evaluations than with a smaller population size. Consequently, the number of generations has to be decreased proportionally to the increase of the population size. The total number of function evaluations is calculated as follows:

$$EF = G \cdot N \tag{4.2}$$

where EF is the total number of function evaluations, N is the population size and G is the total number of generations used for the search of the solution.

Although intuitively, the performance of an evolutionary algorithm may be improved if its population size increases, only a few case studies for simple fitness functions [He et al. (2011)] were investigated in literature. Furthermore, some important questions about the population scalability are still opened [He et al. (2011)]: (i) How should the threshold of the population size be determined, when an algorithm loses its superlinear scalability? (ii) Is there any feasible approach to estimating the population scalability? (iii) How should the population scalability be improved for a specific algorithm?

4.1.2.1 Population Scalability of Different MOEAs

Taking into account that a feasible approach for measuring the population scalability is still not provided, in this section, we experimentally investigate it on the different Multi-Objective Evolutionary Algorithms (MOEAs), which are commonly used in parallel implementations.

Experimental Objective: The objective of current experiment is to investigate the scalability w.r.t. the population size of six baseline state-of-the-art algorithms.

Algorithms and Parameters:

- The following five MOEAs are selected from the most recent survey about MOEAS for parallel implementation, presented in [Falcón-Cardona et al. (2021)]:
  - Non-dominated Sorting Genetic Algorithm II (NSGA-II) [Deb et al. (2000)];
  - Non-dominated Sorting Genetic Algorithm III (NSGA-III) [Deb & Jain (2013)];
  - Controlling Dominance Area of Solutions (CDAS) [Sato et al. (2007)];
  - Multi-objective Evolutionary Algorithm Based on Decomposition (MOED-D) [Zhang & Li (2007)];
  - Indicator Based Evolutionary Algorithm (IBEA) [Zitzler & Künzli (2004)].
- An archive-based method, called ASREA [Sharma & Collet (2010*a*)], is selected as well, because it was developed specially for a massively parallel implementation.

All algorithms were launched without further parameter tuning, the crossover, mutation operator and parameter settings are defined according to the suggested specification of original papers of algorithms, which were mentioned above.

Test Problems: First four widely used 3-objective instances (i.e., the number of objectives $m = 3$) from Deb–Thiele–Laumanns–Zitzler (DTLZ) [Deb et al. (2002)] test suite are employed: DTLZ1-DTLZ4. For more details, one can see Section 3.5.2.

Performance Metrics:

- $\overline{I}_{HV}$ - mean value of hypervolume indicator over 50 independent runs of each algorithm for each test case;
- **Runtime execution** - mean value of total execution time in seconds over 50 independent runs of each algorithm for each test case;

Simulation Settings:

- **the computational budget:** it has fixed number of evaluation functions for all test cases, i.e., $EF = 500000$;
- **the population size:** each algorithm is examined for various specifications of the population size: i.e., $n = 100$, $n = 1000$, $n = 10000$ under the fixed computation budget ;

- **the generation number:** it depends on the value of the population size, i.e., it decreases proportionally to increasing of the population size: $t = 5000$, $t = 500$ and $t = 50$ respectively;

- **the number of decision variables:** it is calculated according the suggested method in the original paper, where DTLZ test suite was presented [Deb et al. (2002)]. More specifically, the number of decision variable for each test function is calculated according to the following equation proposed in [Deb et al. (2002)]: $d = m + k - 1$, where m represents the number of objectives and k is the number of distance parameters. We applied the values of k suggested in [Deb et al. (2002)]: $k = 5$ for DTLZ1, while $k = 10$ for DTLZ2-DTLZ4. Consequently, $d = 7$ for DTLZ1 and $d = 12$ for the other problems.

- **the statistics:** each algorithm is run over 50 independent runs for each test case.

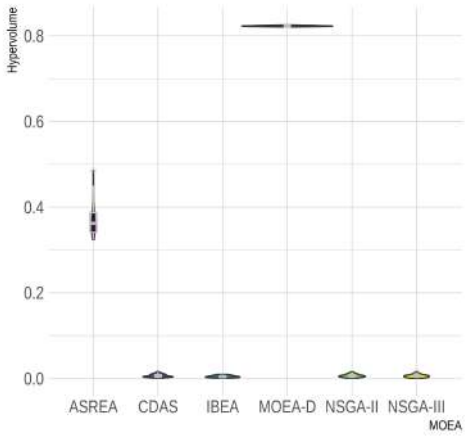
Test Environment:

- All algorithms have a lot of minor unwritten specifications, which can have an impact on their performance. For this experiment we use the source codes of all algorithms, implemented earlier for this purpose in the frame of this thesis in the new version of the platform EASEA 2.20, which serves as a software support (see Section 6.1.1.1).

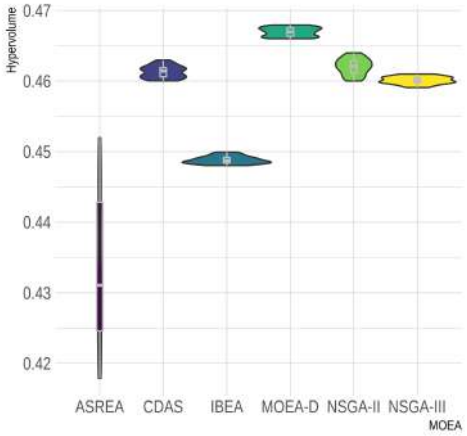
- All the experiments have been conducted on an Intel(R) Pentium(R) CPU 4405U @ 2.10GHz 4 processors laptop via the platform EASEA version 2.20, using the code language (C++) and compiler (g++ 5.4.0).

Experimental Results: We compare the results of MOEAs by average value of the hypervolume indicator in Table 4.1 and by average value of the total runtime in Table 4.2 over 50 independent runs of every algorithm for each test case of each function. More precisely, according to the described above settings, each function has tree test cases: $n = 100/t = 5000$, $n = 1000/t = 500$ and $n = 10000/t = 50$, where only the population size (n) and the number of generations (t) are modified, whereas the other specific parameters of algorithms remain constant. In these tables, the cells containing the best value for each problem have a grey colored background.

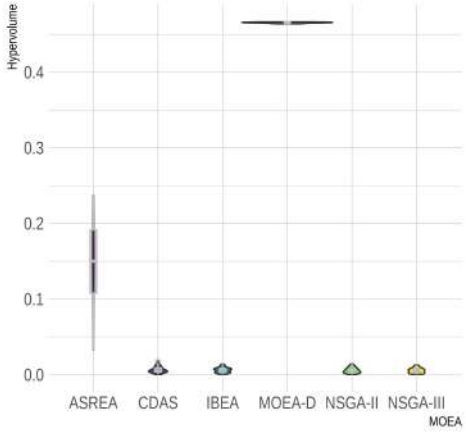
Referring to Table 4.1, it can be concluded that MOEA-D demonstrates the best scalability w.r.t. the population size: it improves of the hypervolume value with increasing the population size under the same computational budget for all functions. Whereas, NSGA-II, NSGA-III and CDAS could improve their hypervolume values working with very large population size only on two test functions DTLZ2 and DTLZ4. It means that NSGA-II, NSGA-III and CDAS can work efficiently with large population size only on uni-modal modal functions (as it was precised in Section 3.5.2: DTLZ1 and DTLZ3 - multi-modal functions, DTLZ2 and DTLZ4 - uni-modal functions). Apparently, for multi-modal functions, they require larger number of generations, which is not desirable for parallel implementation. We can state that an archive-based algorithm, ASREA, has an ability to work with very large population size on multi-modal and uni-modal functions. However its obtained value of hypervolume is smaller (i.e., worse) than MOEA-D and it suffers from high variance of the results. It can be clearly seen in Figure 4.1, where the experimental results by each algorithm with population size $n = 10000$ at $t = 50$ number



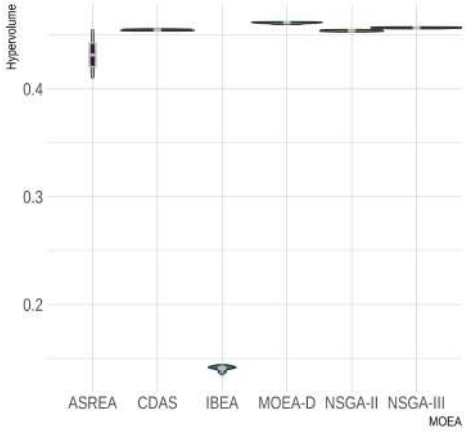
a) DTLZ1



b) DTLZ2



c) DTLZ3



d) DTLZ4

Figure 4.1: Average and standard deviation of the hypervolume indicator I_{HV} of benchmarked algorithms on DTLZ test suite w.r.t. the population size $n = 10000$ at the number of generator $t = 50$ (the computational budget $EF = 500000$) for $m = 3$.

Table 4.1: Average value of hypervolume indicator $\overline{I}_{HV}$ obtained by the benchmarked algorithms on 3-objective problems of the DTLZ test suite w.r.t. the different population sizes the computational budget $EF = 500000$.

Problem	Pop.Size	NSGA-II	NSGA-III	CDAS	MOEA-D	IBEA	ASREA
DTLZ1	100	75.8e-02	71.8e-02	72.8e-02	73.7e-02	71.8e-02	75.4e-02
	1000	81.1e-02	81.0e-02	81.0e-02	80.6-02	81.0e-02	73.4e-02
	10000	0	0	0	82.3e-02	0	33.1e-02
DTLZ2	100	37.1e-02	37.0e-02	37.5e-02	37.5e-02	42.0e-02	37.1e-02
	1000	44.2e-02	42.9e-02	44.2e-02	44.3e-02	45.3e-02	39.4e-02
	10000	46.3e-02	46.0-02	46.2e-02	46.7e-02	44.3e-02	43.7e-02
DTLZ3	100	37.7e-02	25.5e-02	38.4e-02	37.5e-02	0	37.5e-02
	1000	44.3e-02	42.4e-02	44.1e-02	44.4e-02	0	23.9e-02
	10000	0	0	0	46.6e-02	0	15.8e-02
DTLZ4	100	37.5e-02	36.5e-02	38.5e-02	0	0	37.1e-02
	1000	44.1e-02	43.8	44.1e-02	43.9e-02	21.1e-02	40.0e-02
	10000	45.4e-02	45.7e-02	45.6e-02	46.2e-02	14.1e-02	43.6e-02

Table 4.2: Average computation time (in seconds) of benchmarked algorithms on first four functions of DTLZ test suite w.r.t. the different population sizes for $m = 3$ and the computational budget $EF = 500000$.

Problem	Pop.Size	NSGA-II	NSGA-III	CDAS	MOEA-D	IBEA	ASREA
DTLZ1	100	2.7	4.1	4.1	2.4	4.1	2.6
	1000	14.2	17.2	17.2	11.4	17.2	2.7
	10000	51.2	58.2	58.2	116.2	1432.5	3.0
DTLZ2	100	3.5	5.6	3.8	3.3	13.4	3.3
	1000	23.3	27.1	23.6	11.8	125.7	3.8
	10000	200.6	198.1	199.6	113.9	1772.6	3.8
DTLZ3	100	3.4	4.4	3.3	3.7	13.2	3.1
	1000	12.4	13.8	12.9	12.3	115.2	3.5
	10000	53.4	59.9	54.7	89.6	1532.4	3.8
DTLZ4	100	3.6	4.8	3.9	3.5	10.2	3.1
	1000	22.6	25.1	24.1	12.3	131.1	3.9
	10000	157.8	165.7	173.2	110.1	1925.5	4.2

of generations for each function over 50 runs, including mean and standard deviation of the hypervolume value, are presented. As it is shown in Figure 4.1, ASREA achieve higher performance than NSGA-II, NSGA-III, CDAS and IBEA on DTLZ1 and DTLZ3 functions, but has a big variance, especially on multi-modal functions.

Regarding total runtime, from Table 4.2 it is obvious that ASREA demonstrates significant advantage comparatively with the other MOEAs: its average value of total runtime with the very large population size $n = 10000$ is around 4 seconds, which is around in 20 times faster than total runtime of MOEA-D and in 40 times faster than NSGA-II, NSGA-III, CDAS.

Among all tested MOEAs, we selected Archived-based Stochastic Ranking Evolutionary Algorithm (ASREA) [Sharma & Collet (2010*a*)] as the baseline algorithm for solving in parallel different time-consuming multi- and many-objective problems of the model of Active Magnetic Regenerator (AMR). This choice can be explained by the following reasons:

1. ASREA has low computational complexity $\mathcal{O}(man)$, where m is the number of objectives, a is the size of archive and n is the population size.

2. ASREA demonstrates very small value of total runtime comparatively the other MOEAs, due to its low computational complexity.

3. ASREA can ensure the population scalability, but with high variance of the results.

4. ASREA has a structure, which can be easily modified.

In the next section, we will investigate and analyze the design and performance of ASREA in the context of optimization of the simulation model of Active Magnetic Regenerator.

4.1.3 Analysis of ASREA

Though ASREA has a low computational complexity and consequently, demonstrates a small value of total runtime on different test functions (see Table 4.1 and Table 4.2), the population scalability is a concern, which wasn't investigated in the original paper [Sharma & Collet (2010*a*)]. As it has been seen in Section 4.1.2.1 from the results of the comparative experiment on Deb–Thiele–Laumanns–Zitzler (DTLZ) [Deb et al. (2002)] test suite aiming at investigating the population scalability of different MOEAs, ASREA shows a high variance of hypervolume value for the population size $n = 10000$ (see Figure 4.1). For better visualize the results, Figure 4.2 shows Pareto fronts obtained by two runs of ASREA with different seed values on DTLZ1 function.

Consequently, the population scalability has to be further improved. Below, we analyzed Algorithm 1, in order to improve the population scalability of ASREA and makes it fit to the other possible particularities of the optimization problems of the Active Magnetic Regenerator (AMR) model: e.g., the scalability w.r.t. the dimensions of the objective (target) space, the scalability w.r.t. the dimensions of the search space and the efficiency against the negative impact of dominance resistant solutions. At least the four following important potential weaknesses of ASREA have been identified:

1. **Ranking assignment based on Pareto dominance:**

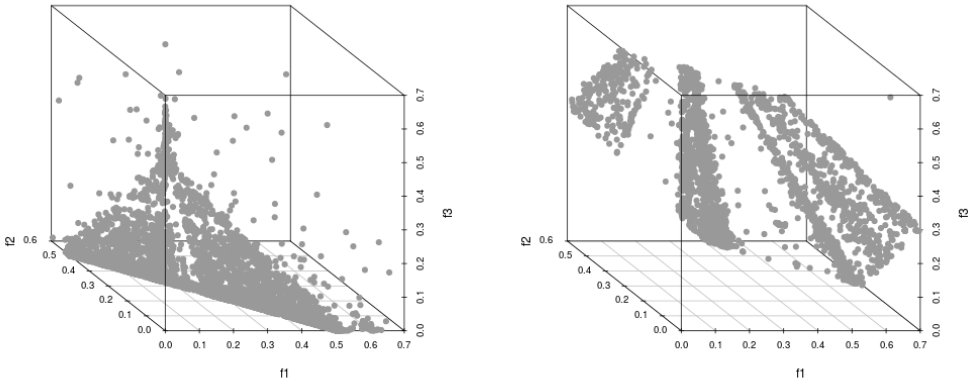


Figure 4.2: Pareto Fronts for DTLZ1 ($a = 60, d = 7, m = 3, n = 10000, t = 50$) problem obtained by ASREA with different values of seed.

- The ranking assignment by Pareto dominance might not be effective with growing of the dimension of objective (target) space (i.e., $m > 2$). Indeed, with increasing the number of objectives, the characteristics of multi-objective landscapes viewed in terms of non-dominated fronts can be changed: the number of fronts reduces and each front has more solutions [Aguirre & Tanaka (2007)], [Sato et al. (2007)], [Purshouse & Fleming (2003)]. In other words, by increasing the number of objectives, almost all solutions become non-dominated. It means that many sampled solution of the current generation do not dominate each other and the efficiency of the Pareto selection weakens. According to the requirements to an algorithm for solving the problems of the model of AMR, a MOEA has to provide scalability w.r.t. the dimension of target space. Consequently, in order to solve many-objective problems (i.e., $m > 3$), this weakness has to be improved.
- Pareto dominance-based algorithms are not efficient for solving the problems with so called Dominance Resistant Dolutions (DRSs). It means that some solutions can have extremely good values for some objectives and extremely bad values for other objectives, which is the case for the optimization problem of the AMR model. Obviously, the presence such kind of solutions degrades the search ability of Pareto dominance operator, because the extremely good values of objective functions make DRSs non-dominated by other solutions.

2. **Archive size:**

- Finding the appropriate size of archive, which allows not only to accommodate and evenly spread non-dominated solutions, but also to guarantee the convergence to the approximate Pareto front, is a difficult task, because it depends on many factors such as number of objectives, population size and etc. [Sharma & Collet (2010*a*)]. We assume that the small archive size, i.e., $a = 10 \cdot m$, can be a cause of the high variance of the value of hypervolume indicator, because of its influence on the ranking assignment procedure, where the children solutions are only compared with the solutions from the archive. Thus, in this

case, to maintain a good distribution of solutions with a large population size n becomes difficult. If our assumption is correct, the archive size has to be increased, but without significant loss of the computational complexity.

- As the number of non-dominated solutions is limited by the small size of the archive: i.e., $a = 10 \cdot m$, the final front contains not only non-dominated solutions [Sharma & Collet (2010*a*)]. This fact can explain the degradation of the value of hypervolume indicator with growing the population size presented in Table 4.1, because the difference between the archive size and population size is increased. Consequently, a mechanism to increase the number of non-dominated solutions in the Pareto front is required, but without significant loss of the computational complexity for single generation.

3. **Crowding distance operator:** ASREA invokes the target space-based crowding distance operator in archive update operator and in the selection strategy, when one among the solutions with the same rank must be selected. Although this operator is widely used in many MOEAs as NSGA-II [Deb et al. (2000)], it can be useless, when one efficient point in the objective space can correspond to more than one non-dominated solutions in the search space. In this case, the solutions with identical or close values of objective functions can be lost (only if they are not extreme solutions) because of their insignificant values of crowding distance. Taking into account that such kind of situations is quite common for a number of real-world problems [Hiroyasu et al. (2005)], [Preuss et al. (2010)], [Kudo et al. (2011)] and their appearance during the optimization of the dual-mode operating AMR model is confirmed by our preliminary investigations of the model, this issue has to be improved.

4. **Variation Operators:**
   As it is seen from the Algorithm 1, ASREA creates a child population $\boldsymbol{X}^t$ through two operators:

   - Simulated Binary crossover (SBX) presented in [Deb et al. (1995)]. Like most crossover operators, it works with two parent individuals and produce two children individuals. This operator simulates the binary operator in conventional genetic algorithms.
   - Polynomial mutation presented in [Deb & Deb (2014)]. In this operator, a polynomial probability distribution is used to perturb a solution in a parent's vicinity. The probability distribution in both left and right of a variable value is adjusted so that no value outside the specified range $[a, b]$ is created by the mutation operator, where a and b are lower and upper bounds of the variable.

   Both operators are separable variation operators, which are efficient mostly on separable functions as ZDT test suite [Sharma & Collet (2010*a*)], where they quickly explore search space. On the other hand, because of this separable property ASREA demonstrates a significant difficulty when it deals with non-separable functions.

   However, in the frame of this thesis, we have to consider the optimization problems: (i) as a black-box problem for the model of AMR and (ii) as separable or partially separable problems for the model of Hamiltoniam for simulating physical properties

of magneto caloric materials. Taking both cases into account, we try to find a mutation operator, which can be more efficient on both, separable and non-separable, problems.

To summarize, the structure of ASREA has to be modified in order to improve the identified weaknesses and the population scalability without a significant increasing of computational complexity for single generation.

4.2 FastEMO Algorithm

In this section we present the design of the improved version of ASREA, called FastEMO, in which we try to eliminate the weaknesses of its predecessor, ASREA, while trying to maintain a computational complexity close to $\mathcal{O}(man)$. As ASREA, FastEMO is an archive-based algorithm and stores the set of non-dominated solution in an archive $\boldsymbol{A}^t$ and generally, it has the same structure as ASREA. The detailed description of FastEMO is presented below.

Description of FastEMO: As ASREA, FastEMO starts with a randomly initial population $\boldsymbol{P}^{t=0}$ and an empty archive $\boldsymbol{A}^{t=0}$, evaluates each solution $\boldsymbol{p}^{t=0} \in \mathbb{R}^{\mathrm{d}}$ (where d is the dimension of search space) in $\boldsymbol{P}^{t=0}$ and copies all non-dominated members of $\boldsymbol{P}^{t=0}$ to $\boldsymbol{A}^{t=0}$. Then, the steps presented by the pseudo-code in Algorithm 5 are performed for each generation t. Note that the pseudo-code of FastEMO uses the same notations, which are summed up in Table 3.1 (see Section 3.4.1). One new notation S is introduced in FastEMO, which specifies the control parameter of expansion or contraction of the dominance area of solutions. This control parameter is defined by the user and its role and impact on the performance will be explained further in Section 4.2.1).

As it is shown in Algorithm 5, FastEMO has the following modifications:

1. **line 4:** The mutation operator based on a Cauchy distribution (detailed in Section 4.2.5), replaces the polynomial mutation operator in order to:
   (a) exploit the heavy tails of the Cauchy distribution for improving: (i) the diversity of solutions; (ii) the performance on multi-modal problems and (iii) the convergence speed;
   (b) investigate the efficiency of Cauchy distribution-based mutation on: (i) separable problems; (ii) non-separable problems.
2. **line 6:** The operator for converting the objective values (detailed in Section 4.2.1). This operator is based on the technique of Control the Dominance Area of Solutions (CDAS) [Sato et al. (2007)] and is introduced in order to improve the inefficiency of the conventional Pareto dominance on many-objective problems.
3. **lines 8-10:** At the last generation: (i) the archive size is increased to the population size (discussed in Section 4.2.3), in order to improve the value of the hypervolume indicator and obtain a larger number of non-dominated solutions in the final front; (ii) the control parameter of expansion/contraction of the dominance area of solutions ($\boldsymbol{S}$) is set to 0.5, in order to to obtain "correct" (conventional) final Pareto front.

Algorithm 5: FastEMO: Pseudo-code of procedure at generation t.

Input: $\boldsymbol{P}^t$, $\boldsymbol{S}$
Result: $\boldsymbol{P}^{t+1}$
1 **for** $k = 0; k < n; k = k + 1$ **do**
2 Select randomly parent solutions $\boldsymbol{p}_i^t$ and $\boldsymbol{p}_j^t$ from $\boldsymbol{P}^t$
3 $\boldsymbol{x'}_k^t = \text{SBX}(\boldsymbol{p}_i^t, \boldsymbol{p}_j^t)$; // Apply SBX Crossover
4 $\boldsymbol{x}_k^t = \text{Mut}_{CAUCHY}(\boldsymbol{x'}_k^t)$; // Apply Cauchy-based Mutation with non-state mutation probability (Alg. 8)
5 EVAL($\boldsymbol{x}_k^t$); //Evaluate $\boldsymbol{x}_k^t$
6 CONVERT$_{CDAS}(\boldsymbol{x}_k^t, \boldsymbol{S})$; // CDAS-Converting of objective values (Eq. 4.6)
7 **end**
8 **if** $t == T - 1$ **then**
9 $a = n; \boldsymbol{S} = 0.5;$
10 **end**
11 **for** $k = 0; k < n; k = k + 1$ **do**
12 RANK$_{CDAS}(\boldsymbol{x}_k^t, \boldsymbol{A}^t)$ // CDAS-based ranking of child $\boldsymbol{x}_k^t$ w.r.t. $\boldsymbol{A}^t$
13 ACD$(\boldsymbol{x}_k^t, \boldsymbol{A}^t)$ // Alternative Crowding Distance assignment (Alg. 6)
14 $\boldsymbol{A}^{t+1} = \text{UPDATE}_{CDAS+ACD}(\ \boldsymbol{x}_k^t, \boldsymbol{A}^t, a)$; // Archive updates (Alg. 7)
15 **end**
16 $\boldsymbol{P}^{t+1} = \text{SELECT}(\boldsymbol{X}^t, \boldsymbol{A}^t))$// Selection strategy to new population:

4. **line 12:** The ranking assignment procedure works by the same algorithm as in ASREA, but it uses the objective values, which are converted by the CDAS technique, which is detailed in Section 4.2.1. This technique transforms the conventional Pareto dominance to the expanding/contracting dominance for: (i) providing the scalability w.r.t. the number of objectives, (ii) improving diversity of candidate solutions due to the "non-so-good" candidate solutions can obtain a good rank, (iii) fixing the problem of dominance resistant solutions.

5. **line 13:** The classical objective space-based crowding distance operator is replaced by Alternative Crowding Distance (ACD) [Deb & Tiwari (2008)] detailed in Section 4.2.2), for handling the solutions, which are identical on the objective (target) space, but different on the search space;

6. **line 14:** The archive updating operator (detailed in Section 4.2.4) uses the same algorithm as ASREA, but employs the CDAS-based dominance and the ACD operator.

7. **line 16:** The selection strategy uses by the same algorithm as ASREA, but employs the CDAS-based dominance and the ACD operator.

The sections below explain these modifications.

4.2.1 Conversion Operator

In this section, we present a conversion operator (line 6 in Algorithm 5) for enhancing the effectiveness of the Pareto dominance that is required in order to provide scalability w.r.t.

the objective space and consequently, to solve many-objective problems. Indeed, for two random solutions with m objectives, the probability that one solution dominates the other one is $\left(\frac{1}{2}\right)^{(m-1)}$. Since it is rare for one solution to strictly dominate the other one in high-dimensional space, a technique, which can modify the objective values of solutions and consequently, change the the selection pressure and the dominance probability is required. For this purpose, the technique, which modifies the values of objective functions, called Controlling Dominance Area of Solutions (CDAS) is borrowed from [Sato et al. (2007)].

In this section, we, first, describe in details the CDAS technique mainly based on the original paper [Sato et al. (2007)] and then, explain how it is integrated in FastEMO in order to solve many-objective problems.

4.2.1.1 Controlling Dominance Area of Solutions

In order to ensure the scalability w.r.t. the target space and consequently, to solve not only multi-objective, but also many-objective problems, the effectiveness of the Pareto dominance on many-objective functions has to be improved. For this purpose, we apply the technique called Controlling Dominance Area of Solutions (CDAS) and originally presented in [Sato et al. (2007)], which contracts and expands the dominance area of each candidate solution by a specified angle for each objective and by modifying the values of each objective value.

The mechanism for modifying the values of objective function goes from trigonometry and based on the Law of Sines[1], which defines the relationship between the sides and angles of triangles.

According to the Law of Sines:

$$\frac{r}{\sin(\varphi_i)} = \frac{f_i'(\boldsymbol{x})}{\sin\big(\pi - (\omega_i + \varphi_i)\big)} \tag{4.3}$$

where $\varphi_i = S_i \cdot \pi$ is an angle of control of modification, $i \in [1, m]$, m is dimension of target space (number of objectives), S_i is a user-defined parameter of contraction/expansion of the dominance area for i-th objective, r is the norm of vector $\boldsymbol{f}(\boldsymbol{x})$, which can be calculated by the following equation:

$$r = \sqrt{(f_1(\boldsymbol{x})^2 + ... + f_m(\boldsymbol{x})^2)} \tag{4.4}$$

.

where $\boldsymbol{x}$ is a solution in the search space (vector of decision variable),
$f_i(\boldsymbol{x})$ is a value of i-th objective for the solution $\boldsymbol{x}$,
$f_i'(\boldsymbol{x})$ is a value of modified i-th objective for the solution $\boldsymbol{x}$,
$\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}))$ is a vector of objectives for the solution $\boldsymbol{x}$,
$\boldsymbol{f'}(\boldsymbol{x}) = (f_1'(\boldsymbol{x}), f_2'(\boldsymbol{x}), \ldots, f_m'(\boldsymbol{x}))$ is a vector of modified objectives for the solution $\boldsymbol{x}$,
ω_i is the declination angle between $f_i(\boldsymbol{x})$ and $\boldsymbol{f}(\boldsymbol{x})$, which can be calculated with the following equation:

$$\omega_i = \arccos\Big(\frac{f_i(\boldsymbol{x})}{r}\Big) \tag{4.5}$$

.

[1] https://en.wikipedia.org/wiki/Law_of_sines

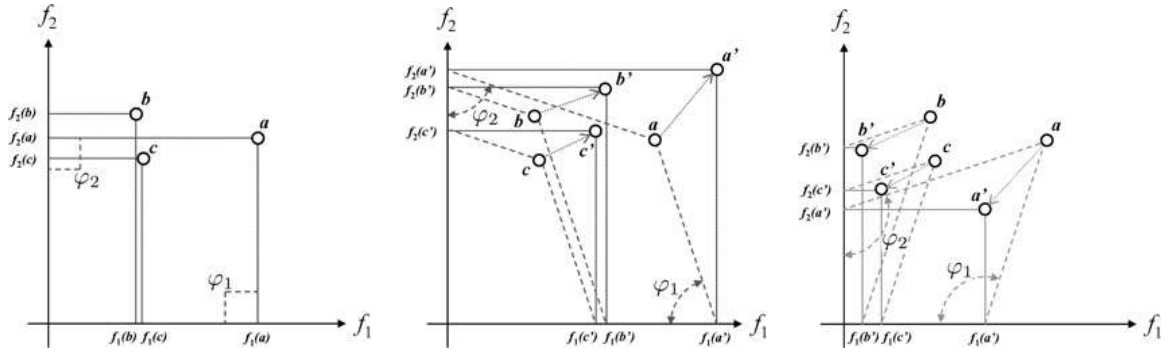
Consequently, $f_i'(\boldsymbol{x})$ can be defined as follows:

$$f_i'(\boldsymbol{x}) = \frac{r \cdot \sin(\omega_i + \varphi_i)}{\sin(\varphi_i)} \tag{4.6}$$

Equation 4.6 describes the working principle of CDAS: a contraction and expansion of the dominance area of solutions is organized by modifying the values of the objective functions by changing the angle φ through a control parameter S. Note that S can be defined separately for each objective: S_i (where $i \in [1, m]$). In order to bring out desirable search performance, the user has to experimentally find out S_i values that control the dominance area appropriately.

According to Equation 4.6, CDAS keeps the conventional Pareto dominance when $S_i = 0.5$. It means that the value of angle $\varphi_i = \frac{\pi}{2}$ does not change the value of the objective function $f_i(\boldsymbol{x})$: $f_i'(\boldsymbol{x}) = f_i(\boldsymbol{x})$. Consequently, the user can increase or decrease a value of objective function $f_i'(\boldsymbol{x})$ by setting the value of $S_i < 0.5$ ($\varphi_i < \frac{\pi}{2}$) or $S_i > 0.5$ ($\varphi_i > \frac{\pi}{2}$) respectively.

Let us show an example from [Sato et al. (2007)] for bi-objective space. In Figure 4.3 (a), the solution $\boldsymbol{c}$ is dominated by $\boldsymbol{a}$, but the pairs of solutions ($\boldsymbol{a}$,$\boldsymbol{b}$), and ($\boldsymbol{b}$,$\boldsymbol{c}$) are non-dominated by each other according to the conventional Pareto dominance rule. In this case $f_i'(\boldsymbol{x}) = f_i(\boldsymbol{x})$ and $S_1 = S_2 = 0.5$. But if we want to expand the dominance area, we have to modify the value S_i as $S_i < 0.5$, which changes the objective value $f_i'(\boldsymbol{x}) > f_i(\boldsymbol{x})$, so that the Pareto front will be produced as it is shown in Figure 4.3 (b), where solution ***a'*** dominates ***b'*** and ***c'***. Otherwise, for contracting the dominance area, we have to increase the value of S_i as $S_i > 0.5$ to modify the objective value $f_i'(\boldsymbol{x}) < f_i(\boldsymbol{x})$, so that no one solution will be dominated (see Figure 4.3 (c)).



a)$S_1 = S_2 = 0.5$ Conventional dominance — b) $S_1 = S_2 < 0.5$ Expanding dominance — c) $S_1 = S_2 > 0.5$ Contracting dominance

Figure 4.3: An impact of S_i value on the dominance area of solutions. The illustration is adapted from [Sato et al. (2007)].

Paper [Sato et al. (2007)] shows that if we deal with many-objective problems, we have to expand the area of dominance by decreasing the value of the control parameter S below 0.5. As demonstrated in [Sato et al. (2007)], the maximum expansion of dominance area was obtained with the value of the control parameter $S = 0.25$, where each front has only one solution. And visa versa, the maximum contraction of the dominance area was obtained with $S = 0.75$, where all solutions are contained in one front.

To summarize, by changing parameter S and consequently, by the controlling the dominance area of solutions, we convert the values of objectives and induce different

projected objectives spaces in which the algorithm provides different functionality: i.e., diversity or convergence.

Even though in [Sato et al. (2007)] it was shown that significantly better performance can be achieved on the convergence and diversity of solutions obtained by CDAS rather than by using conventional Pareto dominance, this efficiency highly depends on the value of control parameter S. This value has to be defined by taking into consideration the number of objectives, the number of decision variables and the complexity of the problem. Consequently, the value of S has to be defined experimentally. It is an important disadvantage of the CDAS technique, because the value of S is not easily determinable when facing different problems.

4.2.1.2 Computational Complexity of CDAS

The computational complexity to convert the dominance with the CDAS technique is $\mathcal{O}(mn)$, where m is the number of objectives and n is the population size.

4.2.1.3 CDAS Integration in FastEMO

In FastEMO, the evaluated values of the objectives of each child solution $\boldsymbol{x}_i^t$ (where $i \in [1, n]$ and n is the population size) from the child population $\boldsymbol{X}^t$ are converted by the CDAS technique according to the value(s) of control parameter ($\boldsymbol{S}$) in the main loop of Algorithm 5 (at line 6). As explained earlier, this conversion modifies the conventional Pareto dominance relations between solutions.
Then, we apply these new relations:

1. in the ranking assignment procedure (Algorithm 5, line 12);
2. in the archive updating procedure (Algorithm 7) in the dominance operator $\succ$ and $\prec$ (at line 5 and 9), aiming at selecting the non-dominated solutions into the archive;
3. in the selection strategy (Algorithm 5, line 16), aiming at selecting the best solution between two candidates for the parent population;

Such kind of control of the dominance area can preserve diversity, because depending on the value of $\boldsymbol{S}$, the solutions of the child population $\boldsymbol{X}^t$ that are originally dominated by others can become non-dominated and be selected in the archive.

We treat the control parameter of the dominance area $\boldsymbol{S}$ as a m-dimensional vector, where m is the number of objectives. The choice of the value(s) of the control parameter $\boldsymbol{S}$ can be different. $\boldsymbol{S}$ can be a vector of a constant value(s) or varying during the optimization process. Moreover, they can be defined as the same value for all objectives ($i = 1, 2, ...m$) or as different values for each objective.

In our work, we propose to uniformly randomly choose the value of S_i for each i at each generation t in the experimentally found ranges. The ranges are defined by taking under consideration only the number of objectives and are presented in the Table 4.3.

To summarize, the CDAS-based converting operator provides the following useful features:

- it can manage the size of the non-dominated set and regulate convergence that improves of the scalability w.r.t. the number of objectives, especially when the proportion of locally non-dominated solution is growing while the number of objectives increases;

Table 4.3: The experimentally defined ranges of the possible values of control parameter of the dominance area in dependence on the number of objectives.

Number of objectives m	Range of control parameter S_i, $i \in [1, .., m]$
2	[0.495, 0.505]
3	[0.49, 0.5]
4	[0.485, 0.5]
5	[0.47, 0.49]

- it allows the algorithm to keep some inferior solutions in the archive so as to preserve diversity.

In future works, a more sophisticated scenario has to be considered in order to modify the value of the control parameter according to some performance feedback criteria.

4.2.2 Alternative Crowding Distance Operator

The standard objective space-based Crowding Distance (CD) presented in [Deb et al. (2000)] is one of the mostly used operator for preserving diversity in Multi-Objective Evolutionary Algorithms (MOEAs). It is usually applied to select one from two feasible solutions being non-dominated to each other by the comparison of the crowding distance value on the objective (target) space, where the solution with the larger crowding distance values wins. However, in some cases, it can be inefficient. E.g., when one point in the objective space can correspond to more than one non-dominated solutions in the search space, in this case some interesting solutions can be lost [Deb & Tiwari (2008)].

According to our preliminary tests on the model of Active Magnetic Regenerator (AMR), it was found that depending on the configuration of optimization problem, the different combinations of control and design parameters of the model can be very close to each other on the objective space. In order to take it into account, we decided to replace the crowding distance operator used in the archive updating procedure of ASREA, by an alternative technique, which can maintain the diversity of solutions by controlling solutions not only in the objective space, but also in the search space. This is important for an effective exploration of the design of the dual-mode operating AMR model.

As an appropriate alternative technique, we selected the Alternative Crowding Distance (ACD) procedure originally presented in [Deb & Tiwari (2008)]. The pseudo-code of the ACD procedure is presented in Algorithm 6, which is borrowed from the original paper [Deb & Tiwari (2008)]. Note that this pseudo-code uses the same notations, which are summed up in Table 3.1 (see Section 3.4.1). We integrate several additional notations: $|\boldsymbol{A}|$ is current archive size, cd_i^{obj} is the value of target space-based CD of current solution i, cd_i^{var} is the value of search space-based CD of current solution i, cd_{avg}^{obj} is the average value of target space-based CD, cd_{avg}^{var} is the average value of search space-based CD, $norm_i^{obj}$ is the normalized value of a solution i for j-th objective and $norm_i^{var}$ is the normalized value of a solution i for j-th variable.

According to [Deb & Tiwari (2008)], in order to compute the normalized value of objective space-based CD of solution i for the j-th objective, we first sort the population

members in increasing order of the objective value and then apply Equation 4.7:

$$norm_i^{obj} = \frac{f_j(\text{right of } i) - f_j(\text{left of } i)}{(f_j^{max} - f_j^{min})} \tag{4.7}$$

Similarly, the normalized value of a search space-based CD of a solution i for the j-th variable is calculated as it is presented in Equation 4.8:

$$norm_i^{var} = \frac{x_j(\text{right of } i) - x_j(\text{left of } i)}{(x_j^{max} - x_j^{min})} \tag{4.8}$$

For two solutions, which are non-dominated to each other, it is possible that they can have identical objective function values, thereby making the cd^{obj} values to be zero (unless they are the extreme solutions). However their cd^{var} values will be non-zero. As it seen in Algorithm 6 in the lines 37-43, since a solution's crowding distance value is chosen as the maximum of the two crowding distance values, these solutions will inherit the cd^{var} values if they are more diverse in the population. Thus, non-dominated solutions can survive due to their sparsity either in the objective space or in the decision variable space [Deb & Tiwari (2008)].

4.2.2.1 Computational Complexity of ACD

As the ACD procedure involves not only computations of the values of classical target space-based crowding distance, but also the values of search space-based crowding distance, the computational order to the ACD procedure for single generation is $\mathcal{O}(mn\log(n)) + \mathcal{O}(dn\log(n))$ computations, where m is the number of objectives, d is the number of decision variables, n is the population size.

Due to the computation of the search space-based crowding distance, the computational complexity of the ACD procedure is slightly larger than the classical crowding distance operator, which is used in ASREA.

4.2.2.2 Integration of ACD in FastEMO

In FastEMO, the value of the Alternative Crowding Distance (ACD) is calculated for each child solution $\boldsymbol{x}_i^t$ (where $i \in [1, n]$ and n is the population size) from the child population $\boldsymbol{X}^t$ at each generation t in the main loop of Algorithm 5 (at line 13). Then, the values of ACD are used:

1. in the archive updating procedure provided in Algorithm 7, when the archive is full (lines 17-20), in order to remove the worst ACD-wise solution;

2. in the selection strategy (Algorithm 5, line 16) when two solutions with the equal ranks are selected: the solution with larger ACD value will be selected.

To summarize, the ACD operator provides the required feature for solving the optimization problems of the model of Active Magnetic Regenerator: i.e., it allows the algorithms to keep some interesting solutions, when two solutions with the same ranks are similar on the objective space, but different on the search space. However, it increases the total computational complexity.

Algorithm 6: Alternative Crowding Distance.

Input: **A** - *archive of non-dominated solutions*
Result: **acd** - *alternative crowding distance calculated for each solution in the archive* $\boldsymbol{A}$

//initialize all distances to zero
for *each* $\boldsymbol{i} \in \boldsymbol{A}$ **do**
  $cd_i^{obj} = 0$ *// target (objective) space CD*
  $cd_i^{var} = 0$ *// search space CD*
end
//objective space crowding
for *j=1 to m* **do**
  for *each* $\boldsymbol{i} \in |\boldsymbol{A}|$ **do**
    if *i is a minimum solution in j-th objective* **then**
      $cd_i^{obj} = \infty$
    end
    else
      $cd_i^{obj} += norm_i^{obj}$
    end
  end
end
//variable space crowding
for *j=1 to d* **do**
  for *each* $\boldsymbol{i} \in |\boldsymbol{A}|$ **do**
    if *i is a boundary solution in j-th variable* **then**
      $cd_i^{var} += 2 \cdot norm_i^{var}$
    end
    else
      $cd_i^{var} += norm_i^{var}$
    end
  end
end
//normalize distance and compute population average
for *each* $\boldsymbol{i} \in |\boldsymbol{A}|$ **do**
  $cd_i^{obj} = cd_i^{obj}/m$
  $cd_i^{var} = cd_i^{var}/d$
end
$cd_{avg}^{obj} = \sum_{i=1}^{|\boldsymbol{A}|}(cd_i^{obj})/|\boldsymbol{A}|$
$cd_{avg}^{var} = \sum_{i=1}^{|\boldsymbol{A}|}(cd_i^{var})/|\boldsymbol{A}|$
//if above average, assign larger of the two distances,
//else assign smaller of the two distances,
for *each* $\boldsymbol{i} \in |\boldsymbol{A}|$ **do**
  if $cd_i^{obj} > cd_{avg}^{obj} || cd_i^{var} > cd_{avg}^{var}$ **then**
    $acd_i = max(cd_i^{obj}, cd_i^{var})$
  end
  else
    $acd_i = min(cd_i^{obj}, cd_i^{var})$
  end
end
return **acd**

4.2.3 Archive Size

As in ASREA, FastEMO uses an elite archive strategy to enhance its convergence speed and precision. However, finding the appropriate value of the archive size is an issue [Sharma & Collet (2010*a*)]. Comparatively with ASREA, where the value of the archive size is suggested as $10 \cdot m$, we integrated the following modifications:

- the archive size increases to the population size on the last generation (Algorithm 5, lines 8-10) in order to obtain more non-dominated solutions in the final Pareto front.
- the archive size is experimentally found for small ($n = 100$), large ($n = 1000$) and very large ($n = 10000$) population sizes by different tests on DTLZ [Deb et al. (2002)] and WFG [Huband et al. (2005)] test suites for problems with a number of objectives from 2 to 5, in order to ensure the better performance in terms of the value of the hypervolume indicator. The defined values of archive size depending on population size and number of objectives are presented in the Table 4.4.

Table 4.4: The experimentally defined values of the archive size (a) in dependence on the population size (n) and the number of objectives (m).

n	a
100	$10 \cdot m$
1000	$15 \cdot m$
10000	$20 \cdot m$

In Section 4.3.1.1, we confirm that too small archive size of ASREA, i.e., $a = 10 \cdot m$ is not appropriate for handling large population size and has an impact on the high variance of the obtained results. We show that the proposed sizes of the archive helps to improve the quality of obtained hypervolume indicator and reduces the variance, because larger number of good solutions can be included in the last front. However, the first modification increases the computational complexity at the last generation in an order of $\mathcal{O}(mn^2)$ computations, where m is the number of objectives and n is the population size.

4.2.4 Archive Updating Operator

In this section we present a pseudo-code of the archive updating operator in Algorithm 7, because it wasn't provided in the original paper of ASREA. This pseudo-code uses the same notations, which are summed up in Table 3.1 in Section 3.4.1.

Comparatively with ASREA, the main modifications in Algorithm 7 are:

1. **line 5 and 9:** the CDAS-based dominance is used instead of Pareto dominance, where the values of objective functions are modified according to Equation 4.6 w.r.t. the control parameter $\boldsymbol{S}$; in order to insert/reject new solution to/from the archive,
2. **lines 17-20:** if a new solution is non-dominated, but the archive is already full, the Alternative Crowding Distance (ACD) operator is used instead of the classical crowding distance operator, in order to estimate of the neighboring memberships

Algorithm 7: Pseudo code of Archive Updating Operator.

Input: $\boldsymbol{x}^t$ - *new solution*,
$\boldsymbol{A}^t$ - *archive to be updated*,
a - *size of archive*
Result: $\boldsymbol{A}^{t+1}$ - *archive after update*

```
if A^t == ∅ then
    A^t = {x^t}; // if archive is empty, new solution is inserted to archive
    Goto 23;
end
if x^t ≺ any a_i ∈ A^t then
    Goto 23; // if new solution is dominated by any member of archive it is
     rejected and not allowed to archive
end
for each a_i ∈ A^t do
    if x^t ≻ a_i then
        A^t=A^t/{a_i} // if new solution dominates one or several members of
         archive, all the dominated members are deleted from archive
    end
end
if size(A^t) < a then
    A^t=A^t∪x^t; // if archive is not full, new non-dominated solution is included
     to archive
    Goto 23;
end
s_max = max(acd); // Select solution with the largest ACD value
if acd(x^t) == acd(s_max) then
    Goto 23; // x^t is the same as s_max
end
A^{t+1} = A^t/{s_max} ∪ x^t; // if ACD value of new solution is better than the worst
 ACD value of archive member, new solution replaces this member
return A^t
```

between the member of archive and new solution in the space of objective and spaces of decision variable and to drop the worst ACD-wise solution from the archive.

.

4.2.5 Mutation Operator

The mutation operator can have a significant impact on the performance of evolutionary algorithms, because it ensures the maintenance of diversity in evolving populations [Deb & Deb (2014)]. Consequently, the choice of mutation operator is an important issue. Generally, the most commonly-used mutation operators are polynomial [Deb & Algorithms (2001)], [Deb & Deb (2014)], Gaussian distribution-based [Deb & Deb (2014)] and Cauchy distribution-based mutation operators [Hansen et al. (2006)], [Choi et al. (2020)], [Lee & Park (2014)].

The common property of all these operators is the exploitation of separability. It means that such kinds of operators are less efficient for problems having correlations between the decision variables of the solutions. Even though all these operators have the property of separability exploitation, their effectiveness differs from problem to problem. E.g., it is a common belief that the heavy tails of Cauchy distribution are more efficient comparatively with the exponentially decreasing tails of Gaussian distribution, especially for optimization of multi-modal problems, due to the fact that the long jumps can lead to better solutions [Yao & Liu (1997)], [Hansen et al. (2006)]. As well as in [Szu & Hartley (1987)], [Choi et al. (2020)], [Lee & Park (2014)] it was demonstrated that a Cauchy distribution can increase the convergence speed comparatively with a mutation based on a Gaussian distribution. But on non-separable functions the difference in performance provided by Cauchy and Gaussian distributions is negligible [Hansen et al. (2006)].

Although the design of FastEMO is mainly focused on solving the optimization problems of the model of Active Magnetic Regenerator, which are assumed as black-box problems, the separable and partially separable problems of the model of Hamiltonian of magneto caloric materials also have to be taken under consideration. Consequently, in this thesis, we do not aim at developing a new mutation operator, which can be invariant w.r.t. coordinate transformations of the search space and thus, be more efficient on non-separable functions with some correlations between variables. Instead, in this section, we propose a mutation operator developed for FastEMO, which is based on the following techniques:

- Cauchy distribution [Hansen et al. (2006)];
- non-static mutation probability [Doerr et al. (2017)].

We present below the main parts of this operator.

Cauchy Distribution:

The mutation operator presented in this section, uses an uni-variate Cauchy distribution, whose probability density function is defined in Equation 4.9 [Hansen et al. (2006)]:

$$f(x, \mu, \gamma) = \frac{1}{\pi} \frac{\gamma}{(x - x_0)^2 + \gamma^2} \tag{4.9}$$

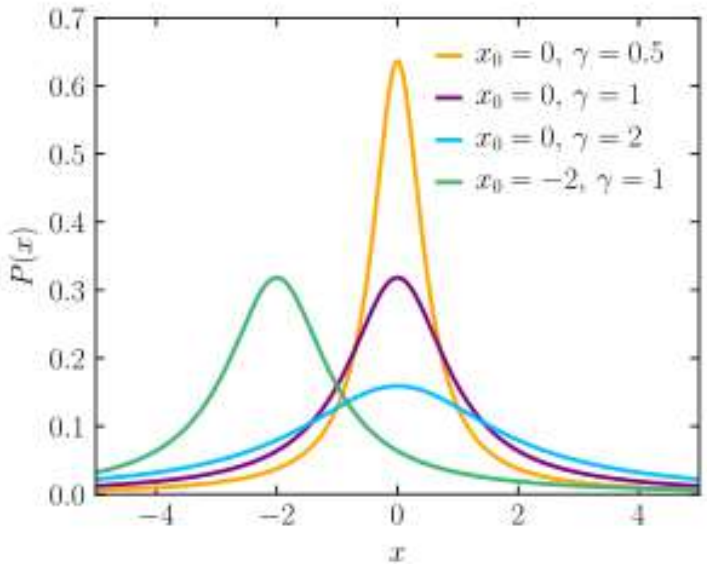


Figure 4.4: Probability density function of the Cauchy distribution. The illustration is adapted from `https://en.wikipedia.org/wiki/Cauchy_distribution`.

where x_0 is a location parameter, and γ is a scale parameter that determines the shape of the Cauchy distribution. Figure 4.4 visualises this function described by Equation 4.9. If a higher value is set to γ, the height of the peak of the probability density function will be shorter, and its width will be wider. If a lower value is assigned to γ, the height of the peak of the probability density function will be taller, and its width will be narrower.

Mutation Probability:

The choice of the mutation probability, p_m, can be a sensitive issue and mainly depends on the problem, which needs to be solved. Roughly, the mutation probability can be classified as:

- **static:**
  Commonly, a value of p_m is used to be assigned as $\frac{1}{d}$ (where d is the number of decision variables), so that on an average, one variable gets mutated per solution. A random number $u \in [0, 1]$ is created for every variable of each solution from child population, and if $u \leq p_m$ the variable is mutated using our Cauchy-based mutation operator. Such mutation probability gives asymptotically optimal expected optimization times for some simple uni-modal test problems [Doerr et al. (2017)]. However, it has been shown in [Doerr et al. (2013)] and [Doerr et al. (2017)] that such mutation probability is not ideal, and is far from optimal for some class of function, e.g., multi-modal problems. Furthermore, in [Doerr et al. (2017)] it has been demonstrated that any static mutation probability, like $\frac{1}{d}$ can give sub-optimal results on some functions: e.g., on most jump functions [Doerr et al. (2017)].

- **non-static:**
  A value of p_m is not fixed and changes according to defined rules for every solutions at each generation in order to improve the performance. The positive examples are shown and discussed in [Doerr et al. (2017)] and [Doerr et al. (2013)]. E.g., in [Doerr

et al. (2017)], a random mutation probability allows to obtain a performance close to the optimal one for multi-modal functions.

In the presented Cauchy mutation operator, we use a non-static mutation probability for the following reasons:

- Even if we find "optimal" parameter values of the mutation probability for one problem, these may be much different for similarly-looking problems, which is the basis for so-called parameter choice by analogy. It makes an application of static mutation probability very difficult, due to the fact that the model of Active Magnetic Regenerator and the model of Hamiltonian have different optimization problems.

- The best values of the mutation probability depend not only on the given problem but also on the current state of the optimization process and thus, have to be change over time.

Inspired by the non-static scheme based on the random mutation probability introduced in [Doerr et al. (2017)], the presented mutation operator also has a non-static probability $\frac{\beta}{d}$, where β is selected randomly for every solution from $\{1, ..., \frac{d}{2}\}$ according to the Pareto distribution whose probability density function is defined in Equation 4.10 [Hansen et al. (2006)]:

$$f(x) = \left\{ \begin{array}{rl} \frac{\alpha x_m^\alpha}{x^{\alpha+1}} & \text{for} \quad x \geq x_m \\ 0 & \text{for} \quad x < x_m \end{array} \right\} \tag{4.10}$$

where $x_m > 0$ is a scale (real value) and $\alpha > 0$ is a shape (real value).

Figure 4.5 visualises this function described by Equation 4.10.

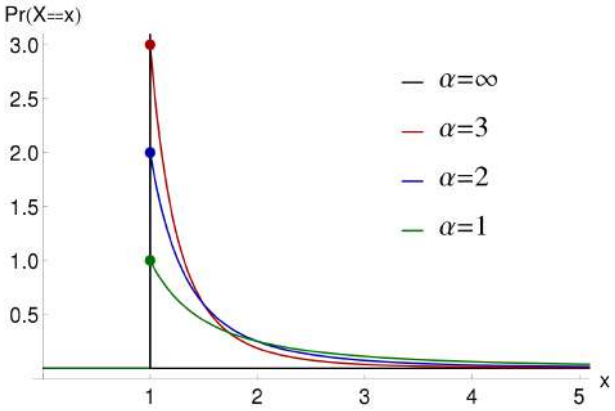


Figure 4.5: Probability density function of Pareto distribution for various α with $m = 1$. The illustration is adapted from `lhttps://en.wikipedia.org/wiki/Pareto_distribution`.

The idea to use the Pareto distribution is also borrowed from [Doerr et al. (2017)] in order to take a benefit from "power of power law". As we can see from Figure 4.5, the probability density function of a Pareto distribution (also called a power law distribution) has few extreme values on the left hand side of the curve and a very long tail of much less popular values on the right hand side of the curve. A Pareto distribution does not

show a well-behaved mean or variance. A power law, therefore, has no average that can be assumed to represent the typical features of the distribution and no finite standard deviations upon which to base confidence intervals. Such a distribution ensures that the number of mutated variables is not strongly concentrated around its mean, which eases having jumps of all sizes in the search space [Doerr et al. (2017)].

Proposed Mutation Operator:

To summarize all together, the pseudo-code of the proposed operator is presented in Algorithm 8 that mutates d-elements of real-value solutions. This pseudo-code uses the same notations, which are summed up in Table 3.1 in Section 3.4.1.

Relatively to Table 3.1, the following additional notations are introduced in Algorithm 8: (i) the Pareto distribution D^{α}_{2d} on the range $[1..2d]$, where d is the number of decision variables and $\alpha \in \mathbb{R}_+$ is a shape value of the Pareto distribution; (ii) $\beta \in \mathbb{R}_+$ is the parameter of a non-static mutation probability randomly selected according to Pareto distribution D^{α}_{2d} from the range $[1..2d]$; (iii) $p_m = \frac{\beta}{d}$ is mutation probability; (iv) $\boldsymbol{\gamma}^t_i \in \mathbb{R}^{\mathrm{d}}$ is a non-static mutation step sizes for each solution for the i-th decision variable, which is produced by the division of two values: σ_i a randomly selected value according to the Pareto distribution D^{α}_{d} for the i-th decision variable and $2 \cdot m \cdot d$; (v) $\mathcal{C}$ is the Cauchy distribution; (vi) $\mathcal{U}$ is the Uniform distribution.

A new child solution $\boldsymbol{x'}^t$ is created by mutating each decision variable of the solution $\boldsymbol{x}^t$ independently, according to the Cauchy distribution with non-static probability $p_m = \frac{\beta}{d}$ (line 2-6 in Algorithm 8).

Algorithm 8: Pseudo code of Cauchy distribution based mutation operator with non-static mutation probability.

Input: $\boldsymbol{x}^t$ - solution to be mutated
Result: $\boldsymbol{x'}^t$ - mutated solution

```
1 Choose β ∈ [1..2d] randomly according to D^α_{2d};
2 for i = 0; i < d; i = i + 1 do
3 |   if U(0,1) · d ≤ β then
4 |   |   Choose σ_i ∈ [1..d] randomly according to D^α_d;
5 |   |   γ^t_i = σ_i / (2md)
6 |   |   x'^t_i = C(x^t_i, γ^t_i);
7 |   end
8 end
9 return x'^t
```

The parameter of the Pareto (power-law) distribution $\alpha > 1$ has a constant value, which is found experimentally and is used in all experiments: $\alpha = 2.5$.

In the experimental design of FastEMO, we investigate the efficiency of the proposed mutation operator and compare it against the polynomial mutation [Deb & Algorithms (2001)], which is used in the original design of ASREA [Sharma & Collet (2010*a*)], aiming at defining the most appropriate operator for ensuring a stable performance in different use-cases,

In the final design of FastEMO, we replaced the polynomial mutation operator, which was used in ASREA, by the mutation operator presented in Algorithm 8.

4.3 Experimental Validation

Experimental Objectives:

1. Validation of FastEMO design and the efficiency of its elements: the modification of the archive size, the proposed mutation operator and the final design of FastEMO with an integrated conversion operator.

2. Validation of FastEMO applicability for a parallel implementation for solving computationally intensive problems: i.e., the scalability w.r.t. the population size. We have to confirm the ability of FastEMO to reach the optimal front in small number of generations, by working with a large population size.

3. Investigation of other properties of FastEMO required for solving the optimization problems of the model of Active Magnetic Regenerator (AMR) and the model of Hamiltonian: (i) the scalability w.r.t. the dimension of target space m, as the optimization problem of the AMR model can be multi - and many-objective problem; (ii) the robustness w.r.t. separability, as the optimization problems of the model of Hamiltonian is separable or partially separable; (iii) the scalability w.r.t. the dimension of search space d, as the optimization problem of the AMR model potentially can be a large scale problem; (iv) the robustness w.r.t. different properties of the optimization functions: uni-modality, multi-modality, etc.

We do not separately validate the impact of the technique of Controlling Dominance Area of Solutions (CDAS) on the performance of the algorithm in this set of experiments, because it was carefully investigated and confirmed in the original paper [Sato et al. (2007)]. However, we investigate its impact for solving many-objective problems, when we benchmark the scalability of FastEMO w.r.t. the dimension of objective space.

The experimental validation of FastEMO proceeds according to the listed above objectives.

Algorithms and Parameters:

All algorithms were launched without further parameter tuning, the crossover, mutation operator and parameter settings are defined according to the suggested specification of the original papers of algorithms, which were mentioned above. The values of the used hyper-parameters of FastEMO and NSGA-III are presented in Table 4.5.

Test Environment:

- All algorithms have many minor unwritten specifications, which can have an impact on their performance. For this experiment, we use the source codes of all algorithms, implemented earlier for this purpose in the frame of this thesis in the new version of the platform EASEA 2.20, which serves as a software support (see Section 6.1.1.1).

- All the experiments have been conducted on an Intel(R) Pentium(R) CPU 4405U @ 2.10GHz 4 processors laptop via the platform EASEA version 2.20, using the C++ language with g++ 5.4.0 compiler.

Table 4.5: Parameter settings of peer MOEAs.

Parameter	FastEMO	NSGA-II
SBX probability p_c	0.9	0.9
SBX Distribution index η_c	20	20
Mutation probability p_m	β/d	$1/d$
Poly. mut. distribution index η_m	-	20
Cauchy mut. non-static parameter β	$\in \{1, ..., 2d\}$	-
Control parameter of CDAS S	$\in [0.495, 0.505]$, when $m = 2$ $\in [0.49, 0.5]$, when $m = 3$ $\in [0.485, 0.5]$, when $m = 4$ $\in [0.47, 0.49]$, when $m = 5$	-
Archive size a	see Table 4.4	-

4.3.1 Validation of the FastEMO Design

In this section, in order to validate the design of FastEMO, we modify ASREA step by step, according to the design of FastEMO:

- we change the archive size in accordance with the description in Section 4.2.3;
- we replace the polynomial mutation by the proposed Cauchy distribution based mutation operator presented in Section 4.2.5;
- we integrate the converting technique based on the technique of Controlling Dominance Area of Solutions (CDAS) and Alternative Crowding Distance operators.

4.3.1.1 Validation of the Modification of the Archive Size

First, we perform the validation of the modification of the archive size, which is presented in Section 4.2.3.

Experimental Objective: to validate a positive impact of the modified archive size on the performance in terms of the value of hypervolume indicator and its deviation.

Test Problems: the first four 3-objective problems ($m = 3$) from DTLZ test suites are employed. For more details, one can see Section 3.5.2.

Test Algorithm: ASREA with the default archive size ($a = 10 \cdot m$) *vs* ASREA with the modified archive size presented in Section 4.2.3.

Performance Metrics: $\overline{I}_{HV}$ - mean value of hypervolume indicator and standard deviation over 30 independent runs of each algorithm for each test case.

Table 4.6: Average value of the hypervolume indicator $\overline{I}_{HV}$ and the standard deviation (in brackets) obtained by ASREA with the modified archive size (ASREA$_{mod}$) and with the default archive size (ASREA$_{def}$) on the DTLZ test suite w.r.t. the different population sizes.

Problem	Pop.Size	ASREA$_{def}$	ASREA$_{mod}$
DTLZ1	100	$75.4e-02_{(1.1e-04)}$	$75.6e-02_{\textbf{(2.3e-05)}}$
	1000	$73.4e-02_{(1.0e-02)}$	$75.5e-02_{\textbf{(3.9e-04)}}$
	10000	$33.1e-02_{(6.1e-02)}$	$54.7e-02_{\textbf{(3.0e-03)}}$
DTLZ2	100	$37.1e-02_{(2.3e-03)}$	$37.2e-02_{\textbf{(3.1e-04)}}$
	1000	$39.4e-02_{(8.4e-03)}$	$42.5e-02_{\textbf{(8.3e-04)}}$
	10000	$43.7e-02_{(2.8e-02)}$	$45.5e-02_{\textbf{(1.6e-03)}}$
DTLZ3	100	$37.5e-02_{(1.3e-04)}$	$37.5e-02_{\textbf{(1.2e-04)}}$
	1000	$23.9e-02_{(9.1e-03)}$	$37.4e-02_{\textbf{(5.6e-04)}}$
	10000	$15.8e-02_{(1.1e-01)}$	$23.7e-02_{\textbf{(2.0e-03)}}$
DTLZ4	100	$37.1e-02_{(3.2e-03)}$	$37.5e-02_{\textbf{(6.7e-04)}}$
	1000	$40.0e-02_{(7.3e-02)}$	$42.2e-02_{\textbf{(9.2e-04)}}$
	10000	$43.6e-02_{(1.0e-01)}$	$45.2e-02_{\textbf{(5.1e-03)}}$

Simulation Settings:

- **computational budget:** it has fixed number of evaluation functions for all test cases, i.e., $EF = 500000$;
- **population size:** $n = 100$, $n = 1000$, $n = 10000$;
- **number of generations:** $t = 5000$, $t = 500$, $t = 50$;
- **number of objectives:** $m = 3$;
- **number of decision variables:** $d = 7$ for DTLZ1, $d = 12$ for DTLZ2-DTLZ4. The defined values are calculated for each test problem according to the rules and suggested values described in Section 3.5.2 and originally presented in [Deb et al. (2002)].
- **statistics:** each algorithm is run over 30 independent runs for each test case.

Experimental Results: The performance obtained by ASREA with the modified archive size and with default archive size ($a = 10 \cdot m$) is presented in Table 4.6. This table shows the comparative results in both test-cases on the DTLZ problem suite regarding the mean and standard deviation of the hypervolume values, where the gray background represents the best results of $\overline{I}_{HV}$ and the best value of standard deviation is presented on bold. Thus, we can estimate the impact of the modification of the archive size, proposed in Section 4.2.3 by the average value of $\overline{I}_{HV}$ and standard deviation.

We can observe from Table 4.6 that for a small population size $n = 100$, the average values of the hypervolume indicator ($\overline{I}_{HV}$) obtained in both cases are very similar. It is predictable, as the difference between the size of archive and the population size is not very large and consequently, the integrated modifications do not play a big role. However,

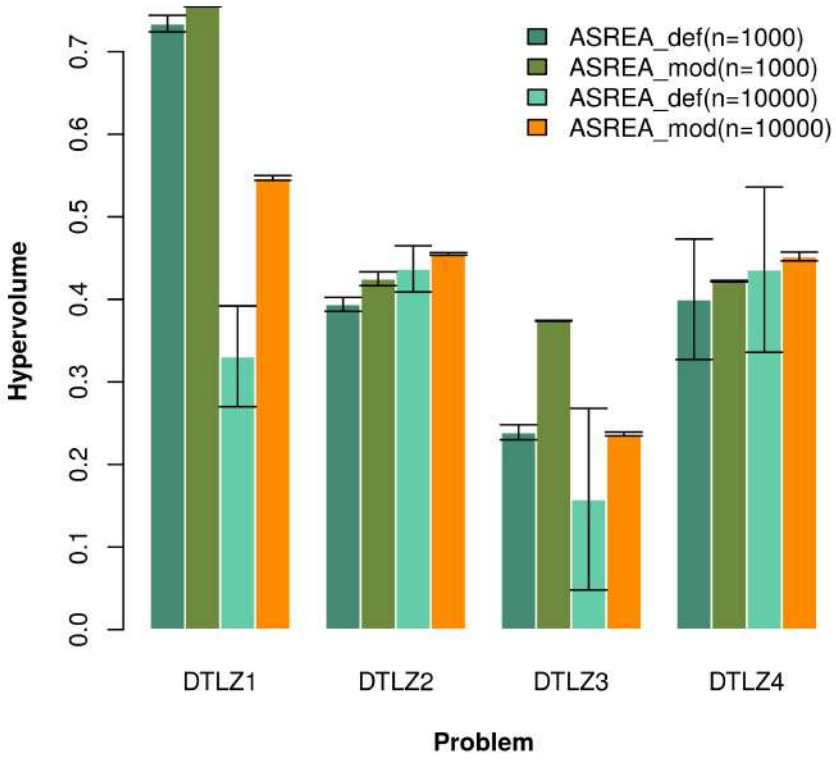


Figure 4.6: Average value of the hypervolume indicator $\overline{I}_{HV}$ with standard deviation σ obtained by ASREA with the modified archive size (ASREA$_{mod}$) and with the default archive size (ASREA$_{def}$) on DTLZ w.r.t. the different population sizes: $n = 1000$ and $n = 10000$.

for large ($n = 1000$) and very large ($n = 10000$) population size, the accuracy of $\overline{I}_{HV}$ values is improved in the results obtained by ASREA with modified archive size.

To make things clear, we present these results in Figure 4.6, where the mean values of the hypervolume $\overline{I}_{HV}$ and the standard deviation obtained by ASREA with and without modified archive size are compared.

From Figure 4.6 we can make the following conclusions about the results obtained on all test problems by ASREA with a modified archive size:

- the standard deviation, especially for very large population size, is reduced;
- the performance is improved, since: i.e., the value of $\overline{I}_{HV}$ is increased;
- the population scalability is improved. Event though, under the same computational budget, on multi-modal problems (DTLZ1, DTLZ3) the value of $\overline{I}_{HV}$ for $n = 10000$ is still less than for $n = 1000$, its accuracy is higher comparatively with the results of the original ASREA.

The presented results in Figure 4.6 and Table 4.6 confirm the positive impact of the proposed modifications of archive size on the performance of the algorithm, when it handles large/very large population size;

4.3.1.2 Validation of Mutation Operator

In this section, we perform the validation of Cauchy-based Mutation Operator with non-static probability proposed in Section 4.2.5.

For this purpose, two algorithms, FastEMO and a peer state-of-the-art algorithm NSGA-II, are employed to investigate the effect of the proposed mutation operator against polynomial mutation on the performance for both algorithms.

Experimental Objective: to validate the efficiency of the proposed mutation operator, working in two different MOEAs: NSGA-II and FastEMO.

Test Problems: the first four 3-objective problems ($m = 3$) from DTLZ and WFG test suites are employed. For more details, one can see Section 3.5.2.

Performance Metrics: $\overline{I}_{HV}$ is the mean value of the hypervolume indicator and standard deviation over 30 independent runs of each algorithm for each test case.

Simulation Settings:

- **computational budget:** it has fixed number of evaluation functions for all test cases, i.e., $EF = 500000$;
- **population size:** $n = 1000$;
- **generation number:** $t = 500$;
- **archive size:** $a = 45$
- **number of objectives:** $m = 3$;

- **number of decision variables:** $d = 7$ for DTLZ1, $d = 12$ for DTLZ2-DTLZ4 and $d = 24$ for WFG1-4. The defined values are calculated for each test problem according to the rules and suggested values described in Section 3.5.2 and originally presented in [Deb et al. (2002)] and [Huband et al. (2005)] for DTLZ and WFG respectively.

- **statistics:** each algorithm is run over 30 independent runs for each test case.

We choose a population size of 1000 and run each algorithm for 500 generations to capture as many Pareto-optimal solutions as possible by both algorithms. Moreover this test case provides intermediate conditions for both algorithms, which makes this experiment unbiased.

Experimental Results: The performance in terms of $\overline{I}_{HV}$ of both algorithms with Cauchy-based and polynomial mutation operators is shown in Figure 4.7 and Table 4.7, where: (i) the highlighted light and dark grey colours represent the best obtained the mean value of the hypervolume by FastEMO and NSGA-II respectively; (ii) the values of the standard deviation are presented in brackets with the best values of each algorithm on bold. Thus, we can estimate the impact of the mutation operator on both algorithms.

Table 4.7: Average value of the hypervolume indicator $\overline{I}_{HV}$ and standard deviation (in brackets) obtained by FastEMO and NSGA-II with polynomial and Cauchy-based mutation operators on DTLZ and WFG problems: $m = 3, n = 1000, a = 45, t = 500$.

Problem	FastEMO Poly	FastEMO Cauchy	NSGA-II Poly	NSGA-II Cauchy
DTLZ1	$75.5e-02_{(1.0e-02)}$	$79.5e-02_{(1.0e-02)}$	$81.1e-02_{(\mathbf{9.0e-03})}$	$81.2e-02_{(1.1e-02)}$
DTLZ2	$42.5e-02_{(8.4e-03)}$	$42.5e-02_{(\mathbf{8.0e-03})}$	$44.2e-02_{(\mathbf{7.9e-03})}$	$44.7e-02_{(8.0e-03)}$
DTLZ3	$37.4e-02_{(\mathbf{9.1e-03})}$	$41.0e-02_{(9.2e-03)}$	$42.8e-02_{(6.9e-03)}$	$4.3e-02_{(\mathbf{6.5e-03})}$
DTLZ4	$42.2e-02_{(\mathbf{9.2e-03})}$	$42.2e-02_{(9.3e-03)}$	$44.1e-02_{(1.9e-02)}$	$44.3e-02_{(\mathbf{8.9e-03})}$
WFG1	$31.4e-02_{(1.3e-02)}$	$56.7e-02_{(\mathbf{1.0e-02})}$	$38.4e-02_{(\mathbf{7.0e-03})}$	$47.4e-02_{(8.1e-03)}$
WFG2	$89.9e-02_{(4.3e-03)}$	$92.6e-02_{(\mathbf{3.3e-03})}$	$92.5e-02_{(\mathbf{2.0e-03})}$	$92.6e-02_{(3.3e-03)}$
WFG3	$29.7e-02_{(2.6e-03)}$	$31.8e-02_{(\mathbf{1.6e-03})}$	$32.6e-02_{(\mathbf{1.0e-03})}$	$32.9e-02_{(1.2e-03)}$
WFG4	$32.0e-02_{(\mathbf{1.7e-03})}$	$43.1e-02_{(1.8e-03)}$	$42.1e-02_{(\mathbf{1.1e-03})}$	$42.4e-02_{(2.0e-03)}$

Mutation in the design of NSGA-II:

As seen from Figure 4.7 and Table 4.7, in the design of NSGA-II, both mutation operators perform almost equally well for all 8 problems. However, we observe that Cauchy-based mutation operator provides slightly better (large) value of $\overline{I}_{HV}$ on all 8 problems. On average, in the design of NSGA-II, the scale of improvement and weakening of $\overline{I}_{HV}$ value provided by Cauchy-based mutation is 1e-03.

On separable functions, i.e., DTLZ1-4, WFG1 and WFG4, the most significant improvement is obtained on the WFG1 problem, which is uni-model biased and hasa complex

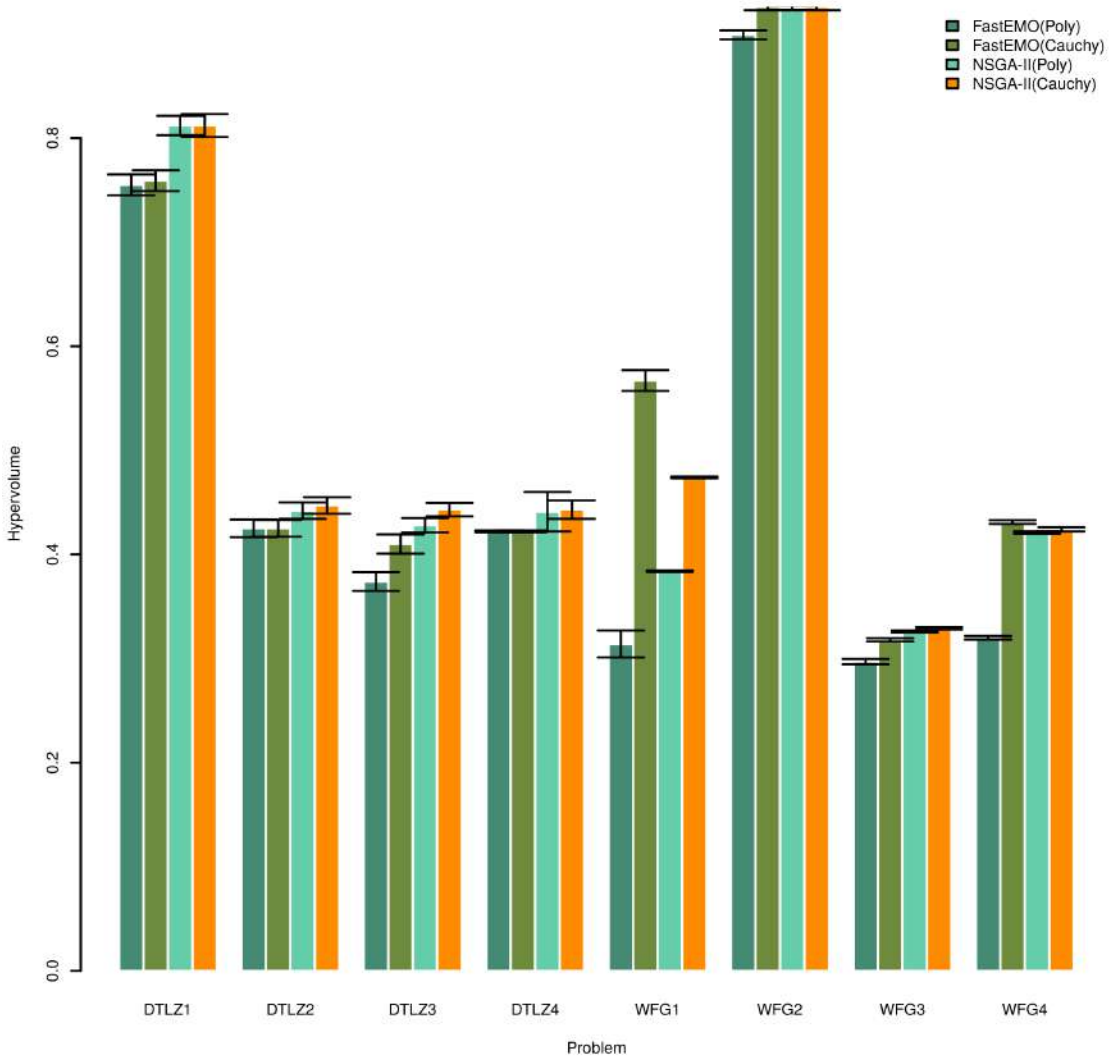


Figure 4.7: Average value of the hypervolume indicator $\overline{I}_{HV}$ and the standard deviation obtained by FastEMO and NSGA-II with the polynomial and the Cauchy-based mutation operators on DTLZ and WFG problems: $m = 3, n = 1000, a = 45, t = 500$.

Pareto front. This result can be explained by the efficiency of the Cauchy distribution on separable problems [Hansen et al. (2006)].

On non-separable problems, i.e., WFG2, WFG3, the Cauchy distribution based mutation operator performs similarly to a polynomial operator in the design of NSGA-II, which is quite predictable [Hansen et al. (2006)].

The differences in the values of standard deviation are negligible.

Mutation in the design of FastEMO:

In this test, under FastEMO, we consider ASREA with the modified archive size. In the design of FastEMO, the Cauchy-based mutation operator shows a positive impact on the performance of the modified archive-based algorithm, where it outperforms the polynomial operator on 6 out of 8 problems: DTLZ1, DTLZ3, WFG1-WFG4.

On separable multi-modal functions, we observe an increase of the $\overline{I}_{HV}$ value of the scale of 1e-02, 1e-01 and 1e-01 on DTLZ1, DTLZ3 and WFG4 respectively. This observation supports the idea that the so-called heavy tailed mutation operator, based on the Cauchy distribution, can help to improve the diversity of the population.

On separable uni-modal functions, DTLZ2 and DTLZ4, the impact of the Cauchy-based mutation is less significant. However, as in the design of NSGA-II, we can see an improvement of the value of $\overline{I}_{HV}$ on the uni-modal complex WFG1.

On non-separable problems, i.e., WFG2 and WFG3, we also observe an improvement of the $\overline{I}_{HV}$ value of scale 1e-01.

Comparison of Pareto fronts:

In order to complete this comparative study, in Figure 4.8 we demonstrate the Pareto fronts obtained by FastEMO and NSGA-II with both mutation operators on separable uni-modal biased WFG1 test problem. A shape of the optimal Pareto front of WFG1 problem is shown in Figure 4.8 (a).

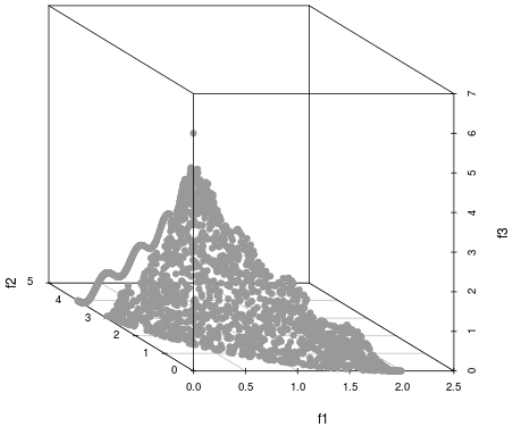
Figure 4.8 (b) presents the Pareto front obtained by FastEMO with Cauchy mutation operator with a very large population size $n = 10000$. As we can see, this front locates in the correct area of the objective (target) space and its shape is close to the optimal front.

Figure 4.8 (c) and (d) show the Pareto front obtained by FastEMO with polynomial mutation and Cauchy-based mutation respectively. While FastEMO with polynomial mutation (Figure 4.8 (c)) fails to maintain an adequate distribution of solutions, FastEMO with Cauchy mutation (Figure 4.8 (d)) successfully achieves a well distributed set of solutions covering the entire front.

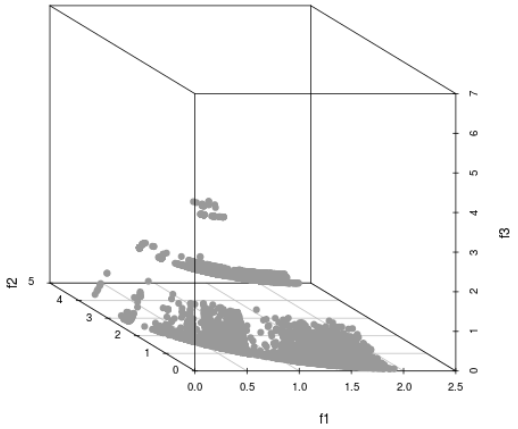
Figure 4.8 (e) and (f) show the Pareto front obtained by NSGA-II with polynomial mutation and Cauchy-based mutation respectively. Even though NSGA-II achieves to maintain a lot of solutions in the front and an adequate distribution of solutions, in both cases, we observe a lack of precision: the location of the obtained front is shifted on axis $f1$ and $f3$. However, we can observe a little improvement of the front location, when Cauchy mutation is applied.

Summary:

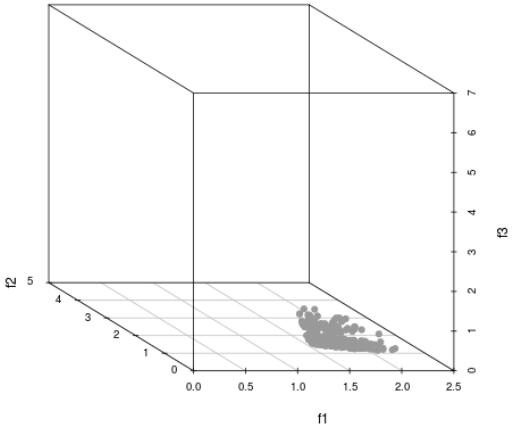
To summarize, from Table 4.7 we can conclude that the best results of both algorithms are obtained mainly with a Cauchy-based mutation operator. This can partly be explained by the efficiency of the Cauchy-based mutation on separable functions (6 out of 8 applied



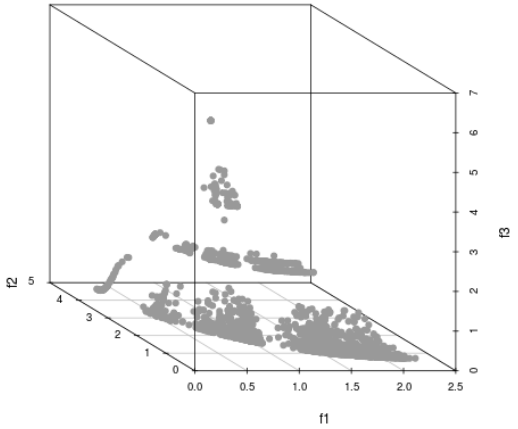
a) Optimal Pareto front



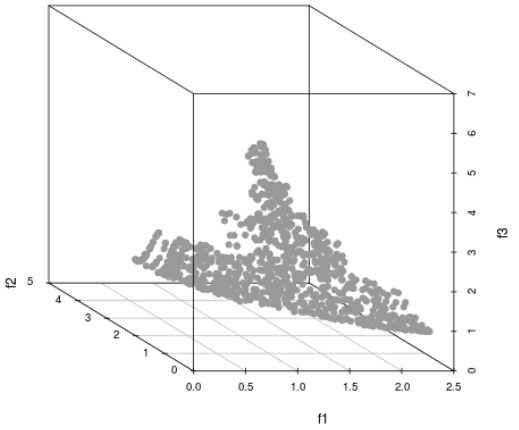
b) FastEMO with Cauchy mut. $n = 10000$



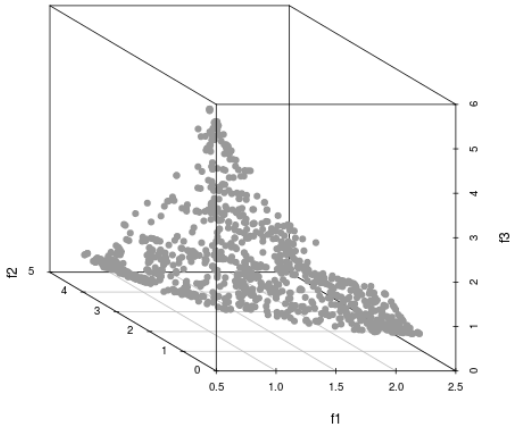
c) FastEMO with poly.mut $n = 1000$



d) FastEMO with Cauchy mut. $n = 1000$



e) NSGA-II with poly. mut.$n = 1000$



f) NSGA-II with Cauchy mut. $n = 1000$

Figure 4.8: Pareto front obtained by NSGA-II and FastEMO with polynomial and Cauchy mutation under a total computational budget of $EF = 500000$.

test problems are separable). This result affirms the effectiveness of the proposed operator in the design of FastEMO for solving the separable or partially separable optimization problem of the Hamiltonian model. However, from Table 4.7, one should mention that the mean value of $\overline{I}_{HV}$ of FastEMO is slightly smaller (worse) than that of NSGA-II on 6 problems out of 8. This fact is explained by the population size $n = 1000$, which is not large enough for the efficient performance of FastEMO. For comparison, one can see the results of FastEMO on DTLZ1-DTLZ4 with $n = 10000$ presented in Table 4.8.

The obtained results confirm the applicability of the proposed Cauchy-based mutation operator with non-static probability in the design of FastEMO, as well as in the design of different MOEAs, and its competitiveness in comparison with classical polynomial mutation operator.

4.3.1.3 Benchmarking FastEMO *vs* ASREA on DTLZ Problems with Large Population Size

Experimental Objective: to validate the design of FastEMO by comparing its performance against that of its predecessor - ASREA.

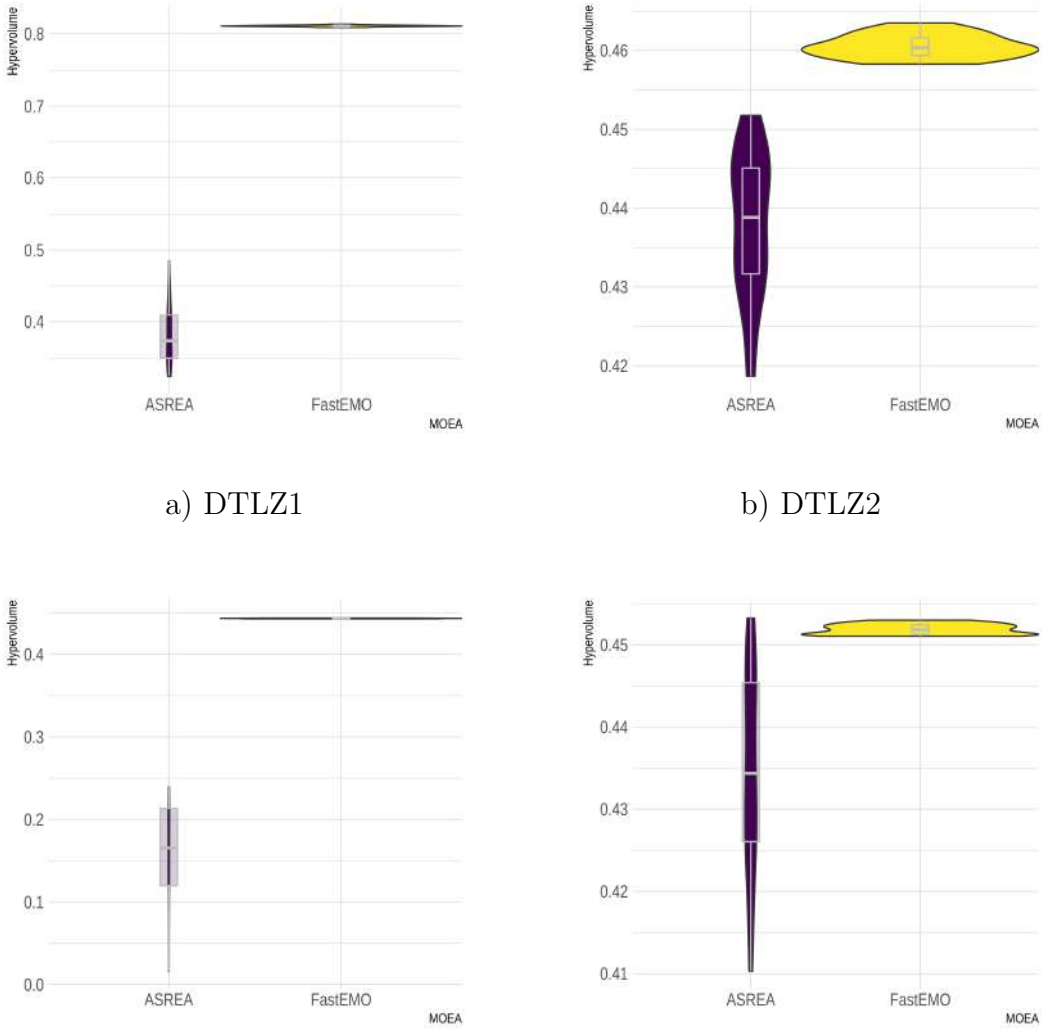
Test Problems: the first four 3-objective problems ($m = 3$) from DTLZ test suites are employed. For more details, see Section 3.5.2.

Performance Metrics: best, worst, mean value of hypervolume indicator (I_{HV}), its standard deviation (σ) and mean value of runtime in seconds over 30 independent runs of each algorithm for each test case.

Simulation Settings:

- **computational budget:** it has a fixed number of evaluation functions for all test cases, i.e., $EF = 500000$;
- **population size:** $n = 10000$;
- **number of generations:** $t = 50$;
- **archive size:** $a = 20 \cdot m$
- **number of objectives:** $m = 3$;
- **number of decision variables:** $d = 7$ for DTLZ1, $d = 12$ for DTLZ2-DTLZ4. The defined values are calculated for each test problem according to the rules and suggested values described in Section 3.5.2 and originally presented in [Deb et al. (2002)].
- **statistics:** each algorithm is run over 30 independent runs for each test case.

We choose a population size 10000 and run each algorithm for 50 generations to check the most interesting use-case for parallel implementation, which allows both algorithms to capture as many Pareto-optimal solutions as possible.



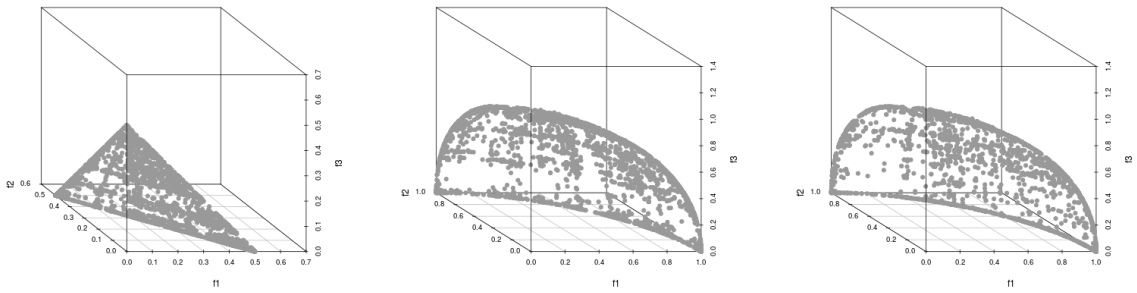
a) DTLZ1 b) DTLZ2

c) DTLZ3 d) DTLZ4

Figure 4.9: Average value of the hypervolume indicator $\overline{I}_{HV}$ and standard deviation obtained by benchmarked algorithms on 3-objective problems of the DTLZ test suite w.r.t. the population size $n = 10000$ and the number of generations $t = 50$ (computational budget $EF = 500000$).

Table 4.8: Performance of FastEMO vs ASREA on DTLZ1-DTLZ4 problems: $d = 7$(DTLZ1)/$d = 12$ (DTLZ2-4), $m = 3, n = 10000, t = 50$.

Metric	Algorithm	DTLZ1	DTLZ2	DTLZ3	DTLZ4
Best I_{HV}	FastEMO	81.1e-2	46.4e-2	45.2e-2	45.5e-2
	ASREA	48.5e-2	45.2e-2	24.0e-2	45.5e-2
Worst I_{HV}	FastEMO	81.0e-2	44.1e-2	44.3e-2	43.3e-2
	ASREA	32.0e-2	41.7e-2	5.0e-3	45.1e-2
Mean $\overline{I}_{HV}$	FastEMO	81.1e-2	46.1e-2	44.4e-2	45.3e-2
	ASREA	33.1e-2	43.7e-2	15.8e-2	43.6e-2
σ	FastEMO	3.0e-3	1.6e-2	2.0e-3	5.1e-3
	ASREA	6.0e-2	2.8e-2	1.1e-1	1.0e-1
Mean runtime $\overline{T}$(s)	FastEMO	5.4	9.7	8.8	12.2
	ASREA	3.0	3.8	3.8	4.2



DTLZ1 ($d = 7$) DTLZ2 ($d = 12$) DTLZ4 ($d = 12$)

Figure 4.10: Pareto Fronts ($m = 3, n = 10000, a = 60, t = 50$) obtained by FastEMO on DTLZ test suite.

Experimental Results:

The experimental results are summarized in Table 4.8 and depicted in Figure 4.9.

Table 4.8 shows the comparative results of the investigated algorithms, i.e., ASREA and FastEMO, on the DTLZ problem suite regarding the best, worst, mean, standard deviation of the values of hypervolume indicator and mean runtime, where the gray background represents the best results. On the whole, FastEMO is the better performing algorithm, comparatively with ASREA by the accuracy of the best, worst and mean values of hypervolume indicator. Moreover, FastEMO reduces the value of standard deviation and provides more stable results. From Figure 4.9 we can state that FastEMO achieves better accuracy of the average hypervolume indicator on separable multi-modal functions (Figure 4.9 (a,c)) and provides less variance, while handling a very large population size ($n = 10000$).

We visualized some three-objective problems solved with FastEMO in Figure 4.10: i.e., DTLZ1, DTLZ2 and DTLZ4 problems. According to obtained Pareto fronts from Figure 4.10, we can see that FastEMO achieves well-distributed and well-spread solutions

for those separable problems.

Thus, we can conclude that the new design overcomes the main drawbacks of the baseline algorithm, ASREA: (i) the value of the standard deviation of the hypervolume indicator is reduced; (ii) the accuracy of the value of the hypervolume indicator is improved on separable uni- and multi-modal problems; (iii) FastEMO successfully handles a very large population size ($n = 10000$) with small increase of overall mean runtime comparatively with that of ASREA on DTLZ1-4 3-objective test problems (in average only in 2.4 times slower than ASREA). This increase of average runtime is explained by the modification of the archive size and the computational complexity of the converting operator and the alternative crowding distance operator.

4.3.2 Validation of FastEMO Properties

4.3.2.1 The Population Scalability

Let us start the validation of the FastEMO properties by confirming its applicability for a parallel implementation. Running the objective functions computations in parallel is a source of speedup, which raises the question of the scalability of FastEMO w.r.t. the population size.

Test Problems: the DTLZ benchmark suite. For more details, one can see Section 3.5.2.

Performance Metrics: $\overline{I}_{HV}$ is the mean value of hypervolume indicator and average runtime in seconds.

Simulation Settings: Like in the previous test, we use the overall computation budget of 500000 function evaluations. However the proportion between the population size n and the number of generations t is changed during the experiments, because the scalability w.r.t. the population size has to be checked. The number of objectives m is fixed at 3.

Experimental Results: The performance of FastEMO in terms of $\overline{I}_{HV}$ and average runtime w.r.t. the population size is shown in Table 4.9, where the highlighted grey colour represents the best results.

As can be seen from Table 4.9, the best result for all problems is obtained with the largest value of $n = 10000$ and in only 50 generations. This result is predictable, because the large population size helps to maintain diversity and is useful for solving multi-modal functions: i.e., DTLZ1 and DTLZ3. We assume that the performance with the smaller population size can be improved, by calibrating the hyper-parameters of FastEMO defined in Table 4.5.

However, the obtained results confirm the applicability of FastEMO for efficiently solving computationally intensive problems in parallel, i.e., the optimization problems of the model of Active Magnetic Regenerator and the Hamiltonian model for studying the physical properties of Magneto Caloric Materials.

For the sake of completeness, we investigate the average execution time of FastEMO per generation w.r.t. the population size on the very fast DTLZ2 test function (only

Table 4.9: Average $\overline{I}_{HV}$ and mean runtime of FastEMO on DTLZ test suite w.r.t. population size n.

Problem	Metric	$n = 100$ $t = 5000$	$n = 1000$ $t = 500$	$n = 10000$ $t = 50$
DTLZ1	I_{HV}	74.9e-02	78.2e-02	81.1e-02
	T_{avg}	5.1s	5.6s	5.4s
DTLZ2	I_{HV}	37.9e-02	43.0e-02	46.1e-02
	T_{avg}	9.1s	8.8s	9.7s
DTLZ3	I_{HV}	35.3e-02	41e-02	44.4e-02
	T_{avg}	7.5s	8.2s	8.8s
DTLZ4	I_{HV}	38.0e-02	42.2e-02	45.1e-02
	T_{avg}	11.2s	12.7s	12.2s
DTLZ7	I_{HV}	28.5e-02	30.7e-02	32.4e-02
	T_{avg}	5.2s	7.1s	7.7s

Table 4.10: Average runtime per generation (in sec.) on DTLZ2 w.r.t. population size.

Algorithm	$n = 100$	$n = 1000$	$n = 10000$	$n = 100000$
FastEMO	0.00086	0.007	0.071	1.921
MOEA-D	0.00089	0.020	0.904	3.512
NSGA3	0.00157	0.034	4.584	65.764
IBEA	0.03442	0.230	65.764	3150
CDAS	0.00095	0.031	3.489	64.166

4.10^{-3}s evaluation time). The obtained result is compared against the results of NSGA-III, MOEA-D, CDAS and IBEA. We are taking into account that some algorithms, like CDAS and NSGA-III, do not have the same complexity for every generation. That is why the average execution time per generation is considered for evaluation.

All mentioned state-of the-art algorithms use the parameter settings of NSGA-II defined in Table 4.5.

The comparative results are presented in Table 4.10 and Figure 4.11. This study shows that using a too large populations (i.e., a population size $>$ 10000 solutions) makes NSGA-III, CDAS and IBEA slow and inefficient. Whereas MOEA-D and FastEMO work efficiently even when it is executed with a very large population. As seen from Table 4.10, FastEMO is slightly faster than MOEA-D on all of the population sizes. This result experimentally demonstrates the effectiveness of the low computational complexity of FastEMO, when the population size is supposed to be very large and the objective function computations will run in parallel.

This feature makes FastEMO very useful for solving computationally intensive problems of the simulation models, i.e., the AMR model and the Hamiltonian model.

4.3.2.2 Scalability w.r.t. Objective Space Dimension

As the optimization problem of the AMR model can be formulated as a multi- and many-objective problem, scalability w.r.t. dimension of the objective (target) space has to be investigated.

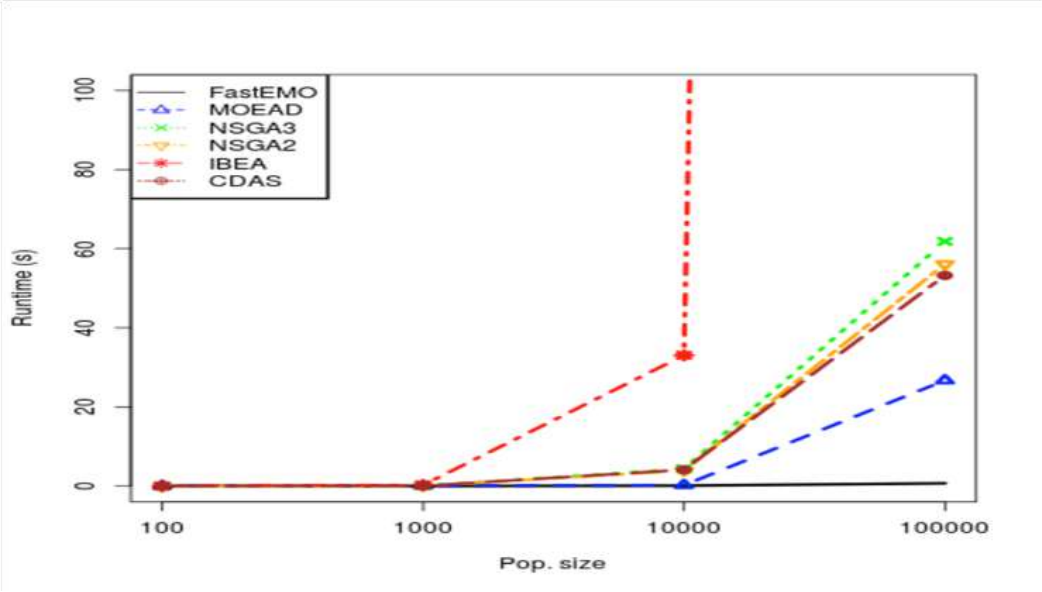


Figure 4.11: Average runtime per generation w.r.t. population size obtained by 5 algorithms on DTLZ2 test problem.

Test Problems: WFG benchmark suite. For more details, one can see Section 3.5.2.

Performance Metrics: $\overline{I}_{HV}$ is mean value of the hypervolume indicator.

Simulation Settings: The population size n is set to 1000 and the number of generations t is fixed to 500. For investigating the scalability of FastEMO w.r.t. the dimension of the objective (target) space, we limit our experiment to $m \in \{3, 4, 5\}$, because these dimensions are the most interesting from the point of view of the optimization problems of the AMR and because the bi-objective tests are separately conducted in this experimental study.

Like before, the number of decision variables is calculated for each test problem according to the rules defined in [Huband et al. (2005)] (more details are presented in Section 3.5.2). The performance of each algorithm is evaluated over 30 independent runs per problem.

Experimental Results: The performance in terms of I_{HV} of both algorithms, FastEMO with the Cauchy-based mutation operator and NSGA-II with Polynomial mutation operators is shown in Table 4.11, where the highlighted light, normal and dark grey colours represent the best results obtained by both algorithms for dimensions $m = 3$, $m = 4$ and $m = 5$ respectively.

From Table 4.11 we see that FastEMO outperforms NSGA-II on 5 problems out of 9 for $m = 3$, $m = 4$ and $m = 5$. The most significant performance of FastEMO in contrast to NSGA-II is obtained on problems WFG1, WFG6 and WFG7 for all dimensions, where WFG1 and WFG7 are separable problems. These results are quite predictable due to the property of separability of Cauchy-based mutation. On the discontinuous multi-modal problem, WFG2, FastEMO shows a more stable performance than NSGA-II with an increasing number of objectives. However, we observe a degradation of the

Table 4.11: Performance ($\overline{I}_{HV}$) of FastEMO and NSGA-II on WFG problems: $m \in \{3, 4, 5\}, n = 1000, t = 500$.

Problem	$m = 3$		$m = 4$		m = 5	
	FastEMO	NSGA-II	FastEMO	NSGA-II	FastEMO	NSGA-II
WFG1	57.4e-02	38.4e-02	22.9e-02	14.3e-02	22.9e-02	14.9e-02
WFG2	92.6e-02	92.6e-02	96.0e-02	94.8e-02	90.2e-02	87.3e-02
WFG3	31.8e-02	32.6e-02	21.8e-02	22.3e-02	11.2e-02	8.0e-02
WFG4	43.1e-02	42.1e-02	44.4e-02	51.5e-02	43.9e-02	51.2e-02
WFG5	39.5e-02	39.5e-02	48.9e-02	49.1e-02	52.7e-02	53.2e-02
WFG6	40.6e-02	39.4e-02	53.7e-02	49.2e-02	53.9e-02	46.4e-02
WFG7	43.5e-02	42.6e-02	57.2e-02	54.2e-02	53.4e-02	47.9e-02
WFG8	37.8e-02	37.3e-02	71.1e-02	73.8e-02	79.9e-02	83.4e-02
WFG9	34.6e-02	34.6e-02	48.5e-02	48.2e-02	49.6e-02	50.1e-02

Table 4.12: Average runtime (s) of FastEMO and NSGA-II on all WFG problems: $m \in \{3, 4, 5\}, n = 1000, a = 45, t = 500$.

$m = 3$		$m = 4$		m = 5	
FastEMO	NSGA-II	FastEMO	NSGA-II	FastEMO	NSGA-II
7	24	10	32	15	38

$\overline{I}_{HV}$ when increasing the number of objectives, with FastEMO on multi-modal separable function WFG4. This bad result is quite surprising, due to the strong positive impact of the Cauchy-based mutation on separable functions. We suppose that this issue can be improved by increasing the population size or by a more careful selection of the control parameter of contraction/expansion of the dominance S. In order to check the first hypothesis, we increased the population size to 10000 and decreased the number of generations to 50, in order to keep the same number of function evaluations. After 30 independent runs, the mean value of $\overline{I}_{HV}$ is in about 1.2 times better than with $n = 1000$ and $t = 500$.

Going back to the second hypothesis, one should note that the role of the Controlling Dominance Area of Solutions (CDAS) technique in the performance of FastEMO is increasing with growing the number of objectives. We show this on several examples. Let us take two following problems: WFG2 (the discontinuous multi-modal problem) and WFG7 (the separable biased function). If we set S to 0.5, which will be equivalent to the conventional Pareto dominance, the results of FastEMO on these functions is about in 1.2 times worse for $m = 4$ and about in 1.4 times for $m = 5$. It confirms that the choice of a value for the control parameter of contraction/expansion of the dominance S is very delicate issue. We experimentally found the most appropriate values in the general case and defined them in Table 4.5. However, more sophisticated approach to handle parameter S during the optimization process is needed.

The average execution time is presented in Table 4.12, where the highlighted light,

normal and dark grey colours represent the best time achieved by the algorithms for dimensions $m = 3$, $m = 4$ and $m = 5$ respectively.

One should mention that in general, the I_{HV} of FastEMO and NSGA-II presented in Table 4.11, are not significantly different. However, from Table 4.12 we can see that FastEMO is significantly faster: on average, FastEMO in 3.4 times faster than NSGA-II for $m = 3$, in 3.2 times faster for $m = 4$ and in 2.5 times for $m = 5$.

4.3.2.3 The Scalability w.r.t. the Search Space Dimensions

For a comparative study, we use the reference results of three different state-of-the-art algorithms: Strength Pareto Evolutionary Algorithm II (SPEA-II) [Zitzler et al. (2001)], Non-dominated Sorting Genetic Algorithm II (NSGA-II) [Deb et al. (2000)] and Multi-Objective Covariance Matrix Adaptation Evolution Strategy (MO-CMA-ES) [Igel et al. (2007)]. We again employ NSGA-II, because it is the main reference algorithm in this work. Then, we select SPEA-II in order to compare the performance of FastEMO with that of conventional archive-based method. The third reference algorithm in this study, MO-CMA-ES, is chosen as one of the robust methods w.r.t. different properties of optimization functions, as the validation of this robustness is one of two aims of this experiment.

The reference results of NSGA-II, MO-CMA-ES and SPEA-II are provided using the COCO platform [2] [Hansen et al. (2021)].

Test Problems: 54 problems of the bi-objective BBOB test suite [Hansen et al. (2012)], where each problem has 15 different instances. One can find more details about the BBOB test suite in Section 3.5.2.

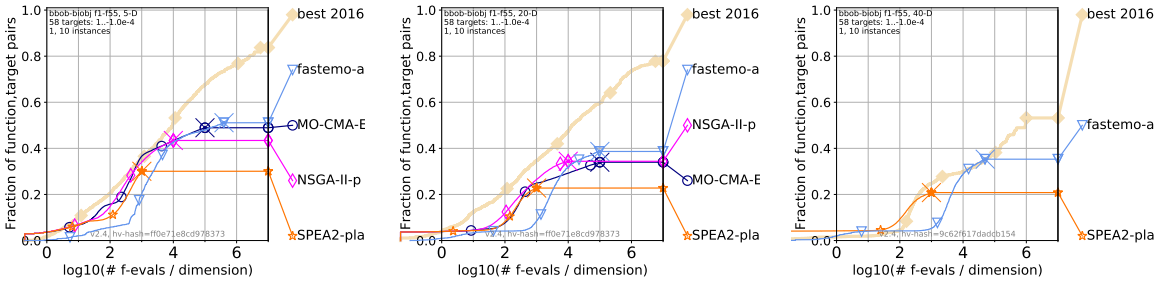
Performance Metrics: Empirical Cumulative Distribution Functions (ECDFs) described in Section 3.5.1.

Simulation Settings: The population size n is set to 1000 and the number of generations t to 500. The archive size a is fixed at $15 \cdot m$, where $m = 2$. The number of decision variables, the dimension of search space is defined as follows: $d \in \{5, 20, 40\}$.

Experimental Results: Figure 4.12 shows the empirical cumulative distribution of runtimes of FastEMO, NSGA-II, SPEA-II and MO-CMA-ES on all 54 functions in dimensions $d = 5$, $d = 20$ and $d = 40$. We observe that FastEMO solved 51%, 39% and 35.5% for 5, 20 and 40 decision variables respectively. Unfortunately, the reference results for 40 decision variables of NSGA-II and MO-CMA-ES are not provided by the COCO platform.

As seen from Figure 4.12 a), for dimension $d = 5$, FastEMO solved the largest number of problems, followed by MO-CMA-ES (%49 of solved problems) and NSGA-II (%42 of solved problems). However, the speedup of FastEMO is less pronounced, especially in the first stage, where MO-CMA-ES and NSGA-II are about in 2 times faster. It is not surprising, because we use a large population size $n = 1000$, and consequently FastEMO can not demonstrate the reduced number of function evaluations. However, in our case

[2] `https://coco.gforge.inria.fr/`



a) $d = 5$ b) $d = 20$ c) $d = 40$

Figure 4.12: Empirical cumulative distribution of runtimes, summarized by all bi-objective function groups on $d = 5$, $d = 20$ and $d = 40$.

it is not a problem, because if the objective functions are computed in parallel, the large population size leads to an overall better performance in terms of the number of solved problems. The same scenario can be observed in Figure 4.12 b) and c) for $d = 20$ and $d = 40$ respectively.

In comparison to the archive-based algorithm SPEA-II, FastEMO outperforms it on 12 over 15 classes of bi-objective BBOB test suite for 20-dimensional search space, and on all 15 classes for 40-dimensional search space.

The results presented in Figure 4.12, confirm the scalability of FastEMO w.r.t. the dimensions of the search space: when increasing the number of decision variables, the overall number of solved functions reduces insignificantly, especially for dimensions from $d = 20$ to $d = 40$. This feature of FastEMO is especially important for solving the optimization problem of the AMR model, where the search space can be scaled approximately from 5 to 100.

4.3.2.4 Robustness w.r.t. Different Properties of Optimization Problems

For the sake of completeness, we benchmark FastEMO to investigate its robustness to solve different problems: i.e., separable, uni-modal, multi-modal, etc. For a comparative study, we use the reference results of the same algorithms, which were employed in the previous experiment.

Test Problems: 54 problems of bi-objective BBOB test suite [Hansen et al. (2012)], where each problem has 15 different instances. One can find more details in Section 3.5.2.

Performance Metrics: Empirical Cumulative Distribution Functions (ECDFs) described in 3.5.1.

Simulation Settings: The population size n is set to 1000 and the number of generations t to 500. The dimension of the search space d is set to 15. The maximum size of archive a is fixed at $15 \cdot m$, where $m = 2$.

Experimental Results: Figure 4.13 shows the empirical cumulative distribution of the runtimes of FastEMO, NSGA-II, SPEA-II and MO-CMA-ES in a 20-dimensional search space for 54 functions aggregated in 15 different groups.

FastEMO performs better for 10 groups out of 15 (see Figure 4.13 (a, b, c, d, e, f, g, i, j, l)). It means that FastEMO solves about 36 problems out of 54 with a better accuracy than reference algorithms. The worst results of FastEMO are shown on multi-modal problems, which is explained by the relatively small population size $n = 1000$ for FastEMO: a larger population size is required to solve multi-modal functions.

We observe that FastEMO performs efficiently on separable problems (Figure 4.13 (a, b, c)), solving about 52% of separable-separable problems, 60% of separable-moderate problems and about 52% of separable-ill-conditioned problems. This results confirm the efficiency of FastEMO to solve separable problems, which is important in the context of the optimization problem of the Hamiltonian model.

As in the previous experiments, FastEMO demonstrates a slow runtime at the first stage of the optimization process for all groups of functions.

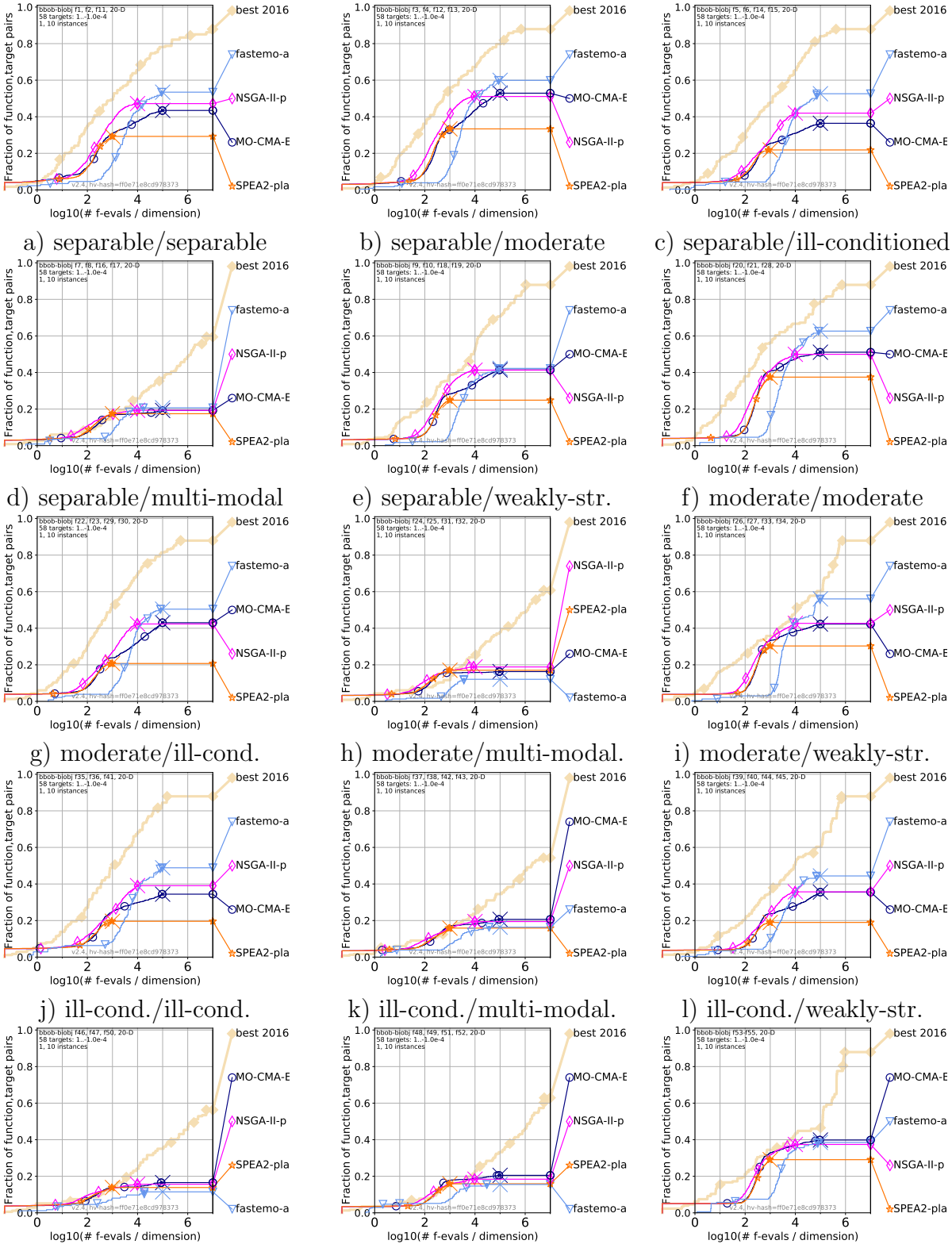
On multi-modal and some ill-conditional functions, the runtime at the first stage is better, but it is still quite slow in the middle stage (see Figure 4.13 (j, k, m, n, o)). These results are explained by the relatively large population size, which is useful for solving multi-modal problems even in the beginning of the optimization process. However, FastEMO requires larger population size to solve multi-modal problems with better precision.

To summarize, on bi-objective problems in a 20-dimensional search space, FastEMO demonstrates robustness w.t.t. different properties of optimization problems: it performs better than the reference algorithms on 36 problems out of 54 (10 groups out of 15), where it shows especially good results in contrast to the reference algorithms on separable and moderate problems by solving about 50%-60% of problems par group. This ability to solve separable problems confirms the applicability of FastEMO to solve the optimization problem of the Hamiltonian model.

4.4 Summary and Discussion

In this chapter, we introduced an improved version of the archive-based evolutionary optimization algorithm, Archived-based Stochastic Ranking Evolutionary Algorithm (ASREA) [Sharma & Collet (2010*a*)], for solving in parallel time-consuming continuous multi- and many-objective problems of the Active Magnetic Regenerator (AMR) model. More specifically, the presented version, called FastEMO, has been developed in order to ensure the following properties:

1. the low worst case computational complexity;
2. the population scalability;
3. the scalability w.r.t. the dimensions of the objective (target) space: i.e., from multi- to many-objective;
4. the efficiency when one point in the objective space can correspond to more than one non-dominated solutions in the search space, in order to not lose some interesting solutions;
5. the scalability w.r.t. the dimensions of the search space, in order to handle the problems with different number of decision variables;



a) separable/separable b) separable/moderate c) separable/ill-conditioned

d) separable/multi-modal e) separable/weakly-str. f) moderate/moderate

g) moderate/ill-cond. h) moderate/multi-modal. i) moderate/weakly-str.

j) ill-cond./ill-cond. k) ill-cond./multi-modal. l) ill-cond./weakly-str.

m) multi-modal /multi-modal. n) multi-modal/weakly-str. o) weakly-str./weakly-str.

Figure 4.13: Empirical cumulative distribution of runtimes, summarized by all bi-objective function groups on 20D.

6. the resistance against Dominance Resistant Solutions (DRS): i.e., solutions, which have extremely good values for some objectives and extremely bad values for other objectives.

ASREA [Sharma & Collet (2010*a*)] is selected as a baseline algorithm, because it ensures a low computational complexity of the worst case in single generation: $\mathcal{O}(man)$, where m is the number of objectives, a is the archive size and n is the population size. However, ASREA has the following disadvantages, which have to be improved:

1. ASREA is not efficient on many-objective problems and is not resistant to Dominance Resistant Solutions (DRS), like all algorithms based on Pareto dominance;

2. ASREA demonstrates a high variance of the value of the hypervolume indicator when increasing the population size;

3. ASREA shows a degradation of the value of the hypervolume indicator when increasing the population size on multi-modal functions.

4. ASREA can lose some important solutions, when several non-dominated solutions are close to each other in the objective space, but different in the search space.

FastEMO inherits the small-sized archive of non-dominated solutions from ASREA and improves the disadvantages of ASREA by integrating the contributions presented below.

In principle, FastEMO can be employed for solving many other real world multi- and many-objective problems, e.g., the problems of the Hamiltonian model of magneto caloric materials.

Contribution: The contributions, in regards to solve the above mentioned issues, are the following modifications, which are integrated into the structure of ASREA:

1. The replacement of the Pareto dominance operator by the operator based on the Control the Dominance Area of Solutions (CDAS) [Sato et al. (2007)] technique, which helps to provide the scalability w.r.t. the objective space and theoretically, to ensure resistance to DRS, but we did not confirm it by benchmarking in this study.

2. The increasing of the archive size to the population size in the last generation that allows the algorithm to obtain a larger number of non-dominated solutions in the final Pareto front, which is required for better exploration of the design of the AMR model. This simple technique increases the total computational complexity, but not significantly.

3. The replacement of the polynomial mutation operator by the operator based on a Cauchy distribution with a non-static mutation probability, which is useful when we have to deal with different optimization problems of the AMR model and thus, where the choice of the value of the mutation probability becomes a difficult issue.

4. The replacement of the Crowding Distance (CD) by an Alternative Crowding Distance (ACD) technique, which involves both the objective and search spaces, in

order to find more diverse solutions. This feature is required for solving the optimization problems of the AMR model, where several non-dominated solutions can be identical in the objective space and different in the search space, but the both can present interesting solutions.

The structural core of FastEMO is straightforward, which allows us to easily parallelize the evaluation of n solutions. In this thesis, the evaluation functions unit is considered as a CPU-parallel part.

Archive Size: The recommended value for the archive size is the following:

- for generations $t < T_{max} - 1$:
  - for small population sizes ($n \approx 100$): $a = 10 \cdot m$, where m is the number of objectives;
  - for large population sizes ($n \approx 1000$): $a = 15 \cdot m$;
  - for very large population sizes ($n \approx 10000$): $a = 20 \cdot m$.

  As seen, the archive size for large and very large population sizes is slightly larger than it was recommended for ASREA, but it is efficient for improving the variance and the population scalability. The proposed range of archive sizes was determined experimentally;
- for generation $t = T_{max} - 1$: a is the size of the child population, which is the same as the initial population size defined by the user.

Computational Complexity :

- Computational complexity for a single generation (t) from $t = 1$ to $t = T_{max} - 1$: $\mathcal{O}(man + mn) + \mathcal{O}(mn\log(n)) + \mathcal{O}(dn\log(n))$, where: (i) $\mathcal{O}(man)$ is the computation complexity of ranking assignment borrowed from ASREA; (ii) $\mathcal{O}(mn)$ is the additional computation complexity for converting the solutions to another objective space by applying Control the Dominance Area of Solutions (CDAS) [Sato et al. (2007)] in order to improve the effectiveness of the conventional Pareto dominance on the high-dimensional target space; (iii) $\mathcal{O}(mn\log(n))$ is the classical target space-based crowding distance; (iv) $\mathcal{O}(dn\log(n))$ is the search space-based crowding distance; (v) m is the number of objectives, a is the size of the archive and n - the population size.
- The computational complexity for the last generation $t = T_{max}$: $\mathcal{O}(mn^2 + mn) + \mathcal{O}(mn\log(n)) + \mathcal{O}(dn\log(n))$.

As we can see, due to the computation of the search space-based crowding distance and the conversion operator, the computational complexity of FastEMO is slightly higher than that of ASREA. However, we want to emphasise that FastEMO achieves needed speedup for parallel implementation by its relatively low computational complexity and improved population scalability (Section 4.3.2).

Discussion: The conducted comparative study between FastEMO and ASREA confirms that the modifications presented in this chapter improve the performance of the algorithm compared with the original version (Section 4.3.1) in terms of the hypervolume indicator and reduced variance of the results.

It is experimentally demonstrated that the efficiency of FastEMO does not degrade with the increase of population size (Section 4.3.2), which confirms the scalability of FastEMO w.r.t. the population size.

FastEMO and NSGA-II demonstrate a very similar scalability w.r.t. the dimension of the objective space on the problems of the WFG test suite: for each dimension, the obtained values of hypervolume indicator by both algorithms are very close. However, on average, FastEMO is 3.4, 3.2 and 2.5 times faster than NSGA-II for three-, four- and five-objective problems respectively (Section 4.3.2).

We assume that the scalability of FastEMO w.r.t. the dimension of the objective space can be further improved by introducing a sophisticated technique for changing the value of the parameter S_i of the converting operator, and hereby, the convergence can be regulated. As it was explained, the technique of CDAS changes the dominance relation of solutions and therefore, modifies the distribution of the fronts. In the case of FastEMO, it means that the number of the non-dominated solutions is changed according to the value of parameter S_i of the converting operator. We found and defined the ranges of the S_i value for different dimensions of the objective space. This method allows the algorithm to improve its performance comparatively to the conventional Pareto dominance method in terms of I_{HV} in about 1.2 and 1.4 times for four- and five-objective problems respectively. We suppose that an adaptation of the value of parameter S_i according to the performance during optimization can significantly help to outperform current results.

On the bi-objective problems of the BBOB test suite for a 20-dimensional search space, the speedup of FastEMO in the first stage is less pronounced comparatively with that of MO-CMA-ES, NSGA-II and SPEA-II. This result is predictable due to a quite large population size ($n = 1000$) that was used. It is not considered as negative result, because all solutions in the population have to be evaluated in parallel and the time to evaluate one generation equals the time required to evaluate one solution. Consequently, in case of FastEMO, the runtime strongly depends on the number of generations rather than on the total number of fitness function evaluations, in order to win the overall execution time.

A positive aspect in the use of a large population size is that it can be useful for multimodal problems, for supporting the evolutionary search on high dimensional search spaces and for obtaining alarger number of non-dominated solutions. The latter is required for the optimization of the Active Magnetic Regenerator (AMR) model.

Although the comparision with the state-of-the-art algorithms demonstrates a better speedup in terms of number of function evaluations at the first stage of the optimization process, FastEMO solves a larger number of overall problems on bi-objective BBOB functions with $d = 5$, $d = 20$ and $d = 40$ dimensions of the search space.

Perspectives: Taking into account that a programming language migration of the SCILAB source code of the Active Magnetic Regenerator (AMR) model to C++ is considered, a long term perspective for further research is an implementation of a GPU-based version of FastEMO, where FastEMO can exploit its population scalability to the magnitude of 10000 solutions.

A short term perspective is to better handle non-separable functions, as the parameters of control and design of the AMR model can be correlated. This issue is going to be solved by improving the Cauchy-based mutation operator: the uni-variant Cauchy distribution has to be changed by the multi-variant, presented in [Lee & Park (2014)], which is demonstrating its usefulness for problems with correlated decision variables.

Another short term perspective is to improve the performance of FastEMO on many-objective problems. It is necessary to introduce a special control element to the algorithm design for tracing the performance, like a feedback "trigger", which controls the performance of the algorithm over time on the fly. Further research will consider to apply such kind of element in order to adjust the value of the control parameter of CDAS during the optimization.

Connection with the Research Problems: In the frame of this thesis, FastEMO is developed to solve optimization problems of the dual-mode operating model of AMR presented in Section 2.3.3. However, FastEMO can be applied for optimization problems of the model of Hamiltonian presented in Section 2.2.2 for simulating physical properties of Magneto Caloric Materials (MCMs). The application of FastEMO to solve the problems of the model of Hamiltonian and of the AMR model are presented in Section 6.2 and in Section 6.3 respectively.

Note that FastEMO is not tested on real word problems with massive parallelism (i.e., thousands of computing units) due to the fact that both models do not support GPU execution yet. Consequently, the scalability of FastEMO w.r.t. population size cannot be employed on 100% in the current work.

In order to make FastEMO publicly available, easy to use and cross-platform, we integrated FastEMO to the EASEA 2.20 platform. One can find more details about EASEA 2.20 in Section 6.1.1.1.

Chapter 5

Quantum-inspired Algorithms

Following the research strategy defined in Section 1.2, in this chapter, we conduct two following studies to address the development of quantum-inspired optimization algorithms for solving the problems of the Hamiltonian model of Magneto Caloric Materials (MCMs) and the model of Active Magnetic Regenerator (AMR), taking into consideration the defined features associated with these models:

1. The first study presented in Section 5.1, aims at investigating an applicability of the Diffusion quantum Monte Carlo (DMC) method presented in Section 3.4.2, as a basis of an optimization algorithm for solving time-consuming separable and partially separable small-scaled problems of the Hamiltonian model for simulating/studying physical properties of MCMs.

   The motivation for this study originates from the two following hypotheses, which are connected to each other:

   (a) as the model for simulating physical properties of MCMs has a quantum structure and is given in the form of the Hamiltonian, theoretically, it seems to be advisable to use subject-oriented research tools, like the DMC method, for optimizing the properties of such models, because these tools can improve the performance of the optimization process by bringing in it quantum properties;

   (b) a new quantum-inspired optimization algorithm based on the DMC method, which we hope to get as a new scientific contribution, can be used for optimizing different separable and partially separable small-scaled problems.

   In Section 5.1, we analyse the DMC method as a baseline technique for an optimization algorithm and justify its applicability for solving optimization problems. Then we present a new single-objective optimization quantum-inspired algorithm based on the DMC method with an integrated evolutionary strategy in its structure and experimentally verify our hypotheses by validating the performance of the proposed algorithm on the noiseless single-objective Black-Box Optimization Benchmarking (BBOB) [Hansen et al. (2012)] problems.

2. The second study presented in Section 5.2 aims at developing an optimization algorithm that is invariant w.r.t. changes of dimension of the target (objective) space and the search space. By this invariance, we mean the scalability of the algorithm

w.r.t. the number of objectives and number of input variables of the optimization problem.

Further in this thesis, the developed algorithm is called "unified", because this term was introduced in [Seada & Deb (2014)] and is usually used for the description of the algorithms, which aim at solving three types of optimization problems: mono-objective, multi-objective and many-objective.

Such kinds of algorithms are presented and discussed in the literature [Deb & Tiwari (2008)], [Seada & Deb (2014)]. In this thesis, the necessity for a unified algorithm comes from the fact that optimization problems have to take into consideration many configurations of the models with different numbers of studied properties or performance metrics (i.e., the number of objectives) and their parameters (i.e., the number of input decision variables).

In order to develop such kind unified algorithm, we employ a fusion method presented in [Ibrahim, Martin, Rahnamayan & Deb (2017)] and discussed in Section 3.2.2, which allows to combine (to fuse) the solutions of several algorithms working simultaneously. Usually, the fusion method is applied to reduce the the challenge of choosing one optimization algorithm to solve complex problems by obtaining the aggregate search characteristics, which are higher in comparison with each algorithm separately.

In Section 5.2, we apply fusion method for providing a new unified algorithm, by combining two following algorithms: a many-objective genetic algorithm NSGA-III [Deb & Jain (2013)] (see Section 3.4.5) and a single-objective quantum-inspired algorithm QPSO [Sun et al. (2007)] (see Section 3.4.4), which have significant differences in their working mechanisms. Such a composition allows the proposed fusion-based algorithm to adapt itself to different dimensions of the objective space. As well, we hope to get better scalability w.r.t. the dimension of the search space.

Compiling all together, we want to contribute by making a step forward the further development of quantum-inspired optimization algorithms and to provide some evidence of the benefits that quantum physics might offer for evolutionary optimization.

5.1 Quantum-inspired Algorithm with Evolution Strategy: QAES

The Diffusion quantum Monte Carlo (DMC) method presented in Section 3.4.2, is a powerful stochastic tool, which is widely applied for finding the ground state expectation values of quantum systems, such as the Hamiltonian model of Magneto Caloric Materials (MCMs). Following the hypothesis mentioned in the introduction of this section, we want to investigate its optimization properties, in order to employ it for finding the global optimum of different separable/partially separable problems of the Hamiltonian model that are aiming at simulating/studying physical properties of MCMs. The explanation of the reasons, which stay behind this hypothesis is provided in the following sub-section.

In this work, the quantum process of the DMC method is not just emulated on classical hardware. We try to take out the best of quantum physics and to improve the

optimization properties of DMC by integrating some conventional optimization technique in its structure.

Thus, we propose a new single-objective quantum-inspired optimization algorithm, which is based on the DMC method and (1+1)-Evolution Strategy described in Section 3.4.2 and Section 3.4.3 respectively. We suppose that this algorithm can be used not only for solving the problems of the Hamiltonian model of MCMs, but also for other different separable/partially separable small-scaled optimization problems.

To our best knowledge, an adaptation of the DMC method for solving optimization problems in continuous search space has never been done before and it is a new contribution. We argue that this work can help to understand some challenges and some key ingredients for a successful design of quantum-inspired algorithms.

The proposed algorithm called Quantum-inspired Algorithm with Evolution Strategy (QAES) and has been published in [Ouskova Leonteva et al. (2020)].

This section is organized in the following way: in Section 5.1.1 we provide the pseudocode of the DMC algorithm used in this work and justify its applicability for solving optimization problems. Next, we propose a new quantum-inspired algorithm, called QAES, which is based on the DMC method and the (1+1)-Evolution Strategy in Section 5.1.2. The proposed algorithm is experimentally validated on the noiseless single-objective Black-Box Optimization Benchmarking (BBOB) [Hansen et al. (2012)] problems and a partially separable single objective problem of harmonic analysis in Section 5.1.3. Finally, Section 5.1.4 presents a discussion with some perspectives for future research.

5.1.1 DMC algorithm

The description of the Diffusion quantum Monte Carlo (DMC) method is presented in Section 3.4.2, which is based on the following references: [Foulkes et al. (2001)], [Kent (1999)], [Kosztin et al. (1996)], [Toulouse et al. (2016)], [Zen et al. (2016)].

Basically, the DMC method is a projector stochastic method that uses the similarity between the imaginary time Schrödinger equation and a generalized diffusion equation, which is solved using stochastic calculus and simulating a random walk. As seen from Figure 3.12 in Section 3.4.2, the DMC algorithm maps onto a time varying population of particles (walkers), where the particles in energetically favorable locations are duplicated and the particles in unfavorable locations are removed. After a projection to the ground state, branching random walk has the greatest density where the wave function (probability) is the largest.

Thus, we can assume that if the DMC method has an ability to accurately approximate the ground state energy for any kind of quantum systems, which is the attained minimum value of potential of a given system, we can try to apply this quantum method for finding the minimum of an optimization problem.

In this section, we present the study of the DMC method as an optimization algorithm, which consists of three stages:

1. proof of the presence of optimization properties in the DMC method;
2. development and research of a simulation algorithm that implements the DMC method;
3. experimental studies of the DMC algorithm on benchmark optimization problems.

5.1.1.1 Hypothetical Optimization Properties of DMC

If the DMC method can find the wave function and the minimum energy corresponding to the ground state of a given quantum system, theoretically, it can be considered as an optimization algorithm to find the optimum of a given single-objective optimization problem, where the calculated value of the objective function is considered as the value of the potential energy.

More precisely, the main idea behind the application of the DMC method as a driving force for a quantum-inspired optimization algorithm is that the maximum value of the ground state wave function corresponds to the global minimum of the potential energy to which the particle is subjected. It means that if the ground state wave function is found by the DMC method, there is a high probability to find the coordinates of a particle, which is close to the global minimum of the potential energy. If an objective function is taken as the potential energy (the global minimum of which is to be found), the solution will be found in the global minimum of the objective function with a maximum probability, when the ground state wave function is found by DMC.

Going further with this hypothesis, we believe that a quantum-inspired algorithm based on the DMC method can be employed to find the global minimum of different small-scaled problems due to following characteristics of DMC, which were shown in different works [Kent (1999)], [Kosztin et al. (1996)], [Toulouse et al. (2016)], [Zen et al. (2016)]:

1. DMC is a very general approach, able to calculate almost any ground-state expectation value, including energies;

2. DMC has the significant computational advantages of easily achieved scalability on parallel architectures and low storage requirements (this is an important point for the feature work);

3. DMC has a very advantageous zero-variance property: as the wave function approaches the exact ground state or any other exact energy eigenstate, the statistical fluctuations in the energy reduce to zero.

Furthermore, the DMC method is used to study the models of quantum systems presented in a form of a Hamiltonian. In this context, we believe that there could be a potential advantage and scientific interest to explore the application of the DMC method to construct an algorithm for optimizing the parameters of the model of Magneto Caloric Materials (MCMs), which is represented in the form of a Hamiltonian. Taking into account the structure of the Blume–Emery–Griffiths–Ising (BEG-I) Hamiltonian model of MCMs presented in Section 2.2.2.1, we are interested in an optimization algorithm, which is able to solve separable/partially separable small-scaled problems, in order to apply it for simulating/studying physical properties of MCMs.

If the optimization properties of the DMC method are proved, it can be implemented as a quantum optimization algorithm on a real quantum hardware in the future.

5.1.1.2 DMC Simulation Algorithm

Earlier, in Section 3.4.2, we provided in details the description of the DMC method, where a quantum system is given by the Hamiltonian $H(\mathbf{R})$ (see Equation 3.5) and its configuration ($\mathbf{R}^{(\tau)} = (\mathbf{r}_1^{(\tau)}, \mathbf{r}_2^{(\tau)}, ..., \mathbf{r}_N^{(\tau)})$) every time τ is represented by a $N^{(\tau)}$-dimensional vector,

specifying the coordinates of $N^{(\tau)}$ particles, and each i-th particle ($\mathbf{r}_i^{(\tau)}$) is represented by a d-dimensional set of coordinate (where $d = 3$, i.e., x, y and z).

For implementing an optimization algorithm based on the DMC method, we selected the simulation method originally presented in [Kosztin et al. (1996)], and described in Section 3.4.2.2, as one of the simplest and efficient method proposed in the literature, which avoids several issues of DMC. For example, one of the problems of the DMC method is that the branching process causes fluctuations in the number of particles N (the population size of solutions) and consequently, can lead to computational overheads: i.e., nothing prevents the population size from decreasing or increasing indefinitely during the Monte Carlo iterations. To escape from this, the selected method presented in [Kosztin et al. (1996)], is forcing the number of particles not to deviate too much by introducing a population control mechanism, where the population size N fluctuates between generations and these fluctuations are controlled by the values of E_T and α. These details were discussed in Section 3.4.2.2.

In this section, the selected simulation method of the DMC method is implemented as an algorithm, which is supposed to be employed for optimization. Below, in Algorithm 9, we present the pseudo-code of our implementation of DMC simulation in accordance with the description provided in Section 3.4.2.2. The notations used in Algorithm 9 are taken from Section 3.4.2.2 and summed up in Table 5.1.

As described in Section 3.4.2.2, the potential-dependent increase or decrease of particles density is defined by the velocity term $(V_p(\mathbf{R}) - E_T)$ of the weight function presented by Equation 3.16 in Section 3.4.2.2. Thus, as a result of the presented pseudo-code, the two following parameters are obtained: the ground state wave function, which is interpreted as the density of diffusing particles, and energy of the system, which is found from averaging the successive reference energies E_{T} (Equation 3.22 in Section 3.4.2.2). These two parameters give the most complete description of a quantum system, so they can be considered as its optimal parameters. So, we can assume that DMC has the ability to find optimal solutions and this fact presents a scientific interest for us.

Theoretically, an optimization process based on DMC, can be very efficient, if the importance sampling technique presented in Section 3.4.2.2 is applied. As it was explained in Section 3.4.2.2, the solutions are attracted in the area of the search space, where the guiding wave function (i.e., trial wave function) has the largest values, and consequently, the minimum value of a given optimization function (potential energy) is expected to be found there, if the guiding wave function is selected/prepared correctly. Consequently, the importance sampling can be taken as an auxiliary technique, which allows to incorporate the assumed knowledge of the problem in the optimization algorithm.

However, we do not integrate the importance sampling technique in the proposed quantum-inspired algorithm, because of the following constraints:

1. time-consuming variational methods are needed to prepare the initial distribution, which is used as a guiding wave function in the DMC. This preparation also requires many calls of a given optimization function that is critical, when this problem is computationally intensive;

2. the guiding wave function depends on a given optimization problem, and consequently, it has to be prepared every time, when the optimization function is modified;

Algorithm 9: DMC - Diffusion quantum Monte Carlo.

```
Initialize input parameters:
N^(τ=0) > 2; N_0 = N^(τ=0); δτ > 0; α = 1/δτ; σ = √δτ; T_max > 0;
Initialize other parameters:
T_nequil = floor(0.4 * T_max + 0.5); τ = 0; E_T^(τ=0) = 0; E_0^(τ=0) = 0; Ψ_0 = 0;
Initialize randomly R^(τ=0) /* the set of coordinates of N particles */;
Evaluate V^(τ=0) /* the set of potential energy */ ;
Calculate E_T^(τ=0) by Equation 3.17 in Section 3.4.2.2;
/* Monte Carlo Loop */
for τ = 0, ..., T_max do
    N^(τ+1) = N^(τ) /* set the previous number of particles before branching */
    /* For each particle r_i^(τ) */
    for i = 0, ..., N^τ do
        Diffusion displacement:
        V_i^(τ)(r_i^(τ)); /* Get the value of potential energy for r_i^(τ) */
        r_i^(τ+1) = r_i^τ + σ · ρ_i /* Transition according to Gaussian distribution (see
          Equation 3.18 in Section 3.4.2.2) */
        V_i^(τ+1)(r_i^(τ+1)); /* Get the value of potential energy for r_i(τ + 1)*/
        Branch:
        /* Branch Green function */
        Compute W(r_i^(τ+1), V_i^(τ), V_i^(τ+1), E_T^(τ)) by Equation 3.16 in Section 3.4.2.2;
        Compute m_i by Equation 3.19 in Section 3.4.2.2;
        Update the number of walkers in ensemble:
        if m_i > 1 then
            /* r_i is copied m_i-times: */
            for j = 0, ..., m_i − 1 do
                /* Since one already exists, add m_i − 1 */
                N^(τ+1) + +;
                r_{N^(τ+1)}^(τ+1) = r_i^(τ+1)
            end
        end
        else
            if m_i < 1 then
                /* Particle r_i disappears */
                r_i^(τ+1) = r_N^(τ+1) /* Set the position of i-th particle as the position of
                  the last particle */
                N^(τ+1) − −;
            end
        end
        Calculate E_T^(τ+1) by Equation 3.20 in Section 3.4.2.2;
    end
    if τ > T_nequil then
        /* Equilibrium period */
        Count:
        Compute Ψ_0 by Equation 3.21 in Section 3.4.2.2;
        Compute E_0 by Equation 3.22 in Section 3.4.2.2;
    end
end
```

Table 5.1: Notations of DMC Algorithm.

Notation	Explication	Value
τ	time	$\tau \in \mathbb{R}_+$
$\delta\tau$	time step	$\delta\tau \in [0, 1]$
T_{max}	max. number of Monte Carlo steps	$T_{max} \in \mathbb{R}_+$
T_{nequil}	non-equilibrium number of Monte Carlo steps	$T_{max} \in \mathbb{R}_+$
$N^{(\tau)}$	current number of particles	$N \in \mathbb{N}_+, N > 2$
N_0	initial number of particles	$N_0 \in \mathbb{N}_+$
i	index of particle	$i \in \mathbb{N}_+, i \in \{0, N^\tau\}$
$\mathbf{R}$	set of coordinates of N particles	$\mathbf{R} = (\mathbf{r}_1, \mathbf{r}_2, ...\mathbf{r}_N)$
$\mathbf{r}_{min}$	set of min. values of coordinates	$\mathbf{r}_{min} \in \mathbb{R}$
$\mathbf{r}_{max}$	set of max. values of coordinate	$\mathbf{r}_{max} \in \mathbb{R}$
$\mathbf{r}_i^\tau$	set of coordinates of i-th particle	$\mathbf{r}_i \in \{\mathbf{r}_{min}, \mathbf{r}_{max}\}$
$\boldsymbol{\rho}^{(\tau)}$	set of Gaussian random numbers	$\boldsymbol{\rho}^{(\tau)} = \mathcal{N}(0, 1)$
m_i	coefficient of birth–death of i-th particle $\mathbf{r}_i$	$m_i \in \mathbb{N}_+$
α	"feedback" parameter	$\alpha \in \mathbb{R}_+$
σ	diffusion displacement step	$\sigma \in \mathbb{R}_+$
V_i^τ	potential energy of $\mathbf{r}_i^\tau$	$V_i^\tau \in \mathbb{R}$
$\mathbf{V}^\tau$	set of potential energy of N particles	$\mathbf{V}^\tau = (V_1^\tau, V_2^\tau, ..., V_N^\tau)$
E_T^τ	offset energy	$E_T^\tau \in \mathbb{R}$
$\boldsymbol{\Psi}_0$	ground state wave function	$\boldsymbol{\Psi}_0 \in \mathbb{R}$
E_0^τ	ground state energy	$E_0^\tau \in \mathbb{R}$
$W(\mathbf{r}_i^\tau)$	value of weight function of i-th particle	$W(\mathbf{r}_i^\tau) \in \mathbb{R}_+$
$\mathcal{U}$	Uniform distribution	
$\mathcal{N}$	Gaussian distribution	

3. the mentioned above circumstances significantly complicate the use of the importance-sampled DMC method for the study of the model of Magneto Caloric Materials (MCMs), since for each change of the model, which is required for different types of materials, it is needed to apply the variational method to obtain a new guiding wave function;

4. the Green function presented in Section 3.4.2.1 (see Equation 3.11), is usually a reasonable approximation in regions where the guiding wave function is smooth and non-zero. However, importance sampling may become problematic whenever particles are in the regions where the guiding wave function takes on very small values: in these cases, the local energy and the quantum force (the drift velocity) may exhibit divergent behaviour, and the approximate Green function needs to be refined.

In this thesis, we investigate the possibility of using a "pure" DMC method to construct an optimization algorithm for finding the parameters of a given Hamiltonian model of MCMs, as well as to test this algorithm for optimizing different beanchmarking functions. However, the development of an optimization algorithm based on the DMC method with

the importance sampling technique in order to increase its efficiency may be the subject of another interesting research.

5.1.1.3 Applying DMC to the Optimization Process

Since the DMC method is implemented, its optimization properties can be investigated.

In order to address the objective to construct an optimization algorithm based on the DMC method, we establish the proposed below correspondences between the elements of a quantum system $H(\mathbf{R})$ and the elements of a single-objective optimization problem $f(\mathbf{x})$.

Analogies between the DMC method and a classical optimization algorithm:

1. number of Monte Carlo steps = number of generations;
2. current number of particles in the system N = population size N at each generation;
3. each particle position ($\mathbf{r}_i$) = each candidate solution ($\mathbf{x}_i$);
4. set of coordinates of a particle $\mathbf{r} = (r_1, r_2, ..., r_d)$, where d is the number of coordinates = set of decision variables of a candidate solution $\mathbf{x} = (x_1, x_2, ..., x_d)$, where d is the dimension of the search space;
5. value of the potential energy of each particle $V_p(\mathbf{r}_i)$ (a performance index of each particle position) = value of the objective function ($f(\mathbf{x})$) of each candidate solution (a performance index of each solution);
6. offset energy E_T and value of the "feedback" parameter α = control mechanism of fluctuations of the population size at each generation;
7. diffusion displacement = mutation;
8. $V_p(\mathbf{R}) - E_T$ = rate of convergence (this value routes the particles in the regions of the search space, where the wave function is increasing);
9. the wave function of the ground state = the distribution of solutions, where the value of an optimization function f is optimal (minimal).

Experiment: verification of the optimization property of DMC

Since the code of the DMC method is implemented and the analogies are defined, we apply the DMC algorithm to solve three single-objective test problems for one-dimensional search space presented in Table 5.2.

For each optimization problem, we run the DMC algorithm 30 times, where the equation of potential energy $V(r)$ in Algorithm 9 is defined as the equation of the current test problem $f(x)$ presented in Table 5.2. For all tests, the same input parameters are used: $N_0 = 100$, $T_{max} = 1000$, $\delta\tau = 0.05$, $\alpha = 1$.

The results of the tests are presented in Figure 5.1. In Figure 5.1 (i), we can see the plots of the potential energy $V(r)$, i.e., the optimization function $f(x)$ in one-dimensional search space, where the horizontal axis presents the one-dimensional coordinate of particles ($r = x$) and the vertical axis is the value of $f(x) = V(r)$. Figure 5.1 (ii) depicts the

Table 5.2: Test functions.

Function	Equation	Global minimum	Search domain
Sphere	$f(\boldsymbol{x}) = \sum_{i=1}^{d} x_i^2$	$f(0,\ldots,0) = 0$	$-\infty \leq x_i \leq \infty$ $1 \leq i \leq d$
Rastrigin	$f(\mathbf{x}) = Ad + \sum_{i=1}^{d}(x_i^2 - A\cos(2\pi x_i))$ where: $A = 10$	$f(0,\ldots,0) = 0$	$-5.12 \leq x_i \leq 5.12$ $1 \leq i \leq d$
Styblinski–Tang	$f(\boldsymbol{x}) = \frac{\sum_{i=1}^{d} x_i^4 - 16x_i^2 + 5x_i}{2}$	$f(-2.9,\ldots,-2.9) = -39.1d$	$-5 \leq x_i \leq 5$ $1 \leq i \leq d$

wave function of the ground state (Ψ_0) of the system, i.e., the density distribution of the solutions corresponding to the minimum of the optimization function $f(x)$: a) Sphere, b) Rastrigin, c) Stybliksi-Tang (where the horizontal axis is the coordinate of particles and the vertical axis is the value of the wave function). As we can see from Figure 5.1 (i) and (ii), the wave functions of the ground state for all test functions correspond to the minimum value of the potential energy and the value of coordinate corresponds to the value x of global minimum of $f(x)$ presented in Table 5.2. E.g., from Figure 5.1 (i, c) it is seen that the value of the global optimum $f(x) = -39.1$ and from Figure 5.1 (ii, c), it is clear that the maximum distribution is observed near $x = -2.9$.

According to the obtained results, we can state that our assumption to present the objective function as a potential energy seems to be correct and our hypothesis to employ the DMC algorithm for optimization problems is justified in general. We have to emphasize that in Figure 5.2 (ii), we present the results as the mean value of the wave function over 30 runs for each test problem.

Since the DMC method can solve different optimization problems on a one-dimensional search space, let us discuss its potential issues as an optimization algorithm.

Issues of the DMC method:

Theoretically, applying the DMC method as an optimization algorithm, we can assume the following possible challenges:

1. The DMC is limited by the number of the coordinates of particles, because, in a quantum system, each particle has 3 coordinates (x, y, z).

   Thus, the DMC method has to be transformed into a quantum-inspired algorithm, which allows to work with a larger search space than three coordinates (the decision variables) for each particle (candidate solution). This algorithm is assumed to solve only small-scale problems. Taking into account that the DMC-based algorithm is assumed to be employed for solving the problems of the Hamiltonian model of Magneto Caloric Materials, which are small scaled (less than 10), the lack of scalability w.r.t. the dimension of search space seems to be acceptable.

2. The ground state energy of a quantum system is not identical to the minimum of the potential. E.g., as it is shown in [Kosztin et al. (1996)], in the case of one-dimensional harmonic oscillator model, the ground state energy $E_0 = 0.5$ is the minimal possible value of a quantum system energy. Consequently, particles can not

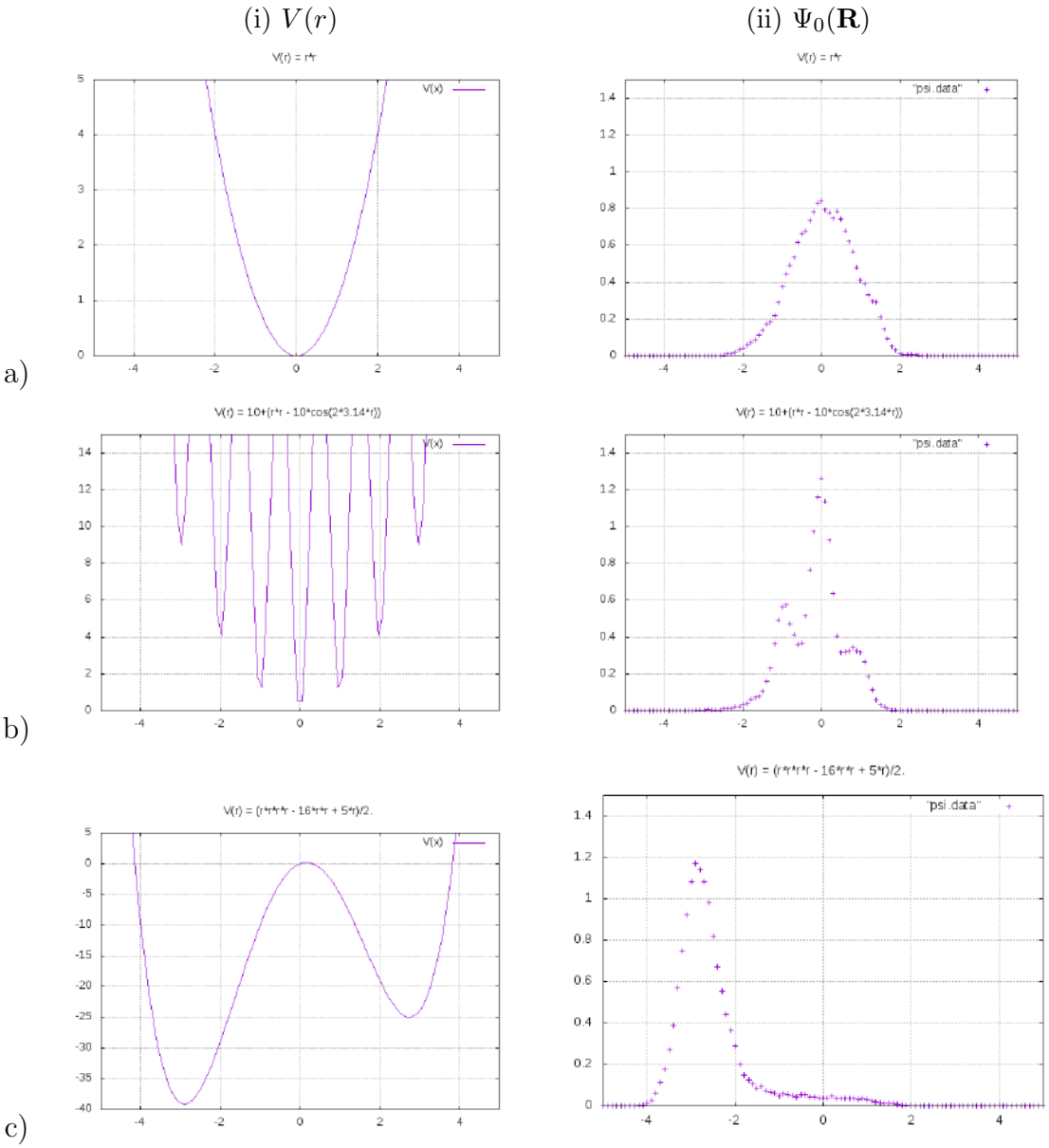


Figure 5.1: (i) Potential energy $V(r)$, i.e., the optimization function $f(x)$ in one-dimensional search space, where the horizontal axis is the one-dimensional coordinate of particle r (i.e., the decision variable x) and the vertical axis is the $f(x) = V(r)$; (ii) the wave function of the ground state (Ψ_0) of the system, i.e., the density distribution of the solutions corresponding to the minimum of the optimization function $f(x)$: a) Sphere, b) Rastrigin, c) Stybliksi-Tang (where the horizontal axis is the coordinate of particles and the vertical axe is the value of the wave function).

achieve the bottom of the potential well (where the potential energy = 0), because it contradicts the uncertainty principle of Heisenberg. For the above mentioned reason, DMC cannot achieve the global optimum of f.

However, as seen from the experimental results presented in Figure 5.1, DMC can find the best probability, where the distribution of the particles corresponds the global optimum of f. Thus, some additional optimization technique is needed for further sampling the particles distributed near the optimum and consequently, for finding the global optimum. We prove this assumption by the experiment presented and explained below (Figure 5.3 (b)).

3. The rate term $V_P(\mathbf{R}) - E_T$, which routes the particles in the regions of the search space where the wave function is increasing, can be very large. For example, it is possible, in the case of ill-conditioned problems (e.g., the non-rotated ellipsoid function from the Black-Box Optimization Benchmarking (BBOB) test suite [Hansen et al. (2009)]). In non-mathematical terms, it means that a small change in the inputs (i.e., the decision variables) can corresponds to a large change in the function value. Thus, for such kinds of function, DMC can not find the ground state wave function, because all particles can be removed at the begining of the optimization. Consequently, the term $V_P(\mathbf{R}) - E_T$ has to be normalized.

4. As mentioned in the analytical study presented in Section 3.4.2.2, the DMC simulation process terminates if the value m for all particles in the branch process is less than zero.

   In order to prevent this possibility, one needs to choose the initial location of the particles with care. In other words, the initial distribution of particles have to be as close as possible to the wave function of the ground state. Maybe one of the simplest way is to change the distribution in the diffusion displacement step, which has to be close to the optimization function.

5.1.2 Proposed Quantum-inspired Algorithm

In order to mitigate the issues presented in the previous section, we propose a quantum-inspired single-objective optimization algorithm based on the Diffusion quantum Monte Carlo (DMC) method with integrated (1+1)-Evolution Strategy (ES), which is called QAES. The pseudo-code of QAES is provided in Algorithm 10. The notations used in this pseudo-code are mainly the same that were used in Algorithm 9 and presented in Table 5.1. The additional notations, which are specific for QAES are provided in Table 5.3.

As seen from Algorithm 10, following the working principle of the DMC method, QAES performs two main iterative processes, Diffusion Displacement and Branching, inside the simulation loop, aiming at obtaining the distribution of the particle, which corresponds to the ground state wave function. In Algorithm 10, instead of the potential energy, we use a given optimization function $f(\mathbf{x})$ and the set of decision variables is assumed as the vector of the particle's coordinates.

Comparatively with the pseudo-code of DMC provided in Algorithm 9, the following modifications are integrated:

1. The vector of decision variables (the coordinates of a particle) can have any required size (the dimension of the search space d). In this case, each solution consists of

d number of decision variables. Consequently, the diffusion displacement step is executed for each decision variable from x_1 to x_d, and the wave function is defined by the distribution of all solutions with dimension d.

The scalability of QAES w.r.t. the dimension of the search space is investigated during the experimental study provided in Section 5.1.3.3.

2. The potential energy (V_P) is replaced by a given optimization function to be minimised (f). However, as it was explained above, the ground state energy of a quantum system is not identical to the minimum of the potential, and the latter can not be achieved by the "pure" DMC. In order to find the minimum value of an optimization problem (f_{best}), which is considered as the minimum value of the potential energy, we integrate an auxiliary conventional optimization algorithm, the (1+1)-Evolution Strategy (ES), described in Section 3.4.3, into the structure of the DMC method.

   The (1+1)-ES algorithm is selected, because it belongs to one of the most competitive classes of optimization algorithms in the field of continuous black-box optimization area. Even though the (1+1)-ES is drastically simpler than other ES variants, like well-known the Covariance Matrix Adaptation ES (CMA-ES), they share several core features, such as randomness and heuristic step-size adaptation mechanisms. Since in this work, we mainly focus on the algorithms, which can solve separable and partially separable problems, thus, it seems to be reasonable to employ the (1+1)-ES as an auxiliary algorithm.

   As it was explained in Section 3.4.3, the (1+1)-ES has two main components, that are employed inside the DMC structure in Algorithm 10, when the equilibrium period is achieved:

   (a) **greedy selection of a candidate solution**: it selects and keeps the better of the two (new or old) positions of each particle (candidate solution) according to potential energy (the objective function);

   (b) **step-size adaptation:** the sampling distribution is adapted during the course of the optimization, according to the coefficient m: i.e., if $m < 0$ the immediate adaptation of the step size of the diffusion displacement based on "failure" is applied; if $m > 0$ the "success" adaptation rule of this step size is used.

   We propose the following assumptions:

   (a) the step-size adaptation can help to sample the distribution of good solutions obtained during the non-equilibrium part of the DMC;

   (b) the greedy selection can help to rapidly find the minimum of a given optimization problem (the potential energy);

   (c) these modifications do not change the ability of the DMC to find the ground state wave function.

   The experimental validation of the proposed combination of the DMC and the (1+1)-ES comparatively with the "pure" DMC is shown in Figure 5.3 (b) and discussed in the next section.

3. The rate term $(f(\mathbf{x}) - E_T)$ of the weight function W is normalized in order to avoid the disappearing of all particles, when its value is a very large for some problems, especially at the beginning of the oprimization process. I.e., in order to calculate the value of W for each particle, instead of Equation 3.16, we apply the following equation:

$$W(\mathbf{x}_\mathrm{i}^{(\tau+1)} \leftarrow \mathbf{x}_\mathrm{i}^{(\tau)}) = e^{-\left(\left(\frac{f(\mathbf{x}_i^{(\tau+1)})+f(\mathbf{x}_i^{(\tau)})}{2} - E_\mathrm{T}\right)/E_\mathrm{T}\right)\delta\tau} \tag{5.1}$$

.

To make it clear, let us show the results obtained without and with the proposed normalization for ill-conditioned non-rotated ellipsoid function ($f2$) from the Black-Box Optimization Benchmarking (BBOB) test suite [Hansen et al. (2009)].

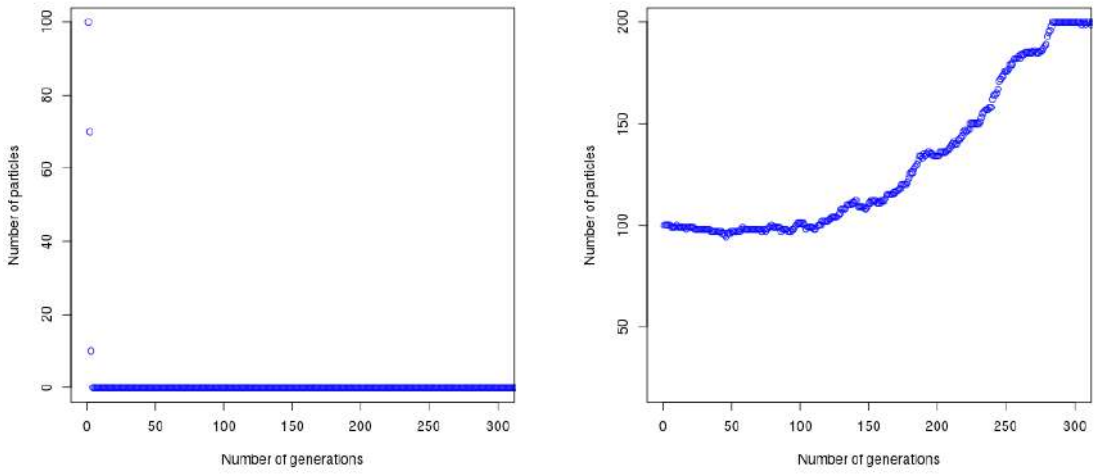
According to the description provided in [Hansen et al. (2009)], $f2$ is globally quadratic, uni-modal and ill-conditioned (conditioning about 10^6) function with smooth local irregularities. For this function, some small changes in the values of the decision variables can correspond to a large changes in the values of the objective function, especially at the beginning of the optimization process. Consequently, the very large value of the term $(f(\mathbf{x}) - E_T)$ causes the particles to be removed. And if all particles are removed, the optimization process will terminate at the beginning and will not achieve the minimum of the objective function. The proposed normalization of the term $(f(\mathbf{x}) - E_T)$ helps to avoid this situation.

In order to prove the efficiency of the introduced normalization, Figure 5.2 shows both scenarios for the ellipsoid function ($f2$) of the Black-Box Optimization Benchmarking (BBOB) benchmarking test suite at the beginning of the optimization process (300 generations): (a) without and (b) with normalization. As we can see in Figure 5.2 (a), without normalization, all particles disappear after the first two generations and the optimization process terminates, while with a normalized value of the term $(f(\mathbf{x}) - E_T)$, the population is growing (Figure 5.2 (b)), as it has to be at the beginning of the optimization process according to the main feature of the DMC algorithm described below.

4. The Cauchy distribution replaces the Gaussian distribution in the diffusion displacement step, due to our previous experience with Cauchy-based mutation operator in Chapter 4, Section 4.2.5. The Cauchy distribution is a heavy tailed distribution, that has no finite mean. So it can help the particles to make long jumps in their coordinates in the diffusion displacement step and consequently, improve the performance.

However, as discussed in Section 4.2.5, the heavy tails of Cauchy distribution can be more efficient in comparison to the exponentially decreasing tails of Gaussian distribution only if the objective function is separable. Taking into account that the problems of the Hamiltonian model of Magneto Caloric Materials (MCMs) can be separable or partially separable, the Cauchy-based mutation operator seems to be preferred.

The experimental validation of an impact of the choice of the distribution on the performance and the efficiency of the Cauchy distribution in the diffusion displacement step for solving multi-modal separable functions is provided in Section 5.1.3.2.



a) without normalization b) with normalization

Figure 5.2: The number of particles (walkers) *vs* the number of generations, during the optimization of Ellipsoid function ($f2$) (a) without and (b) with normalization.

5. The step `Count`, which calculates the ground state energy and ground state wave function, is removed from the algorithm, because their exact values are not needed for the optimization process. Indeed, for optimization, the only interesting value is the potential energy, which is minimal, when the ground state wave function is maximal. This potential energy value is controlled by the (1+1)-ES algorithm integrated in the DMC method. Moreover, it makes it possible to reduce the overall computation complexity of the algorithm.

At the end of the optimization process, the minimum value of the optimization function f_{best} and its best solution ($\mathbf{x}_{best}$) is found.

Table 5.3: Notations used in Algorithm 10.

Notation	Explication	Value
d	dimension of search space	$d \in \mathbb{N}_+$
$\boldsymbol{X}^\tau$	set of solutions	$(\boldsymbol{x}_1^\tau, \boldsymbol{x}_2^\tau, ..., \boldsymbol{x}_N^{\tau\,\tau})$
$\boldsymbol{x}^\tau$	solution	$\boldsymbol{x}^\tau \in \boldsymbol{X}^\tau, \boldsymbol{x}^\tau \in \mathbb{R}^{\mathrm{d}}$
f	objective function	$f \in \mathbb{R}_+$
c_{succ}	success rate for σ adaptation	$c_{succ} > 0$
c_{fail}	failure rate for σ adaptation	$c_{fail} < 0$
$\mathcal{C}$	Cauchy distribution	
$\boldsymbol{\rho}_c$	set of Cauchy random numbers	$\boldsymbol{\rho}_c = \mathcal{C}(0,1)$

Tuning parameters:

QAES, as well as the DMC algorithm, has the following tuning parameters, which have an impact on its performance and can be problem-dependent:

Algorithm 10: Quantum-inspired Algorithm with (1+1)-Evolution Strategy.

Initialize input parameters:
$N^{(\tau=0)} > 2$; $N_0 = N^{(\tau=0)}$; $\delta\tau > 0$; $\alpha = 1/\delta\tau$; $\sigma = \sqrt{\delta\tau}$, d; $T_{max} > 0$;
Initialize other parameters:
$T_{nequil} = floor(0.4 * T_{max} + 0.5)$; $\tau = 0$; $E_{\mathrm{T}}^{\tau=0} = 0$; $E_0^{\tau=0} = 0$; $\mathbf{\Psi}_0 = 0$; $f_{best} = 0$: $\mathbf{x}_{best} = 0$;
Initialize randomly $\mathbf{X}^{(\tau=0)}$ /* *the set of N initial solutions* */;
Evaluate $\mathbf{f}^{(\tau=0)}(\mathbf{X})$ /* *the set of objective functions for initial solutions* $\mathbf{X}$*/ ;
Calculate $\mathrm{E}_T^{(\tau=0)}$ by Equation 3.17 in Section 3.4.2.2;
for $\tau = 0, ..., T_{max}$ **do**
    $N^{(\tau+1)} = N^{(\tau)}$ /* *set the previous number of solutions before branching* */
    for $i = 0, ..., \mathcal{N}^{\tau}$ **do**
        Diffusion displacement:
        $f_i^{(\tau)} = f(\mathbf{x}_i^{(\tau)})$; $\mathbf{x}_i^{(\tau+1)} = \mathbf{x}_i^{\tau} + \sigma \cdot \boldsymbol{\rho}_c$; $f_i^{(\tau+1)} = f(\mathbf{x}_i^{(\tau+1)})$;
        if $\tau > T_{nequil}$ **then**
            if $f(\mathbf{x}_i^{(\tau)}) \leq f(\mathbf{x}_i^{(\tau+1)})$ **then**
                $f(\mathbf{x}_i^{(\tau+1)}) = f(\mathbf{x}_i^{(\tau)})$; $\mathbf{x}_i^{(\tau+1)} = \mathbf{x}_i^{(\tau)}$
            end
            if $f(\mathbf{x}_i^{(\tau+1)}) \leq f_{best}$ **then**
                $\mathbf{x}_{best} = \mathbf{x}_i^{(\tau+1)}$; $f_{best} = f(\mathbf{x}_i^{(\tau+1)})$
            end
        end
        Branch:
        Compute $W(\mathbf{x}_i^{\tau+1}, f_i^{(\tau)}, f_i^{(\tau+1)}, E_T^{(\tau)})$ by Equation 5.1;
        Compute m_i by Equation 3.19 in Section 3.4.2.2;
        Update the number of walkers in ensemble:
        if $m_i > 1$ **then**
            /* $\mathbf{x}_i$ *is copied* m_i*-times*: */
            for $j = 0, ..., m_i - 1$ **do**
                /* *Since one already exists, add* $m_i - 1$ */
                $N^{(\tau+1)} + +$; $\mathbf{x}_{N^{(\tau+1)}}^{(\tau+1)} = \mathbf{x}_i^{(\tau+1)}$
            end
            if $\tau > T_{nequil}$ **then**
                $\sigma^{(\tau+1)} = \sigma^{(\tau)} \cdot e^{c_+}$
            end
        end
        else
            if $m_i < 1$ **then**
                /* *Solution* $\mathbf{x}_i$ *disappears* */
                $\mathbf{x}_i^{(\tau+1)} = \mathbf{x}_N^{(\tau+1)}$ /* *Replace i-th solution by the last one* */
                $N^{(\tau+1)} - -$;
                if $\tau > T_{nequil}$ **then**
                    $\sigma^{(\tau+1)} = \sigma^{(\tau)} \cdot e^{c_-}$;
                end
            end
        end
        Calculate $E_T^{(\tau+1)}$ by Equation 3.20 in Section 3.4.2.2;
    end
end

1. N_0 is the initial number of particles;
2. T_{max} is the number of generations;
3. $\delta\tau$ is the size of diffusion displacement step;
4. α is the parameter, which allows to control the fluctuations of the population size N at each generation.

In this study, we do not apply the hyper-parameter tuning methods and we do not adapt them to each function.

Feature of QAES borrowed from DMC:

Due to the control mechanism of the fluctuation of the population size, the DMC method allows algorithm to "automatically" adjust the number of walkers. At the beginning of the DMC process, the number of walkers is increased, in order to find the ground state; then the number of walkers is decreased to the initial value, when the best probability density distribution of particles and the ground state of quantum system is found. Thereby the number of evaluations of the f function depends on the rate term $V_P(\mathbf{R}) - E_T$ and parameter α.

In order to demonstrate this feature, we apply the DMC algorithm, the pseudo-code of which is provided in Algorithm 9, for solving the test function Sphere presented in Table 5.2. Comparatively to the previous test, we modified only the size of the vector of coordinates, in order to run the algorithm on a 5-dimensional search space.

Figure 5.3 (a) visualises this adjustment process for the Sphere function, where the initial number of particles is 20. We can see that the number of particles is increasing while the ground state is not found. When the ground state energy is obtained (the value of offset or reference energy stabilized and slightly fluctuates around value 83 in Figure 5.3 (b)), the number of particles decreases to the initial number.

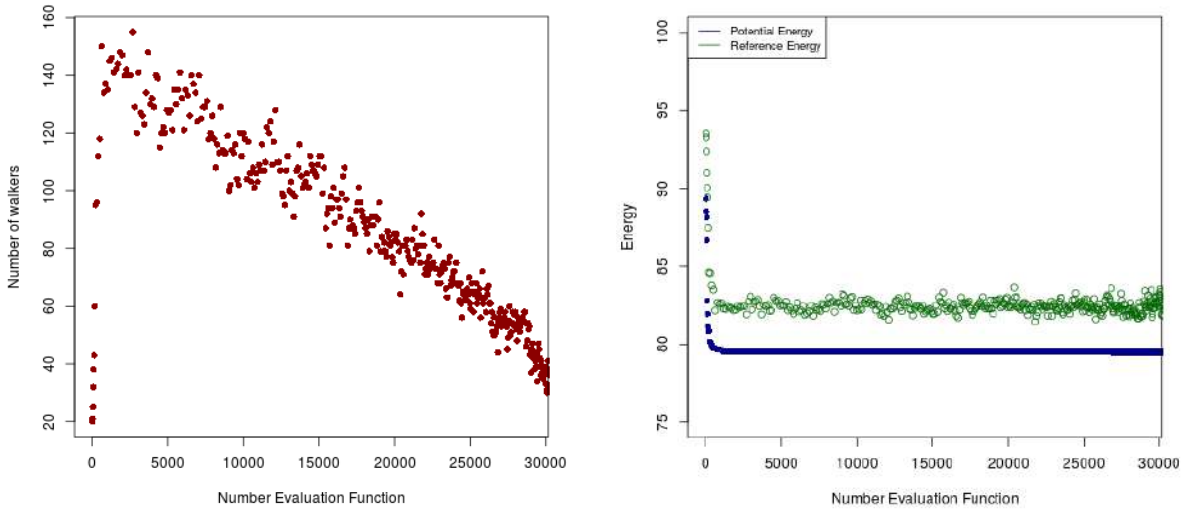
Thus, as seen in Figure 5.3, at the beginning of the algorithm, typically, the number of particles is unstable: the population size becomes larger than the initial size that corresponds to a decrease of the offset (reference) energy (see Figure 5.3 (b)). The control mechanism of the population size reduces the population size by removing the particles with high potential values.

Using the same experiment, we can show the difference between the ground state energy and the potential energy. We can see in Figure 5.3 (b), that the ground state energy values (the green points), obtained by "pure" DMC are not identical the potential energy values (the blue points), which obtained by the quantum-inspired algorithm, QAES, which is based on the DMC method and presented in Section 5.1.2, and which corresponds to the optimum value of $f(\mathbf{x})$ for the Sphere function provided by BBOB [Hansen et al. (2009)].

5.1.3 Experimental Validation

Experimental Objectives:

1. Demonstrate the applicability of the Deffusion quantum Monte carlo (DMC) method for the optimization process.



a) Nb. of walkers vs nb. of evaluations b) Energy vs nb. of evaluations

Figure 5.3: Adjustment of the number of walkers according to energy change in DMC.

2. Confirm the impact of type of the probability distribution in the Diffusion Displacement step on the overall performance.

3. Investigate the robustness of QAES w.r.t. different properties of the optimization functions.

4. Validate the performance of QAES, comparatively to the results of "pure" DMC, the Quantum-behaved Particle Swarm Optimization (QPSO) algorithm [Sun et al. (2007)] and two reference algorithms from the Black-Box Optimization Benchmarking workshop: Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm [Ros (2009)] and Bi–Population Covariance Matrix Adaptation Evolution Strategy (BIPOP-CMAES) [Hansen (2009)], on the single-objective noiseless Black-Box Optimization Benchmarking (BBOB) benchmark suite.

5. Validate the performance of QAES on partially separable real world application problem of harmonic analysis.

Algorithms and Parameters:

The C++-based code of DMC and QAES were developed according to Algorithm 9 and Algorithm 10 respectively. The implementation of QAES is open-source and is available in the new version of the EASEA platform (one can find more details about it in Section 6.1.1).

The C++-based code of QPSO algorithm used in this comparative study, was implemented following by the description in the original paper [Sun et al. (2007)]. Since the source code of QPSO is not official, it is possible that some inaccuracies in the comparative study could be found.

In this Section, all algorithms are benchmarked with their default parameter settings, presented in Table 5.4.

Table 5.4: Setting parameters of peer algorithms.

Parameter	QPSO	DMC	QAES
α	dynamic	1	1
$\sigma^{t=0}$	-	$1/d$	$1/d$
$N^{t=0}$	-	20	20
Computational budget	$(1e+4)\cdot d$	$(1e+4)\cdot d$	$(1e+4)\cdot d$

For a comparative study, we use the reference results of two different state-of-the-art algorithms: BFGS [Ros (2009)] and BIPOP-CMAES [Hansen (2009)]. The reference results of BiPOP-CMAES and BGFS are provided by the COCO platform [1] [Hansen et al. (2021)].

Test Problems: 24 single-objective noiseless functions from the Black-Box Optimization Benchmarking (BBOB) test suite of the COCO platform. For the interpretation the results, one can find the descriptions in Section 3.5. The brief explanation of the benchmarking functions (e.g., the definitions of the multi-modal functions with adequate and weak global structure (g.s.)) of the BBOB test suite are presented in Section 3.5.1.

Note, that the experimental results obtained by QAES on the problems of the Blume–Emery–Griffiths–Ising model are presented in Section 6.2.

Performance Metrics: Expected Run Time (ERT), Empirical Cumulative Distribution Functions (ECDFs) described in 3.5.1.

Simulation Settings: In order to experimentally validate the performance of QAES, we restrict our attention to $d \in \{2, 5\}$ dimensional variants. Reported results are based on 15 independent runs of 15 instances of each function. Thus, every test function was run 225 times for each dimension d. The stopping criterion is reaching target value 10^8, with a computational budget of $10^4 \cdot d$ function evaluations. These settings adhere to the standard benchmarking procedure of the BBOB workshops.

5.1.3.1 Applicability of DMC for Optimization Process

The "pure" DMC algorithm is tested only on $2d$ problems due to its inefficiency for the dimension of the search space larger than 3.

Figure 5.4 (a) shows a relative performance over all 24 functions (as explained in Section 3.5), where we observe that DMC solves about 42 % of problems. However, we notice that DMC works comparatively well in the first stage (x-$axis \in [0, 1]$), because it quickly finds the ground state of the system. From Figure 5.4 (c) and (e) it is clear that during the first stage, DMC outperforms both BFGS and BIPOP-CMAES in about 1.6 and 1.2 times respectively on low/moderate conditioned and multi-modal functions respectively. But after the first target value is found, it performs much slower than QPSO and does not achieve the last target value in many test cases. This result is predictable

[1] `https://coco.gforge.inria.fr/`

and confirms the theoretical explanation of the issues of DMC for the optimization process presented in Section 5.1.1.

5.1.3.2 Impact of Type of Probability Distribution on the Performance

In order to show the impact of the choice of probability distribution on the performance of QAES, we solve the Skew Rastrigin-Bueche test problem ($f4$) for a 5-dimensional search space.

The Skew Rastrigin-Bueche problem is a separable multi-modal function, which is selected for this experiment in order to confirm the efficiency of the heavy tails of Cauchy distribution for solving separable multi-modal functions. The results are obtained over 30 runs for each distribution (Gaussian and Cauchy).

In Table 5.5, we compare the mean Expected Run Time (ERT) with dispersion (in brackets) obtained for the different Δf with Cauchy and Gaussian distributions.

The result is predictable: with a Cauchy distribution, QAES obtains the highest accuracy ($\Delta f = 1e-7$), because of the reasons explained in Section 5.1.2. However, the positive impact of the Cauchy distribution is less pronounced on non-separable functions, where an effect of correlation among decision variables makes the uni-variate Cauchy distribution inefficient.

Table 5.5: QAES runtime with different probability distributions on $f4$ test function.

Δf	$1e+1$	$1e+0$	$1e-1$	$1e-2$	$1e-3$	$1e-5$	$1e-7$
Cauchy	**2.3**$_{(0.5)}$	**76**$_{(10)}$	**75**$_{(52)}$	**76**$_{(40)}$	**76**$_{(120)}$	**76**$_{(60)}$	**77**$_{(23)}$
Gauss	2.4$_{(2)}$	179$_{(30)}$	124$_{(321)}$	∞	∞	∞	∞

5.1.3.3 Scalability and Robustness

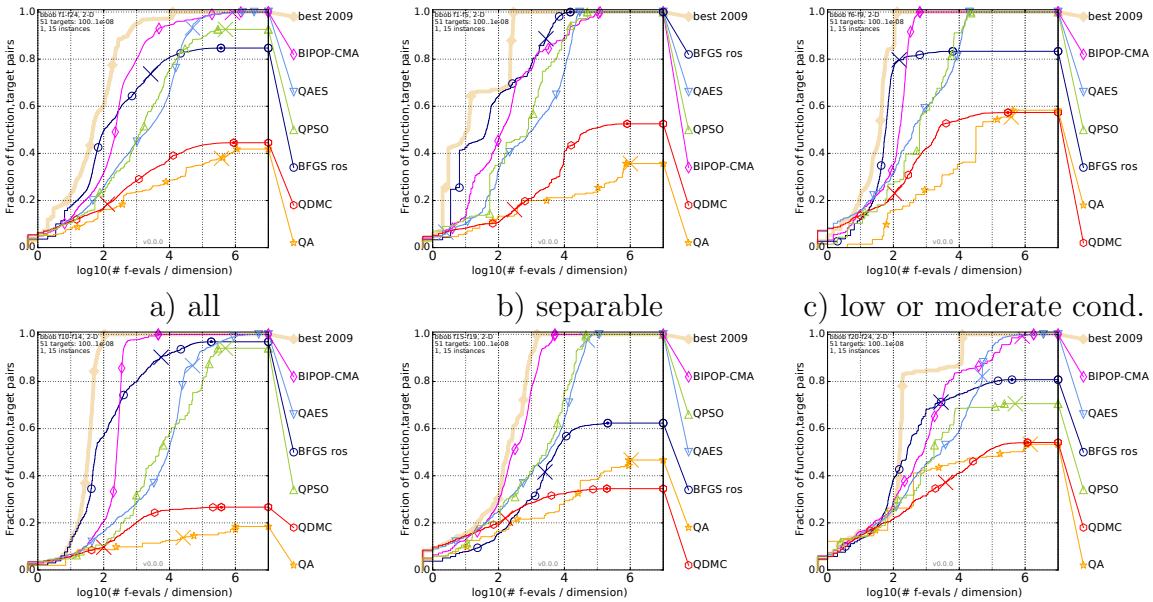
In this section, we investigate the potential scalability of QAES w.r.t. the dimensions of the search space and its robustness w.r.t. different properties of optimization problems.

For this purpose, we benchmark QAES on different problems for 2– and 5– dimensional search spaces: separable, uni-modal, multi-modal, etc.

Performance for 2-dimensional search space problems:

First, we compare the results of QAES and DMC. Figure 5.4 shows that including (1+1)-Evolution Strategy with the control of step size significantly improves the performance of DMC. Aggregated runtime results for the all 24 functions in Figure 5.4 (a) shows that QAES solves 100 % of the problems with the highest accuracy, whereas DMC solves only 42 %. Note that QAES demonstrates the same speedup at the first stage (*x-axis* $\in [0, 1]$) as "pure" DMC, because it inherits the property to efficiently find the ground state from DMC.

Second, we compare the performance of QAES and QPSO. As seen from Figure 5.4 (a) QAES solves 100 % of the 24 problems and slightly outperforms QPSO, which solves 90 %. From Figure 5.4 (f), it is clear that QAES is more efficient on multi-modal functions with weak global structure. It is explained by increasing the number of walkers N^t, which is technically the same as the use of a large population size. On the rest of the functions, both algorithms work identically (see Figure 5.4 (b-e)).



a) all b) separable c) low or moderate cond.

d) high cond./uni-modal e) multi-modal adeq. g.s. f) multi-modal weak g.s.

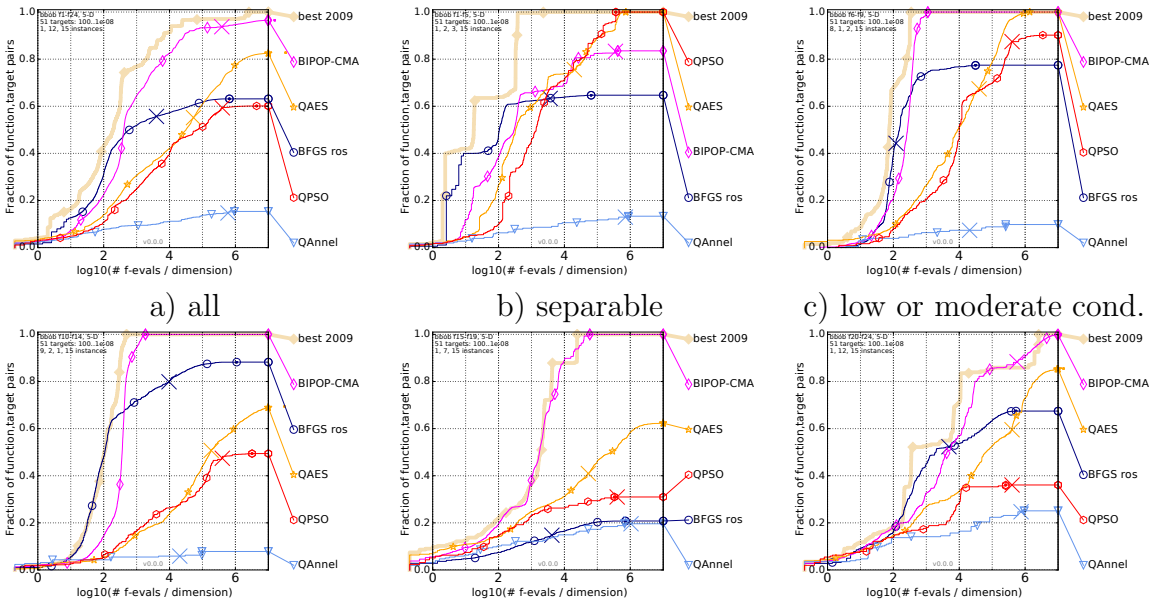
Figure 5.4: Empirical cumulative distribution of runtimes, summarized by function groups on 2-dimensional search space.

Finally, we notice from Figure 5.4 (a-f) that QAES is slower than BFGS and BIPOP-CMA in the middle stage (x-$axis \in [1, 4]$). However, as seen in Figure 5.4 (c, e, f), it solves a larger number of functions than BFGS on multi-modal and low and moderate conditional groups of problems.

Performance for 5-dimensional search space problems:

First, we compare the results of quantum-inspired algorithms: QAES and QPSO. From Figures 5.5 (a), which depicts the global results for the all 24 functions, we can conclude that QAES solves in 1.4 times larger number of problems than QPSO, by using the same number of function evaluations (so called computational budget). On separable functions (Figures 5.5 (b)) both algorithms perform similarly. However, QAES exhibits a slightly better speedup in the middle stage (x-$axis \in [1, 4]$). The significant difference in their performance can be observed on multi-modal functions (see Figures 5.5 (e, f)). It is explained by the fact that QAES does not need a large fixed population size during all the evolution process. QAES adjusts the number of particles on each iteration: after finding the ground state, it uses the minimum number of particles, which is enough for finding the global optimum thanks to the (1+1)-ES algorithm.

Second, we compare the results of QAES, BFGS and BIPOP-CMA-ES. According to Figure 5.5 (a), QAES is the second best method, which solves 82 % of problems, after BIPOP-CMAES (98 %) and followed by BFGS (60 %). However, QAES solves 100 % of separable functions and outperforms BIPOP-CMAES, which solves 82 % of problems (see Figure 5.5 (b)), where QAES shows a slightly slower speedup than BIPOP-CMAES in the the middle stage (x-$axis \in [1, 4]$). However, the speed of QAES becomes worse on uni-modal and multi-modal funcions with adequate global structure (see Figure 5.5 (d, e)): in the middle stage (x-$axis$= 4), QAES is slower than BIPOP-CMEES in about 4.3 times on uni-modal functions and in 2.8 times on multi-modal functions with adequate



a) all b) separable c) low or moderate cond.

d) high cond./uni-modal e) multi-modal adeq. g.s. f) multi-modal weak g.s.

Figure 5.5: Empirical cumulative distribution of runtimes, summarized by function groups on 5-dimensional search space.

global structure. The same as on 2-dimensional problems, in general, QAES is slower in the middle stage comparatively to BFGS and BIPOP-CMAES. This can be explained by the inefficiency of the simple rule for step size control in ES. But, QAES ensures a stable trade-off between exploration and exploitation on functions with low or moderate conditioning and multi-modal functions with weak global structure for a small dimensions of the search space (see Figure 5.5 (c, f)).

In summary, QAES optperforms QPSO by the overall performance in terms of speedup and the number of solved functions (Figure 5.5 (a)). QAES is efficient both in terms of the number of solved problems and speedup for solving different separable problems (Figure 5.5 (b)). Comparatively to QPSO, QAES performs better on multi-modal functions (Figure 5.5 (e, f)).

5.1.3.4 Benchmarking on Partially Separable Real World Problems

In this section, for studying the performance of QAES on partially separable middle-scaled problems, QAES is applied to real world optimization problem of harmonic analysis, because it is computationally cheaper comparatively to the problems of the Hamiltonian model of Magneto Caloric Materials.

In most applications of harmonic analysis, there is a great interest in detecting peaks of very close in frequency. The famous Fast Fourier Transform (FFT) method used in the harmonic analysis of signals, requires very large sample size in order to be able to detect peaks very close of frequency.

The problem presented below, is a part of the PhD thesis of Ulviya Abdulkarimova (CSTB team), on which QAES has been experimented. Here, we show some results obtained by Ulviya Abdulkarimova on the problem of harmonic analysis with QAES.

Formulation of the Optimization Problem:

Aim of the optimization: finding the parameters of sines composing the target signal presented below.

Target signal: is generated in the following form:

$$\mathrm{y}[n] = \sum_{k=1}^{K} A_k e^{-\lambda n} \cdot \sin(\omega_k \cdot n + \phi_k) + \epsilon \tag{5.2}$$

where n is the sample number, A is the amplitude, ω is the angular frequency, ϕ is the phase, K is the number of sines and ϵ is some added white noise.

In fact, we suppose that the data is not perfect: i.e., the signal is damped and the measurements are corrupted by an unwanted noise (ϵ). In this section, we test the algorithm on noisy data with a noise level of 100, meaning that the signal/noise ratio is $\approx$2/1.

Solution representation: QAES encodes the parameters of $K = 3$ sinus into a vector of double precision floating point values semantically grouped by 4:
$\{e_1, a_1, f_1, p_1, ..., e_k, a_k, f_k, p_k\}$, where e, a, f, p are respectively the damping coefficient, amplitude, frequency and phase of each of the k sines.

Since for each sine we have 4 variable: damping, amplitude, frequency and phase, thus, for 3 sines, the solutions of the given problem are formulated on 12-dimension search space: $\boldsymbol{x} \in \mathbb{R}^{\mathrm{d}}$, $d = 12$.

Solutions initialization: initializaton is randomly performed by using values for each of the sines within `[MIN_AMP, MAX_AMP], [MIN_FREQ, MAX_FREQ], [MIN_PHASE, MAX_PHASE], [MIN_EXP, MAX_EXP]` boundary intervals that contain suspected (known) ranges for the simulated data.

```
double fMIN_AMP = 200.0;
double fMAX_AMP = 120000.0;
double fMIN_FREQ = 0.26;
double fMAX_FREQ = 0.27;
double fMIN_PH = 0.0;
double fMAX_PH = 6.283185308;
double fMIN_EXP = 7.0;
double fMAX_EXP = 11.5;
```

Evaluation function:

$$f = \sum_{i=1}^{n} \left(\frac{|y(\mathbf{x}_i) - s_i|^2}{n} \right)^{1/2} \tag{5.3}$$

where $y(\mathbf{x}_i)$ is the predicted output, i is the current index of sample, n is the number of samples and s_i is the expected output.

In this work, non-uniform sampling is used with only 1024 points taken from an interval of size 8192.

The predicted output $y(\mathbf{x})$ is calculated by Equation 5.2: i.e., as a sinusoidal signal using 3 sines ($K = 3$) with very close frequencies, which is noisy and exponentially damped:
$y[n] = \exp(x_1 \cdot n) \cdot x_2 \cdot sin(x_3 \cdot n + x_4) + \exp(x_5 \cdot n) \cdot x_6 \cdot sin(x_7 \cdot n + x_8) + \exp(x_9 \cdot n) \cdot x_{10} \cdot sin(x_{11} \cdot + x_{12}) + \epsilon$

As seen from Equation 5.3, we optimize the square of the difference between the value of target signal (y) calculated with the current set of decision variables and the known *a priori* reference value s.

Precision: since we have to find the vector of decision variables ($\mathbf{x}$), where the output value of signal (y) corresponds to the expected value (s), the required precision for each variable has to be defined. For amplitudes, we are interested in error less than 10^{-2} and for frequencies less than 10^{-6}. We don't pay attention at the moment to phase and damping, as no targets were defined.

Results:

The results of this experiment are statistics over 30 runs and illustrated in Figure 5.6, where the *y-axis* is in $\log_{10}$ scale.

Table 5.6: Decision variables of the optimal solution.

Sine	Parameter	Decision variable x	True value
sine 1	e_1	x_1	-0.00000115187
	a_1	x_2	1150.13000000
	f_1	x_3	0.26883598600
	p_1	x_4	1.41284880000
sine 2	e_2	x_5	-0.00000104313
	a_2	x_5	68139.0882400
	f_2	x_7	0.26883656847
	p_2	x_8	1.41165090000
sine 3	e_3	x_9	-0.00000119361
	a_3	x_{10}	6575.936409000
	f_3	x_{11}	0.26883782723
	p_3	x_{12}	1.40905940000

The values of decision variables of the optimal solution are presented in Table 5.6. As seen from the violin plots on Figure 5.6, all obtained values are acceptable: i.e., the amplitude and frequency errors are within the required range. We can see that for the 3rd sine, the errors are larger but they are acceptable within the requirements.

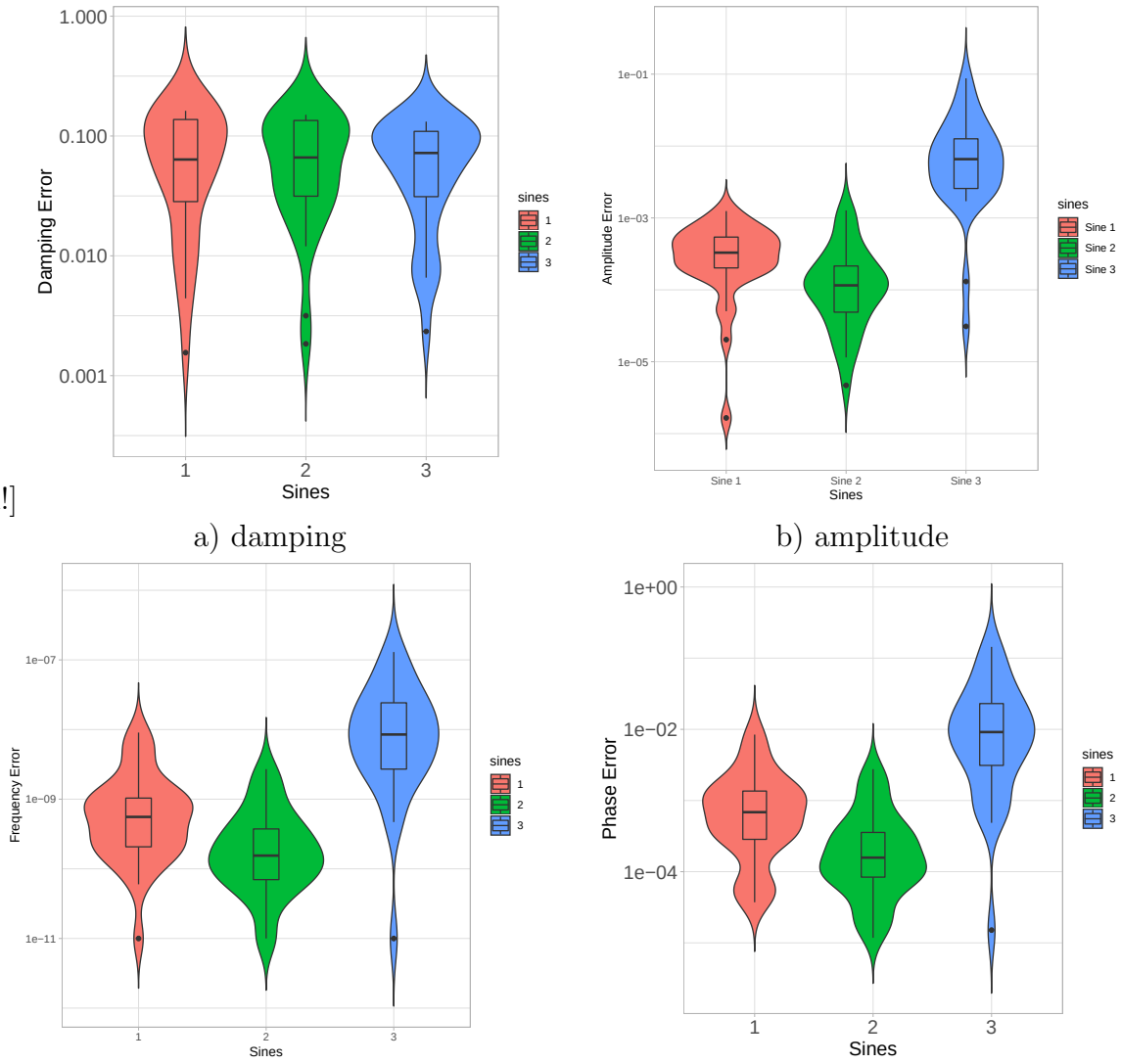
What is interesting in the QAES algorithm is the execution time. As we can see from the violin plot in Figure 5.7, it takes less than 80 seconds for QAES to get the results described above, whereas a standard $(\mu + \lambda)$-ES algorithm from the EASEA platform takes about 2 days to obtain similar results. This is a tremendous speed up. Moreover, QAES takes only 1024 points from an interval of size 8192, whereas with a standard $(\mu + \lambda)$-ES, 2048 consecutive points are needed to obtain approximately the same results.

The weak point of QAES is currently the stability of results which mainly depends on the parameters of the algorithm and the required small-scaled dimension of the search space, which limits the number of sines to be found in a given signal. However, work will continue on this in the near future.

5.1.4 Summary and Discussions

We have proposed a quantum-inspired single-objective algorithm called QAES, in order to solve computationally intensive separable/partially separable problems, aiming at solv-

[h!]



a) damping b) amplitude

c) frequency d) phase

Figure 5.6: QAES: Relative errors on 3 sines function.

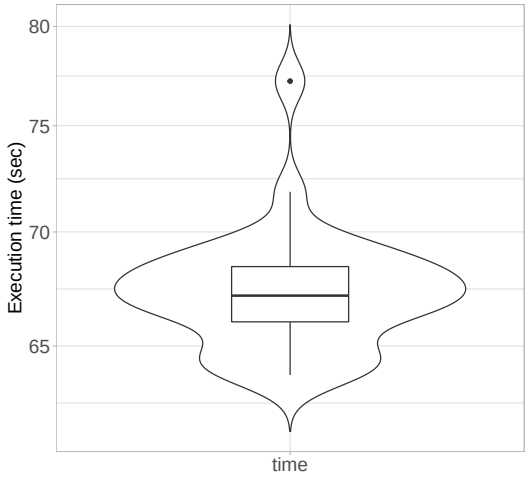


Figure 5.7: QAES: Execution time (in seconds).

ing the optimization problems of the generalized Blume–Emery–Griffiths–Ising (BIG-I) Hamiltonian model of Magneto Caloric Materials (MCMs). In principle, QAES can be used for many other separable problems, where the dimension of the search space is smaller than 12.

Contribution:

The main contributions presented in this section, regard as follows:

1. application of the Diffusion quantum Monte Carlo (DMC) algorithm to solve continious optimization problems;

2. analysis of its issues as an optimization algorithm;

3. improvement of the detected issues by integrating the (1+1)-Evolution Strategy (ES) in the structure of the DMC to create the QAES algorithm.

The key ingredient of the proposed algorithm is the Diffusion quantum Monte Carlo (DMC) method, which solves the Schrödinger equation for finding the ground state wave function and ground state energy of any quantum system.

The main feature of the proposed algorithm is the ability to adjust a current number of candidate solutions at each generation depending on a problem landscape, due to the control mechanism of the fluctuation of the population size. It can help to reduce the overall computational budget and to solve the multi-modal function more efficiently in comparison to QPSO.

Furthermore, DMC does not depend on the defined population size value, which is a critical tuning parameter for many optimization algorithms, e.g., QPSO. Because a number of candidate solutions is not fixed and fluctuates from generation to generation, the population size is not a tuning parameter any more and managed "automatically".

Our observations in this study can be summarized as follows:

- DMC is applicable for continuous optimization;

- DMC can find the ground state of a given quantum system, but cannot find the global minimum of potential energy (objective function);
- ES helps to improve the accuracy and convergence rate of DMC, due to an application of a simple step adaptation and greedy selection;
- QAES (a combination of DMC and ES) has lower computational complexity comparatively with DMC, as it does not need to compute the ground state wave function, since its exact value is not required for optimization;
- QAES solves 100% of problems for 2-dimensional search space, whereas DMC without ES solves only 40%;
- QAES outperforms QPSO in accuracy by solving in 1.4 times larger number of problems on 5-dimensional search space;
- QAES is very efficient on separable functions: it solves 100% of the problems in 2– and 5–dimensional search spaces;
- As QPSO, QAES is slow on uni-modal functions;
- As QPSO, QAES is not invariant under rotations of the search space.

Limitations: The main limitation of QAES is a lack of scalability w.r.t. the dimensions of the search space. Comparing the results obtained on problems with 2– and 5–dimensional search spaces, we can conclude that QAES experiences the curse of dimensionalality and cannot perform efficiently on problems with high dimensional search spaces. However, it is not critical for solving the problems of the Hamiltonian model of Magneto Caloric Materials (MCMs), due to their small-scaled dimension size.

Discussion: As most population-based algorithms, QAES constructs an initial population of random candidate solutions (particles): i.e., the initial wave function is sampled from a uniform random distribution.

However, following the importance sampling concept, an informed initialization of the population based on *a priori* knowledge, can be more efficient than naive and random initialization.

As explained, we do not use importance sampling in this work, in order to avoid additional computational complexity. However, it can be useful to integrate a mechanism to allow the proposed algorithm to be reinforced with an "informed" initialization scheme of the inital wave function. We suppose that such kind of reinforced technique can be based a simulated annealing algorithm.

QAES works well on separable functions, however its results deteriorate when the degree of non-separability of a given problem increases: i.e., when the decision variables are correlated. Mainly, this less pronounced performance on non-separable functions can be explained by an application of uni-variant Cauchy distribution in the diffusion displacement step, which is heavily coordinate-dependent.

It can be interesting to integrate into DMC a technique, which has an independence from the coordinate system, e.g. Covariance Matrix Adaptation Evolution Strategy (CMA-ES), and consequently, can achieve better performance on problems with different properties.

Perspectives: We assume four short term perspectives:

1. Even though the problems of the Blume–Emery–Griffiths–Ising Hamiltonian model are supposed to be small-scaled, a better scalability w.r.t. the dimension of search space has to be ensured.

2. An investigation of the impact of tuning-parameters on performance is required.

3. Further improvements must be done for better solving non-separable problems, where a correlation among the decision variables exists.

4. A mechanism, like importance sampling could be added into the structure of QAES, in order to provide an optimal sampling design.

In a long term perspective, we suppose that the proposed algorithm can be transformed into a hybrid approach, when the DMC part will be implemented on a real quantum computer, such as Noisy Intermediate-Scale Quantum (NISQ) devices, in order to find the ground state wave function.

Connection with the Research Problems: QAES is developed mainly to solve the separable/partially separable single objective problems of the Hamiltonian model of Magneto Caloric Materials presented in Section 2.2.2).

The experimental results obtained by QAES on the problems of the Hamiltonian model are presented in Section 6.2.

5.2 Fusion-based Unified Optimization Algorithm

In this section, we propose an optimization algorithm, which provides the scalability w.r.t. the dimensions of the objective space and search space. We call such algorithms "unified" for short (borrowing this term form [Seada & Deb (2014)]), which aims at unifying mono-, multi- and many-objective optimization.

Motivation:

The attempts to develop a unified algorithm have been already done and presented in [Deb & Tiwari (2008)] and [Seada & Deb (2014)], where the importance of such kind algorithms for solving real-world optimization problems is explained by the diversity of forms and types of these problems. More specifically, if many different problems can be formulated in the frame of one research study, the user first analyzes each problem and chooses a suitable algorithm for solving it. This is because an algorithm efficient for finding the global optimum of a single-objective problem, cannot be adequately applied to find multiple optimal solutions present in another optimization problem [Deb & Tiwari (2008)]. Obviously, to solve different kinds of problems, the user needs to know different algorithms, each specialized in solving a particular class of optimization problem. Next, the user has to find and to set the tuning parameters of each optimiaztion algorithm. Moreover, every time the problem is reformulated, the user must merge the codes of the modified problem and the new algorithm once again. Consequently, conducting a such kind of research is both very difficult and time-consuming. In fact, these reasons can seriously restrict the research, especially if the optimization problem is very time-consuming by itself: if an unsuitable algorithm is selected, a computationally expensive experiment can be useless.

In the frame of our work, the motivation for developing a unified algorithm comes from the following requirements:

1. both models: i.e., the model of dual-mode operating Active Magnetic Regenerator (AMR) and the Hamiltonian model of Magneto Caloric Materials (MCMs) can be used for different study cases. It means that the number of objectives and the number of decision variables can be different, e.g.:
   (a) in case of the simulation of physical properties of MCMs, the number of objectives depends on the number of studied physical properties of materials and in theory, it can vary from 1 to 2-3 objectives;
   (b) in case of the design investigation of the AMR, the number of decision variables is variable according to the number of control and design parameters taken into consideration.
2. both models are under active development, and as a consequence, their optimization problems can be changed according to modifications of the model.

Thus, it seems to be useful to have an algorithm which is capable to handle any number of objectives and variables in order to solve the different problems of the different modifications of the models. Developing this intention further, such kind of algorithm can be useful for simplifying the structure of the overall system, where the AMR model will employ the Hamiltonian model for the simulation physical of properties of the needed materials. In this case, the idea to develop a unified algorithm becomes topical.

Related works:

The related works concerning unified optimization are discussed in Section 3.2.2. Remembering about No Free Lunch Theorem [Wolpert & Macready (1997)], there are only two successful attempts to develop an algorithm capable to solve single-, multi- and many-objective problems:

1. the unified algorithm presented in [Deb & Tiwari (2008)], is expected to solve multi-objective optimization problems in a manner similar to Non-dominated Sorting Genetic Algorithm II (NSGA-II) with some improvements for scaling it down, which can be attributed to restricted selection and a more disruptive mutation operator. But, this algorithm is not suitable for our application case, because it is not scalable for many-objective problems;

2. the unified algorithm proposed in [Seada & Deb (2014)], employs Non-dominated Sorting Genetic Algorithm III (NSGA-III) as a baseline technique and scales it down to solve single- and many-objective problems by increasing the population size and increasing the selection pressure. In general, this algorithm seems suitable for our application case. However, an increase of the population size is an issue, when a given problem is computationally expensive, and an increased selection pressure can lead to premature convergence on multi-modal functions.

Addressing the same idea about a unified algorithm and trying to develop a unified algorithm suitable for our application case, we employ a fusion method described in Section 3.2.2, which can be used to scale down a many-objective algorithm to solve single-objective problems by fusing the solutions from several algorithms with different properties [Ibrahim, Martin, Rahnamayan & Deb (2017)]. We explain this idea in details in Section 5.2.1.

Contribution:

In this regard, a key contribution is an application of fusion method for ensuring the scalability w.r.t. the number of objectives and decision variables. We hope that the proposed unification will be useful for researchers from the following points of view:

1. a new insight about fusion methods capable to be used to constitute scalable algorithms;

2. a new study, where a quantum-inspired algorithm is employed;

3. a new algorithm, which allows users to simplify a real-world research study by working with a single software for solving the optimization problems in different dimensions of the objective and search spaces.

This chapter is organized as follows. In Section 5.2.1 the proposed algorithm is presented. Section 5.2.2 reports its experimental validation. Finally, Section 5.2.3 concludes this topic with a discussion and some perspectives for further research.

5.2.1 Proposed Algorithm

Following the explained above motivation, in this section, we further develop the concept of unified algorithm presented in [Seada & Deb (2014)], by providing a new algorithm,

which employs the fusion of solutions from different algorithms [Ibrahim, Martin, Rahnamayan & Deb (2017)] in order to ensure the scalability w.r.t. the number of objectives and the number of decision variables.

In this section, we largely use the information provided in Section 3.2.2, Section 3.4.5 and Section 3.4.4.

Hypothesis:

The main idea is to endow an efficient many-objective optimization algorithm with a property to solve different single-objective problems by applying the fusion method [Ibrahim, Martin, Rahnamayan & Deb (2017)] described in Section 3.2.2, in order to employ this algorithm for solving the different problems of both models.

Non-dominated Sorting Genetic Algorithm III (NSGA-III) [Deb & Jain (2013)] presented in Section 3.4.5, seems to be the best candidate as a baseline algorithm for this purpose, because it has a small number of tuning parameters and it has an ability to solve efficiently different many-objective problems. Moreover, it was used for the unification algorithm presented in [Seada & Deb (2014)].

However, NSGA-III is not efficient for solving single-objective problems, because of the reasons, which are discussed in [Seada & Deb (2014)] and presented in Section 3.4.5. Below, we briefly remind two main issues:

1. the recommended population size for single objective optimization equals $N = 4$ (defined as the smallest multiple of 4 greater than the number of reference directions, where only one reference direction is considered in case of single objective optimization), which is too small for NSGA-III's recombination operator to find useful child solutions [Seada & Deb (2014)];

2. the solutions are picked randomly for recombination/mutation operators [Seada & Deb (2014)].

The population size of NSGA-III is analysed in Section 3.4.5, where we discuss its impact on the performance of the algorithm. According to this discussion, we can make the deductions presented below.

On one side, to use a small population size seems to be important because of two following reasons:

1. we do not assume that NSGA-III has been designed for a parallel implementation, since NSGA-III cannot ensure good scalability w.r.t. the population size (see experimental results in Section 4.1.2.1). Thus, for a sequential (non-parallel) implementation of this algorithm, total number of calls of evaluation function has to be small, especially if the optimization problem is computationally intensive. In this case it could be desirable to be able to handle a small population size and to find the good solutions in small number of generations;

2. with a large population size, without selection pressure, NSGA-III will work as a random walk algorithm on single-objective problems, and consequently, will require a larger number of calls of evaluation functions [Seada & Deb (2014)].

On the other side, the population, which consists of only four candidate solutions (so called individuals), is too small in order to find the global optimal, especially for multi-modal problems [Seada & Deb (2014)].

For scaling NSGA-III down, we propose to use the fusion method presented in [Ibrahim, Martin, Rahnamayan & Deb (2017)] and discussed in Section 3.2.2, instead of increasing the population size and adding the selection pressure in the structure of NSGA-III, as it was proposed in [Seada & Deb (2014)]. Inspired by particle movement in a quantum system, we propose to fuse the solutions of NSGA-III with the solutions of Quantum-inspired Particle Swarm Optimization algorithm (QPSO) [Sun et al. (2007)], which is described in Section 3.4.4, in order to maintain and control diversity in the default population size of NSGA-III, which consists of only four candidate solutions. QPSO is selected for this purpose, because of the following reasons:

1. it is theoretically guaranteed that QPSO converges to the global optimum (this statement is proven in Section 3.4.4);

2. QPSO is a well-known state-of-the-art algorithm, which has been successfully applied to a vast variety of engineering problems mentioned in Section 3.4.4;

3. in this thesis, QPSO has been largely tested and shown a good performance on different problems of the Black-Box Optimization Benchmarking (BBOB) test suite *versus* the proposed quantum-inspired algorithm (QAES).

.

The proposed fusion-based unified optimization algorithm called QIU-NSA, works with default values of hyper-parameters of NSGA-III and QPSO and it does not require any additional tuning parameters. In QIU-NSA, QPSO helps to improve the performance of NSGA-III on single-objective problems without efficiency loss on many-/multi-objective functions.

We believe that presented algorithm not only allows to solve the different problems of both simulation models, but also proposes an application of quantum-inspired algorithms in fusion-based method, which can be beneficial for further research.

5.2.1.1 QPSO: Modified Version

Following the hypothesis presented above, in this work, we employ the Quantum-behaved Particle Swarm Optimization Algorithm (QPSO) algorithm as an auxiliary part in the frame of the proposed fusion-based algorithm. After a theoretical investigation of different versions of QPSO in Section 3.4.4, we selected the version presented in [Sun et al. (2007)], which employs the standard model of Delta potential well. A more detailed explanation and description of the QPSO algorithm is presented in Section 3.4.4. Here, we briefly remind only the most important analogies used in QPSO: a particle position is a solution candidate, the position coordinates of a particle is the decision variables of a solution candidate, a performance index of each particle position is an objective function, the best global position is the best solution from all solutions in the swarm.

Important limitations:

As discussed in Section 3.4.4, QPSO has several limitations. Since we are interested in its application as a mechanism to maintain the diversity of the population in NSGA-III, we remind the limitations below, which are the most critical in our case and require some modifications in the QPSO algorithm presented in Section 3.4.4:

1. a relatively large population size can be required for obtaining the global solution: i.e., the minimal recommended size in the original papers is $N = 20$ ([Sun et al. (2007)]), which is a critical condition, since we assume that both algorithms, QPSO and NSGA-III, work with the same small population size ($N = 4$) in the proposed fusion-based algorithm;

2. an impact of the contraction expansion parameter (α) on the performance, which has to be somehow adjusted.

Preliminary research:

In fact, both limitations are connected with a lack of control of population diversity. In order to make it clear, let us analyse the second limitation. According to quantum physics, the width of the Delta potential well ($\mathbf{L}$), which determines the search space of each particle at each generation (see Equation 3.31 in Section 3.4.4), goes to zero during the optimization process, where the ground state has to be found. From Equation 3.31 it is seen that this width depends on the value of α and the value of $|\mathbf{p}_{mean} - \mathbf{r}|$: i.e., the deference between the mean coordinates of the personal best positions of all particles and the coordinates of the current particle. Thus, if the population is a small, the value of $|\mathbf{p}_{mean} - \mathbf{r}|$ will be prematurely around zero, and consequently, the width of the Delta potential well will be prematurely too narrow.

According to [Sun, Feng & Xu (2004)], the α parameter can help to improve this issue: it is determined as the most important algorithmic parameter, which aims at controlling the convergence behaviour and the diversity adjustment. This topic is still open and studied in more recent researches: [Sun et al. (2012)], [Rehman, Yang, Zhou, Yang & Khan (2017)], [Rehman, Yang & Khan (2017)]and [Tu et al. (2020)]. Below, we briefly discuss them:

1. In [Sun et al. (2012)], the authors try to find experimentally the optimal static value of α, which can be employed during all the optimization process for different functions. The proposed optimal value is $\alpha = 0.75$, especially for uni-modal problems. The advantage of this method is that the value of α does not depend on the other hyper parameters, like the total number of generations.

2. In [Tu et al. (2020)] and [Rehman, Yang & Khan (2017)], the static value of α is criticized, where the authors assume that without proper adjustment of the value of α, QPSO gets stuck into local optima, especially on multi-modal problems. They propose to change the value of α parameter on the different stages of the optimization process. Following this idea, several methods for dynamically changing α parameter are proposed:

   (a) The most common method in the literature originally presented in [Sun, Feng & Xu (2004)], is to linearly decrease α parameter from 1.0 to 0.5, as presented in Equation 3.32 (see Section 3.4.4). The range $[0.5, 1.0]$ is defined experimentally and explained in details in more recent work [Sun et al. (2012)]. From Equation 3.32 it is clear that the value of α is initially close to 1, because the particle is far away from the mean best position and the whole search space has to be investigated. At the end of the optimization process, near to the ground state, the particle is supposed to be close to the mean best, and thus, the

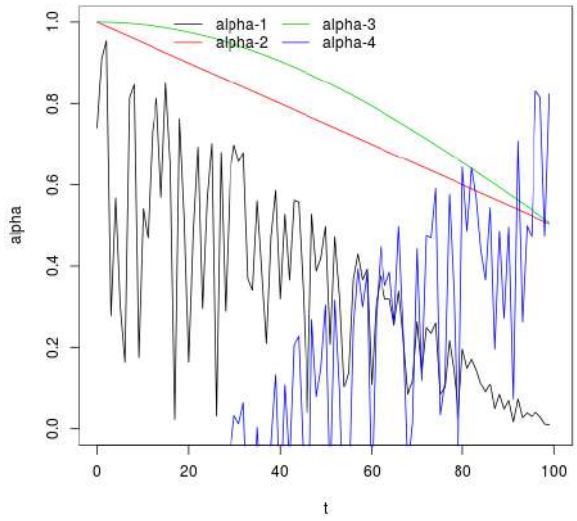


Figure 5.8: Variation of α parameter depending on the current number of generations t.

search space has to be small enough for precising the best position. From the physics point of view, this method seems very logical. The red line ("alpha-2") in Figure 5.8 presents the variation of α parameter by this method. However, it introduces a dependency of the value of α from the other hyper parameter: i.e., the total number of generations defined by the user.

(b) The method proposed in [Tian et al. (2011)], adjusts the value of α according to the cosine function of an argument, which is defined as a relation between the number of current generation and the total number of generations. The proposed variation of α parameter is depicted in Figure 5.8 by the green curve ("alpha-3"). As we can see from Figure 5.8, the cosine function decreases slower than the linear function ("alpha-2"). It can maintain the population diversity better on some problems, but makes the convergence rate slower, especially on uni-modal problems. The same as the linearly decreasing method, the cosine decreasing method depends on the total number of generations defined by the user, and the performance generally depends on the selection of this tuning parameter.

(c) The methods proposed in [Rehman, Yang, Zhou, Yang & Khan (2017)], [Rehman, Yang & Khan (2017)] and [Tu et al. (2020)], also use the value of the total number of generations in their equations of α, but they employ a sine function of an argument, which includes a random uniformly distributed number. In Figure 5.8, the "alpha-1" and "alpha-4" curves present the variation of α parameter according to [Rehman, Yang, Zhou, Yang & Khan (2017)] and [Tu et al. (2020)] respectively.

By analysing these curves, we think that these methods are inexact in the physical meaning because of the following reasons:

i. The black curve ("alpha-1") and the blue curve ("alpha-4") are located much lower than the optimal range of α defined and proved in [Sun, Feng

& Xu (2004)], [Sun et al. (2012)], and consequently, they provide values of α distributed in an inefficient area.

ii. In both cases ("alpha-1" and "alpha-4" in Figure 5.8), the value of α is significantly changed on the same stage of the optimization process because of the random value integrated as a sine argument in the equation of α. It means that the width of the potential well is periodically increased and decreased, which is against the physical meaning and has not any reasonable logic behind. Theoretically, such techniques could help preserving diversity. But the range of randomness is very big.

iii. In the case of the method proposed in [Tu et al. (2020)] (the blue curve), the value of α has a global tendency to increase, which increases the width of the potential well during the optimization process and theoretically, it goes against the physical meaning, since it draws back the particle far from the ground state and makes the optimization process inefficient.

Preliminary experiment: experimental validation of the theoretical analysis

Test objective: to benchmark the QPSO algorithm with the presented above methods and experimentally confirm our analysis.

Test problems: the 24 single-objective noiseless Black-Box Optimization Benchmarking (BBOB) [Hansen et al. (2009)], [Hansen et al. (2012)] test suite of the COmparing Continuous Optimizers COCO platform [Hansen et al. (2021)]. The description of the BBOB test suites with an explanation of the results interpretation are provided in Section 3.5. The expressions and bounds of the test functions are provided in details in [Hansen et al. (2009)].

Performance metrics: the empirical cumulative distribution of runtime [Hansen et al. (2012)], i.e., the runtime in number of function evaluations divided by dimension, on all noiseless functions of BBOB-2009 [Hansen et al. (2009)].

Algorithm: since the official code of QPSO has not been published yet, we implemented a C++-based code according to the version of algorithm described in Section 3.4.4 and originally presented in [Sun et al. (2007)].

Parameters: For benchmarking all methods, we use the following parameters:

1. population size: $N = 4$, since it is our study case in the frame of the proposed hypothesis;
2. number of decision variables $d = 20$;
3. total number of generations $T_{max} = 300000$.

In Table 5.7, we present the equations of α and notations used in this experiment.

Simulation settings:

1. The reported results are based on 15 independent runs of 15 instances for each function. Thus each benchmark function is run 225 times that seems to be sufficient for statistically significant results.
2. The stopping criterion is the number of generations.

Table 5.7: Adjustment of α used in this experiment.

Notation	Equation	Paper
alpha-0.7	$\alpha = 0.75$	[Sun et al. (2012)]
alpha-1	$\alpha = (T_{max} - t) \cdot sin(\mathcal{U}[0,1])/T_{max}$	[Rehman, Yang & Khan (2017)]
alpha-2	$\alpha = 0.5 + 0.5 \cdot (T_{max} - t)/T_{max})$	[Sun, Feng & Xu (2004)]
alpha-3	$\alpha = 0.5 \cdot cos(\frac{\pi \cdot t}{2 \cdot T_{max}}) + 0.5$	[Tian et al. (2011)]

Results: Each graph in Figure 5.9 depicts the empirical cumulative distribution of runtime of the QPSO with annotated method from Table 5.7 on all 24 noiseless functions on dimension $d = 20$. As explained in Section 3.5, the best algorithm covers the largest area under its graph.

Thus, Figure 5.9 clearly shows that the cosine function based decreasing method ("alpha-3") and the linearly function based decreasing method ("alpha-2") outperform, in terms of the portions of the solved problems, the methods with the static value of α ("alpha-0.7") and the sine function based method ("alpha-2").

The transparent line marked "best 2009" in Figure 5.9, is the artificial best algorithm, which presents an aggregation of the best results observed on all functions in 20-dimensional search space on the BBOB-2009 workshop. Thus, we do not take it into account in the presented comparative study, but we show it in order better estimate the performance.

As seen from Figure 5.9, the cosine based method solves around 32% of all functions, as its blue curve achieves value 0.32 on y-axis. The linear based method solves around 28 % of all functions as its yellow curve achieves $y = 0.28$. Whereas, the method used the static value of α and the method based on the sine function solve less then 5% of all functions.

These results confirm our analysis: i.e., the method based on the sine function with a random argument changes significantly the width of the potential well on the same stage of the optmization process, which make this process very slow and inefficient (see the red curve in Figure 5.9).

The cosine based method can better ensure the diversity of the population on some functions, but does not provide any significant advantage over the linear based method on all functions with the small population size.

The presented above analysis and experiment allows us to make the following conclusions:

1. all methods (except the method with a static value of α) depend on the value of the total number of generations in their equations of α, which introduces an additional relation between two tuning parameters: i.e., α and the number of generations;

2. all methods perform similarly on the first stage of the optimization process (the value of y is the same for all curves on the x-axis $[0, 1]$ interval);

3. none of these methods can control the diversity of the population efficiently;

4. none of these methods can work efficiently with a small population size ($N = 4$).

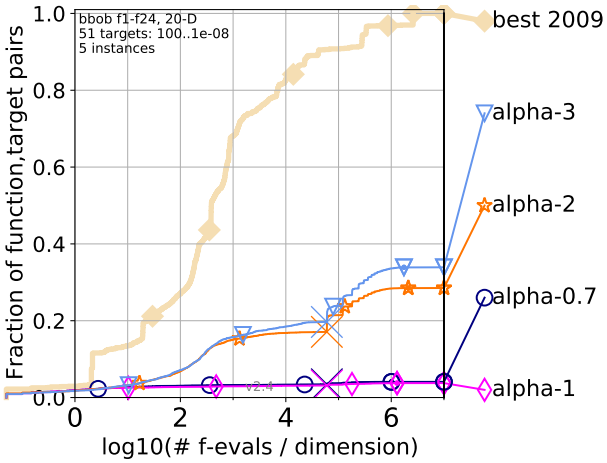


Figure 5.9: Empirical runtime distributions (runtime in number of function evaluations divided by dimension) on all noiseless functions of BBOB-2009 on $d = 20$ dimensional search space.

Proposed modifications:

In this work, we propose an alternative solution: instead of adjusting α parameter according to the values of the total number of generations, we propose to integrate the coefficient of the control of the diversity and directly change the value of the width of the potential well, without extra manipulations with the value of α.

Thus, we slightly changed the QPSO procedure in order to improve its ability to control the diversity of the population by changing the value of the width of the Delta potential well according to the value of an introduced diversity coefficient.

Comparatively to the original version of QPSO presented in Algorithm 3 in Section 3.4.4, we integrate the following modifications, which concern the process of particle creation:

1. a static value of $\alpha = 0.75$ is used for avoiding the additional dependency from the total number of generations;

2. a special coefficient c_{div} is integrated in order to control diversity. It is calculated at each generation according to Eq. 5.4:

$$c_{div}^{(t)} = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{d} |r_{i(j)}^{(t)} - p_{mean_{(j)}}^{(t)}| \tag{5.4}$$

where $r_{i(j)}^{(t)}$ is a j-th coordinate of i-th particle;
$p_{mean_{(j)}}^{(t)}$ is the j-th coordinate of the mean value of the personal best positions of all particles and can be calculated according to Equation 3.33 presented in Section 3.4.4;
N is the number of particles (the population size);
d is the search space dimension;
t is the current generation.

3. if the value of c_{div} is smaller than the threshold value c_{limit}, a small Cauchy distributed random noise is introduced in each coordinate of particle. More specifically, the c_{div} value replaces the value of $\mathbf{L}_i^{(t)}$ (i.e., the width of the Delta potential well in which all particles move), which determines a search space of each particle at each generation and is calculated by Equation 3.31 (line 15 in Algorithm 11).

   Uni-variate Cauchy distribution has already been discussed in Section 4.2.5 and applied in the structures of FastEMO and QAES. Having a positive experience with the heavy tails of a Cauchy distribution on separable problems, we assume that it can help to maintain the diversity in a population size of size 4.

The pseudo-code of this modified procedure is given in Algorithm 11. The notations used in the Algorithm 11 are borrowed from Section 3.4.4 and for sake of traceability are summed up in Table 5.8.

Table 5.8: Notations used in Algorithm 11.

Notation	Explication	Value
t	current generation	$t \in \mathbb{N}_+$
N	population size	$t \in \mathbb{N}_+$
d	number of coordinates	$d \in \mathbb{N}_+$
$\boldsymbol{r}^t$	particle position	$\boldsymbol{r}^t \in \mathbb{R}^{\mathrm{d}}$
i	index of current particle	$i \in \mathbb{N}_+$
j	index of current coordinate	$j \in \mathbb{N}_+$
c_p, c_g	random numbers uniformly distributed	c_p, $c_g \in \mathbb{R}$
$\mathbf{p}_i^{(t)}$	local attractor for the i-th particle	$\mathbf{p}_i^{(t)} \in \mathbb{R}^{\mathrm{d}}$
$\boldsymbol{p}_{mean}^{(t)}$	mean value of the local best position	$\boldsymbol{p}_{mean}^{(t)} \in \mathbb{R}^{\mathrm{d}}$
$\mathbf{r}_{best_{(i)}}^{(t)}$	best local position of the i-th particle	$\mathbf{r}_{best_{(i)}}^{(t)} \in \mathbb{R}^{\mathrm{d}}$
$\mathbf{r}_{best_{(g)}}^{(t)}$	best global position	$\mathbf{r}_{best_{(g)}}^{(t)} \in \mathbb{R}^{\mathrm{d}}$
$L_{i,j}^{(t)}$	width of the Delta potential well	$L_{i,j}^{(t)} \in \mathbb{R}$
c_{div}	diversity coefficient	$c_{div} \in \mathbb{R}$
c_{limit}	boundary value of c_{div}	$c_{limit} \in \mathbb{R}$
α	contraction-expansion coefficient	$\alpha \in \mathbb{R}_+$
$\mathcal{C}$	Cauchy distribution	
$\mathcal{U}$	Uniform distribution	

The experimental validation of the presented modifications is provided in Section 5.2.2

Algorithm 11: Modified QPSO algorithm.

Result: returns the position vector of the global best particle

begin
  Initialize the current positions randomly
  $\alpha = 0.75$
  for $t = 1$ *to* T **do**
    Calculate $\mathbf{p}_{mean}$ by Eq. 3.33
    Calculate c_{div} using Eq. 5.4
    for $i = 1$ *to* N **do**
      Calculate fitness $f\left(\mathbf{r}_i^{(t)}\right)$
      Update $\left(\mathbf{r}_{best_{(i)}}^{(t)}\right)$ by Eq. 3.25
      Update $\left(\mathbf{r}_{best_{(g)}}^{(t)}\right)$ by Eq. 3.26
      $c_p, c_g \sim \mathcal{U}[0,1]$
      Compute the local attractor $\mathbf{p}_i^{(t)}$ by Eq. 3.28
      for $j = 1$ *to* d **do**
        $u \sim \mathcal{U}[0,1]$
        if $c_{div} < c_{limit}$ **then**
          $L_{i,j} = \mathcal{C}(0, c_{div})$
        else
          $L_{i,j} = (2 \cdot \alpha) \cdot |p_{mean_{(i,j)}}^{(t)} - r_{i,j}^{(t)}|$
        $\mathbf{r}_{i,j}^{(t+1)} = p_{i,j}^{(t)} - L_{i,j} \times ln\left(1/u\right)$ with probability 0.5
        otherwise $\mathbf{r}_{i,j}^{(t+1)} = p_{i,j}^{(t)} + L_{i,j} \times ln\left(1/u\right)$
  return $\mathbf{r}_{gbest}$

5.2.1.2 Quantum-Inspired Unified Non-dominated Sorting Algorithm: QUI-NSA

In this section, we present a unified algorithm, called QIU-NSA, which aims at being scalable w.r.t. the dimension of the objective and search spaces. Under the term scalability w.r.t. the dimensions of the objective space, we suppose an ability of the optimization algorithm to solve single-, multi- and many- objective problems with approximately the same efficiency. Following the hypotheses presented above, we employ the fusion method presented in Section 3.2.2, so that a many-objective algorithm can scale down to solve single-objective problems. Indeed, the fusion of solutions from different algorithms maintains the diversity of solutions, which seems to be helpful, especially if we have to handle a small population size.

In general, a fusion method combines the solutions from two or more algorithms such that the resulting algorithm extracts the best solutions by exploiting the different features of all integrated algorithms [Thangaraj et al. (2011)]. This method is usually applied for solving a larger number of problems, but it has never been used for unified optimization. In this work, we make an effort to apply the fusion method to combine the solutions of the many-objective algorithm NSGA-III and the single-objective algorithm QPSO described in Section 3.4.5 and in Section 5.2.1.1 respectively.

The structure of QIU-NSA is shown in Figure 5.10. The notations used for the presentation of QIU-NSA algorithm and in Figure 5.10 are summed up in Table 5.9.

Table 5.9: Notations of QIU-NSA.

Notation	Explication	Value
t	current generation	$t \in \mathbb{N}_+$
N	population size	$t \in \mathbb{N}_+$
d	number of coordinates	$d \in \mathbb{N}_+$
m	number of objectives	$m \in \mathbb{N}_+$
$\boldsymbol{lb}$	vector of lower boundaries	$\boldsymbol{lb} \in \mathbb{R}^{\mathrm{d}}$
$\boldsymbol{ub}$	vector of upper boundaries	$\boldsymbol{ub} \in \mathbb{R}^{\mathrm{d}}$
c_{limit}	threshold of diversity	$c_{limit} \in \mathbb{R}_+$
$\boldsymbol{g}^t$	global solution of QPSO	$\boldsymbol{g}^t \in \mathbb{R}^{\mathrm{d}}$
$\boldsymbol{P}^t$	parent population	$\boldsymbol{P}^t \in \boldsymbol{p}_1, \boldsymbol{p}_2, .., \boldsymbol{p}_N$
$\boldsymbol{Q}^t_{QPSO}$	child population from QPSO	$\boldsymbol{Q}^t_{QPSO} \in \boldsymbol{q}_1, \boldsymbol{q}_2, .., \boldsymbol{q}_N$
$\boldsymbol{Q}^t_{NSGA-III}$	child population from NSGA-III	$\boldsymbol{Q}^t_{NSGA-III} \in \boldsymbol{q}_1, \boldsymbol{q}_2, .., \boldsymbol{q}_N$

As seen from Figure 5.10, QIU-NSA consists of three main modules: NSGA-III, the modified QPSO version and the fusion module. Note that multi- and many-objective problems are solved by NSGA-III, i.e., the proposed fusion-based method is active only for single-objective problems. The modules of NSGA-III and QPSO perform according to pseudo-codes presented in Algorithm 4 and Algorithm 11 respectively.

The values of the tuning parameters are taken by default:

1. QPSO: contraction-expansion coefficient $\alpha = 0.75$;
2. NSGA-III:

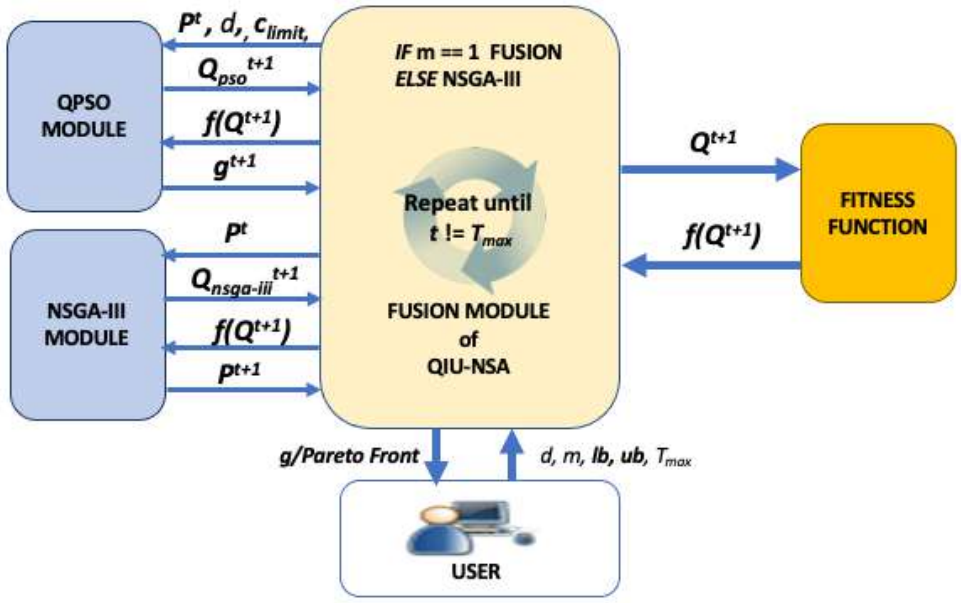


Figure 5.10: Structure of the fusion-based unified algorithm (QIU-NSA).

(a) Simulated Binary (SBX) crossover [Deb et al. (1995)]:
   i. probability of crossover operation (rate): $p_c = 1.0$;
   ii. distance parameter of crossover distribution: $\eta_m = 30$.

(b) Polynomial mutation [Deb & Deb (2014)]:
   i. probability of mutation operation (rate): $p_m = 1/d$;
   ii. distance parameter of mutation distribution: $\eta_m = 20$

Here, we do not provide the descriptions of the hyper parameters of SBX crossover and polynomial mutation, because they are well-known and do not play a key role in this work. However, we give a brief reminder of the role of parameter η for SBX crossover and polynomial mutation: a high value of η will produce children resembling to their parents, while a small value of η will produce solutions much more different.

3. Fusion: the population size (N), which is defined according to the number of reference points of the NSGA-III ($N = 4$ in single-objective case) [Seada & Deb (2014)].

At generation $t = 0$, the algorithm starts in the fusion module with the definition of the following parameters: (i) the dimension of search space (d); (ii) the dimension of target space (m); (iii) the boundaries ($\boldsymbol{lb}$, $\boldsymbol{ub}$) for each decision variable of ($\boldsymbol{x}^t$); (iv) the threshold of diversity (c_{limit}) and (v) the total number of generations (T_{max}). The diversity coefficient (c_{div}) is set to 0. The termination criterion is defined as the total number of generations (T_{max}).

In the fusion module, the initial population $\boldsymbol{P}^{t=0}$ of size N are created randomly according to the defined boundaries ($\boldsymbol{lb}$, $\boldsymbol{ub}$) and evaluated. Then, the following steps are iterated until the termination criterion is satisfied:

1. `Make Child Population:`
   The parent population $\boldsymbol{P}^t$ is sent to the NSGA-III and QPSO modules in order to produce two child populations:

   (a) in the NSGA-III module: $\boldsymbol{Q}^{t+1}_{NSGA-III}$ population is produced by the function `MakeChildPopulation` of NSGA-III from Algorithm 4;

   (b) in the QPSO module: $\boldsymbol{Q}^{t+1}_{QPSO}$ population is produced by the QPSO particle creation procedure from Algorithm 11.

   Thus, each module, NSGA-III and QPSO, creates new sets of solutions $\boldsymbol{Q}^{t+1}_{NSGA-III}$ and $\boldsymbol{Q}^{t+1}_{QPSO}$ according to their original rules and returns them into the fusion module.

2. `Uniform-based Random Selection:`
   Then, in the fusion module, the received populations ($\boldsymbol{Q}^{t+1}_{NSGA-III}$ and $\boldsymbol{Q}^{t+1}_{QPSO}$) are used to select N candidates solutions for the next population $\boldsymbol{Q}^{t+1}$. In order to do this, a simple rule based on a uniform distribution is employed.

   If the uniformly distributed random number $\mathcal{U}(0,1) > 0.5$, then the solution from $\boldsymbol{Q}^{t+1}_{QPSO}$ is accepted, otherwise from $\boldsymbol{Q}^{t+1}_{NSGA-III}$.

3. `Evaluation of` $\boldsymbol{Q}^{t+1}$`:`
   The obtained population $\boldsymbol{Q}^{t+1}$ is evaluated and is sent back with their evaluated values of fitness functions to the QPSO and NSGA-III modules.

4. `Selection for the Next Generation:`
   Now, both modules, QPSO and NSGA-III, have the same candidates for population $\boldsymbol{P}^{t+1}$. As NSGA-III is a core of the algorithm, the solutions for the next population will be selected according to the procedure presented in Algorithm 4.

   Independently, QPSO updates the local best solutions and defines the global solution among the local best solutions according to its original rules presented in [Sun et al. (2007)] and described in Section 3.4.4.

   At the end of generation t, the QPSO module returns the current global candidate solution $\boldsymbol{g}^{t+1}$ and the NSGA-III module returns the new parent population $\boldsymbol{P}^{t+1}$ to the fusion module.

At the end of the optimization process, when $t = T_{max}$:

1. in the case of single-objective optimization: the best solution is found in the current global solution $\boldsymbol{g}^{t+1}$;

2. in the case of multi-/many-objective optimization: the optimal solutions are found in the optimal Pareto front provided by NSGA-III.

In the next section, we present the experimental validation of the proposed algorithm.

5.2.2 Experimental Validation

Experimental Objectives:

1. Confirm an impact of the integrated modifications into QPSO algorithm on its performance.
2. Confirm the proposed hypothesis.
3. Investigate the robustness of QIU-NSA w.r.t. different properties of single-objective problems.
4. Validate the scalability of QIU-NSA w.r.t. the dimension of the search space on single-objective problems.
5. Investigate the performance of QIU-NSA on large scale single-objective problems, comparing it with the reference algorithms from the Black-Box Optimization Benchmarking (BBOB) workshop designed specially for large-scale optimization problems: (i) separable CMA-ES (sepCMA) [Ros & Hansen (2008)] and (ii) Limited Memory CMA-ES (LMCMA) [Loshchilov (2014)].

Algorithms and Parameters:
Since the official codes of QPSO and NSGA-III have not been published yet, we implemented C++-based codes of the following versions:

1. QPSO: the version of algorithm described in Section 3.4.4 and originally presented in [Sun et al. (2007)];
2. NSGA-III: the version of algorithm described in Section 3.4.5 and originally presented in [Deb & Jain (2013)].

In this experimental study, all algorithms are benchmarked with their default parameter settings, presented in Table 5.10.

Table 5.10: Parameter settings of peer algorithms.

Parameter	NSGA-III	QPSO	QIU-NSA
SBX p_c	1.0	-	1.0
SBX η_c	30	-	30
Poly. mut. p_m	$1/d$	-	$1/d$
Poly. mut. η_m	20	-	20
α	-	linearly decreasing	0.75

Test Problems: 24 single-objective noiseless benchmarking problems (*bbob*) and their large-scale extended version ($bbob-largescale$) from Black-Box Optimization Benchmarking (BBOB) [Hansen et al. (2009)], [Hansen et al. (2012)] test suite of the COmparing Continuous Optimizers (COCO) platform [Hansen et al. (2021)]. The description of the BBOB test suites with an explanation of the results interpretation are provided in Section 3.5. The expressions and bounds of the test functions are provided in details in [Hansen et al. (2009)].

Performance Metrics: Expected Run Time (ERT), Empirical Cumulative Distribution Functions (ECDFs) described in 3.5.1.

Simulation Settings:

1. The reported results are based on 15 independent runs of 15 instances of each function. Thus, every test function was run 225 times for each dimension d.
2. The stopping criterion is the number of generations.
3. We restrict our attention to $d \in \{5, 20, 640\}$ dimensional variants.
4. The population size N is defined according the recommendation from the original paper [Deb & Jain (2013)], which is described in Section 3.4.5.4: i.e., $N = 4$ for single-objective optimization problems.

5.2.2.1 Validation of the QPSO Modifications

In order to confirm the efficiency of the integrated modifications in the QPSO algorithm, we compare two QPSO versions: (i) QPSO with the method of linearly decreasing α described in Section 3.4.4 and (ii) the modified QPSO presented in Section 5.2.1.1.

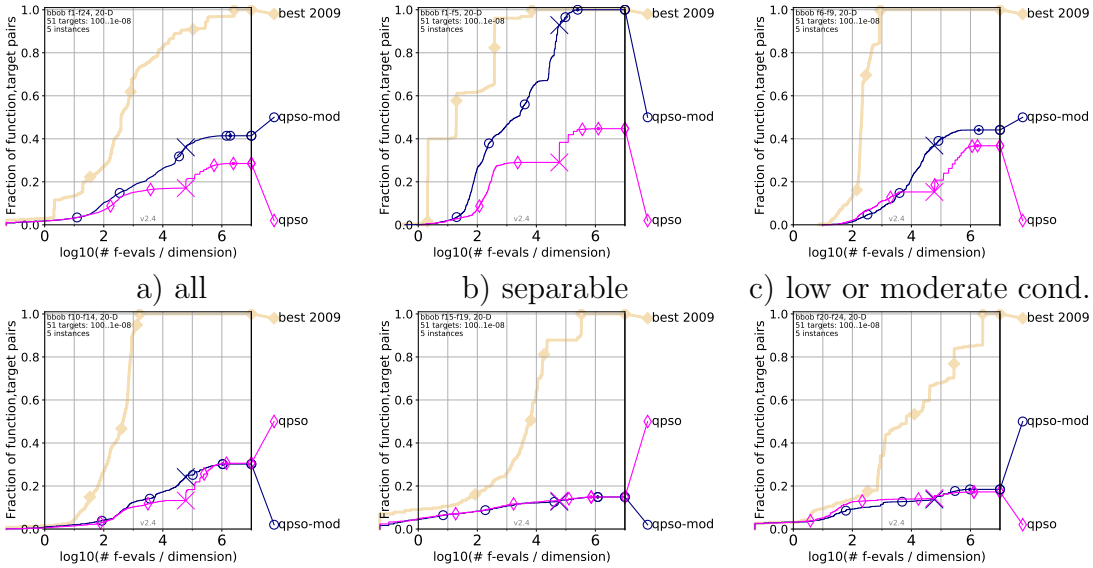
We restrict our attention to the following test configuration:

1. the dimensional search space: $d = 20$;
2. the population size: $N = 4$ (since it is our study case for the fusion method);
3. the total number of generations: $T_{max} = 300000$.

The comparative results are presented in Figure 5.11, where each graph depicts the empirical cumulative distribution of runtime summarized by function groups of the BBOB test suite, obtained by the mentioned above versions of QPSO; the blue curve is the modified version and the red curve is the original one. As usual, the transparent line "best-2009" is an aggregation of the best results observed in the BBOB-2009 workshop and is not taken into account in this study.

Figures 5.11 (a) shows the performance, in the terms of the Empirical Cumulative Distribution Functions (ECDFs), obtained on all test functions, where we can see that the modified version of QPSO covers the largest area under its graph and consequently, performs better than the original version. More precisely, it solves 40% of all functions with the highest precision, as its blue curve achieves the value 0.4 on y-axis. Whereas, the linear based method solves around 28 % of all functions (see the red curve).

Now, let us estimate the performance for each group of functions. Figure 5.11 (b) clearly shows that the modified version works perfectly with a small population size on the separable problems. It solves 100% of the test functions with the highest accuracy and outperforms the original QPSO algorithm by the average runtime almost from the beginning: the blue curve is higher during all intervals of x-axis. This result confirms the positive impact of the integrated modifications on the performance of QPSO on separable problems, which is required in the context of the optimization problems of the Hamiltonian model of Magneto Caloric Materials.



a) all b) separable c) low or moderate cond.

d) high cond./uni-modal e) multi-modal adeq. g.s. f) multi-modal weak g.s.

Figure 5.11: Empirical cumulative distribution of runtime, summarized by function groups on a 20-dimensional search space.

The results obtained on the low/moderately conditioned functions (see Figure 5.11 (c)) also confirm the efficiency of the integrated modifications for small population sizes on this group of functions. However, the positive results are less pronounced: the modified version solves around 42% of the problems of this group comparatively with the original version, which solves 38% of the problems.

The experiments on ill-conditioned and multi-modal functions (see Figure 5.11 (d, e, f) reveal that both versions perform in a similar manner for these groups of functions: their curves cover the same areas. It can be explained by the very small size of the population ($N = 4$), which is unsuitable for multi-modal problems, but was selected in order to test the modified version in the required conditions for the fusion-based method.

We can conclude, that the integrated modifications are effective and allows QPSO to work with a very small population size, but mostly on separable problems. The provided modifications do not take any advantages for solving highly conditioned and multi-modal problems.

5.2.2.2 Validation of the proposed hypothesis

In order to confirm an ability of the proposed fusion method to scale down for solving single-objective problems, we benchmark it, as well as NSGA-III and QPSO, on the following three BBOB test problems in 20-dimesional search space:

1. Rastrigin-Bueche: separable, highly multi-modal, with a highly asymmetric placement of the optima ($f4$, Figure 5.12 (a));

2. Rosenbrock rotated: uni-modal, with rotation, separability cannot be exploited ($f9$, Figure 5.12 (b));

3. Weierstrass: multi-modal and highly rugged ($f16$, Figure 5.12 (c)).

Table 5.11 gives the mean ERT for $\Delta f = 10^{1,0,-2,-3,-7}$ divided by the best Expected Run Time (ERT) obtained during BBOB-2009. The best results are printed in bold and the dispersion is presented in brackets.

We observe that QIU-NSA outperforms QPSO and NSGA-III on all functions and obtains the highest accuracy ($\Delta f = 1e-7$) on $f4$ and $f9$. However, we notice that QIU-NSA and NSGA-III perform in a similar manner at the first stage on all functions: they need a small number of function evaluations, due to a tiny population size (see Table 5.10). But by the same reason, NSGA-III shows a strong non-invariance to rotation ($f9$) and to multi-modality ($f4$, $f16$): the restriction on the population size makes the recombination operator inefficient to find new useful solutions. This effect is growing with increasing dimension of the search space (see Figures 5.13 and 5.14).

The reasons why QIU-NSA is more robust on separable multi-modal and uni-modal rotated function become clearer by looking at the coefficient of diversity and at the evaluation function value (see Figure 5.12), which visually supports our conjecture that the improved version of QPSO maintains diversity of the population and allows the algorithm to find optimum with a high accuracy.

The performance on the non-separable multi-modal function ($f16$) is less pronounced, because they are more difficult to solve with such a small population size. Nevertheless, we observe a small improvement ($\Delta f = 1e+0$) compared with NSGA-III and QPSO.

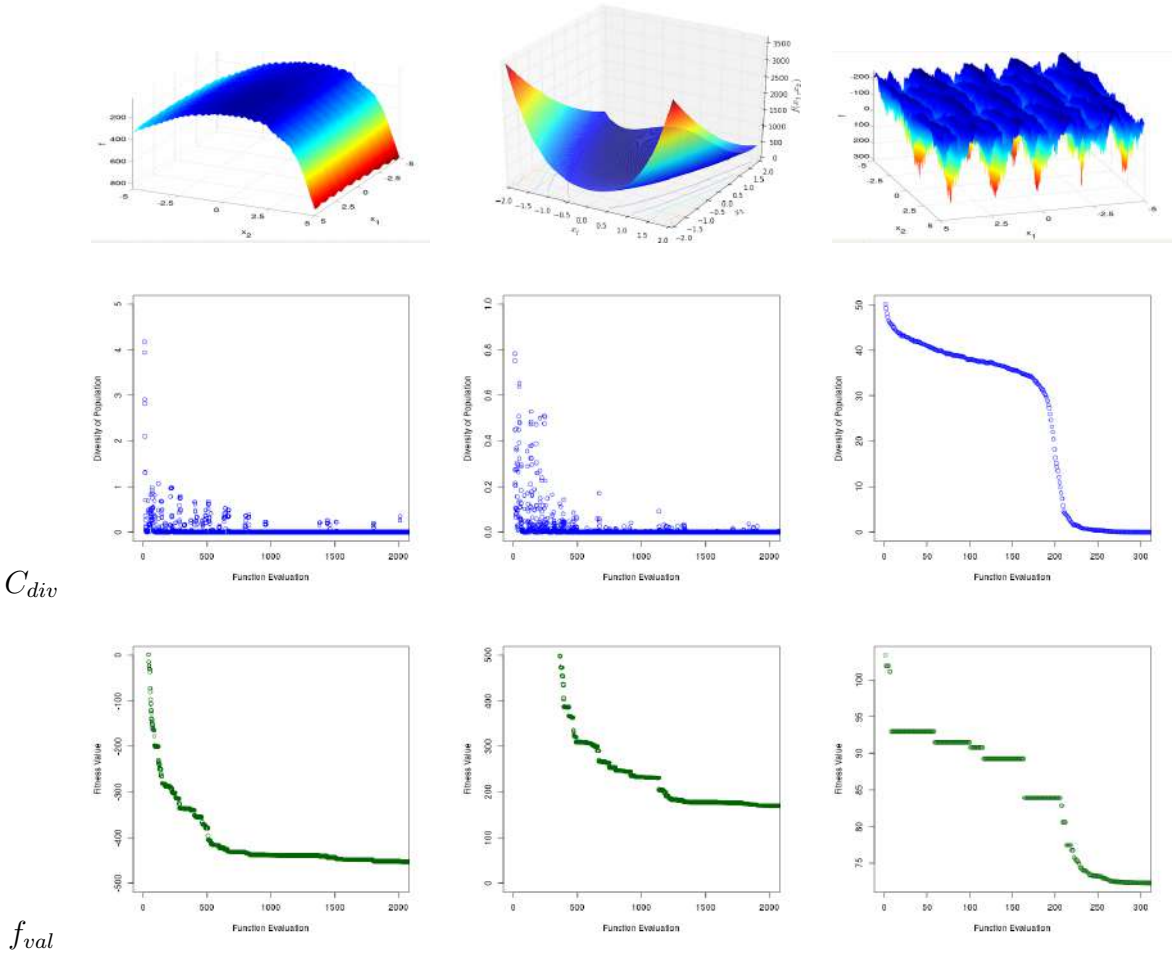
Table 5.11: Comparison of algorithms by expected runtime divided by the respective best ERT measured during BBOB-2009 in a 20-dimensional search space.

Algorithm	Δf	$1e+1$	$1e+0$	$1e-2$	$1e-3$	$1e-7$
QIU-NSA		$1.5_{(0.6)}$	$\mathbf{3.1}_{(1)}$	$\mathbf{3.9}_{(1)}$	$\mathbf{4.0}_{(2)}$	$\mathbf{0.31}_{(0.2)}$
QPSO	$f4$	$5940_{(4659)}$	∞	∞	∞	∞
NSGA3		$\mathbf{1.5}_{(0.1)}$	$3.8_{(1.5)}$	∞	∞	∞
QIU-NSA		$32_{(4)}$	$\mathbf{105}_{(10)}$	$\mathbf{212}_{(31)}$	$\mathbf{417}_{(62)}$	$\mathbf{3168}_{(76)}$
QPSO	$f9$	$670_{(532)}$	∞	∞	∞	∞
NSGA3		$\mathbf{30}_{(6)}$	∞	∞	∞	∞
QIU-NSA		$\mathbf{11}_{(0.2)}$	$\mathbf{2898}_{(420)}$	∞	∞	∞
QPSO	$f16$	$111_{(367)}$	∞	∞	∞	∞
NSGA3		$\mathbf{11}_{(0.2)}$	∞	∞	∞	∞

5.2.2.3 Benchmarking on the Noiseless BBOB-2019 Testbed

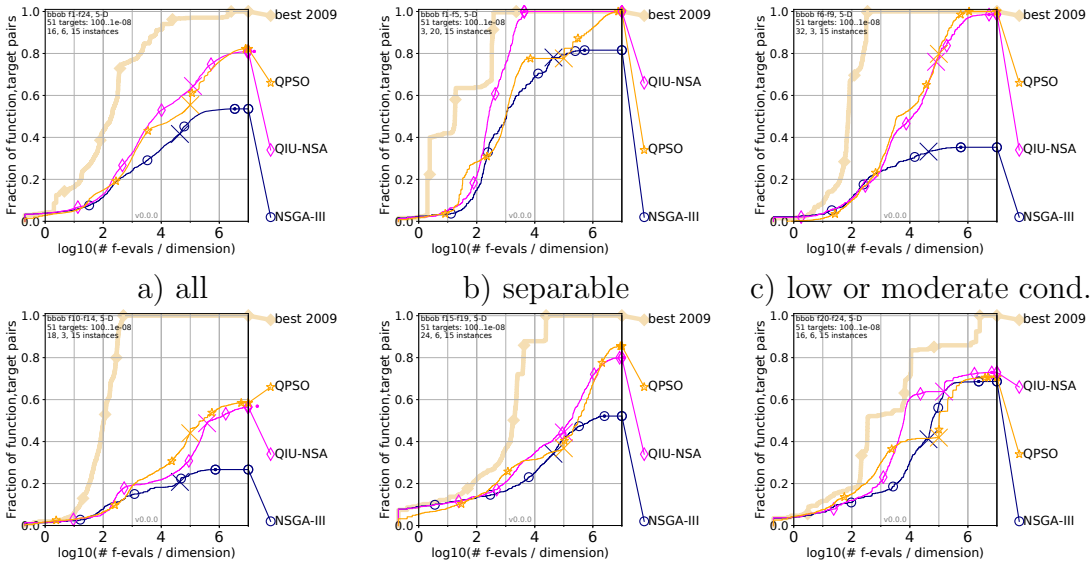
We compare the results of NSGA-III, QPSO and QIU-NSA on the BBOB test suite of 24 noiseless functions (*bbob*), aiming at validating the scalability of QIU-NSA w.r.t. the dimension of the search space and the robustness w.r.t. the different properties of single-objective problems.

First, let us present the results for a 5-dimensional search space on 24 BBOB functions. From Figures 5.13 (a), which depicts the results of ECDF on all functions for 5-dimensional search space, we can conclude that QIU-NSA covers almost the same area under its graph as QPSO, that corresponds to 80% of solved functions with the highest precision, whereas NSGA-III solves only 55% of all functions. Figure 5.13 (b) clearly shows that QIU-NSA and QPSO solves 100 % of separable functions with the highest accuracy, where QIU-NSA demonstrates the best speedup and outperforms both algorithms by the average runtime already from the beginning: its curve is higher during all intervals of x-axis,



a) Rastrigin-Bueche (f4) b) Rosenbrock (f9) c) Weierstrass (f16)

Figure 5.12: Diversity coefficient (c_{div}) and objective function value (f_{val}) w.r.t. the number of function evaluations on a 20-dimensional search space.



a) all b) separable c) low or moderate cond.

d) high cond./uni-modal e) multi-modal adeq. g.s. f) multi-modal weak g.s.

Figure 5.13: Empirical cumulative distribution of runtimes, summarized by function groups on a 5-dimensional search space.

which confirms its property of separability exploitation. Note that this property of QIU-NSA is very desirable for solving the optimization problems of the Hamiltonian model of MCMs.

The experiments on uni-modal, ill-conditioned and multi-modal functions (see Figure 5.13 (c, d, e, f) reveal that QIU-NSA and QPSO perform in a similar manner for these groups: their curves cover the same areas. Both algorithms significantly outperform NSGA-III on uni-modal and ill-conditioned functions by average runtime and accuracy.

We notice, that QIU-NSA performs twice slower on the low/moderately conditioned functions functions and almost 3 times slower on high ill-conditioned uni-modal functions compared with its results on separable functions (see Figure 5.13 (b, c, d)). We can conclude, that the search cost of QIU-NSA is sensitive to the conditions number of the ill-conditioned functions.

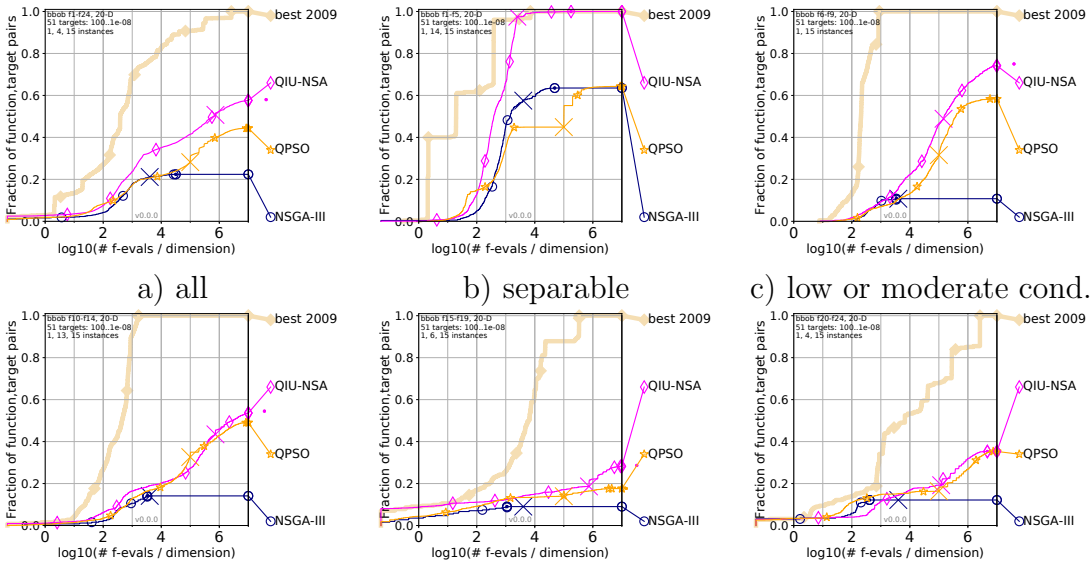
Next, we study the results for a 20-dimensional search space: Figure 5.14 shows the comparative results of the same algorithms on the *bbob* test suite.

From Figures 5.14 (a), which presents the results of ECDFs for all 24 problems on a 20-dimensional search space, we notice that QIU-NSA lost 20% of solved functions comparatively with the results obtained on a 5-dimensional search space. However it solves 60% of problems, followed by QPSO, which solves only 42% of them. The curve of QIU-NSA covers a larger area than QPSO, which shows a better speedup (QIU-NSA is around 1.25 times faster).

Figures 5.14 (b) confirms the scalability of QIU-NSA w.r.t. the dimensions of the search space on separable functions. Its curve covers almost the same area as for a 5-dimensional search spaces.

On the low/moderately conditioned functions (Figures 5.14 (c)) functions, QIU-NSA slightly outperforms QPSO, which demonstrates a better invariance to rotations than QPSO.

On the rest of the functions (see Figure 5.14 (d, e, f)) the performance of QIU-NSA



a) all b) separable c) low or moderate cond.

d) high cond./uni-modal e) multi-modal adeq. g.s. f) multi-modal weak g.s.

Figure 5.14: Empirical cumulative distribution of runtimes, summarized by function groups on a 20-dimensional search space.

and QPSO is almost identical.

Overall, QIU-NSA outperforms NSGA-III and QPSO on all groups of functions.

5.2.2.4 Benchmarking on Large Scale BBOB-2019 Testbed

Now, we benchmark QIU-NSA on the large-scale benchmark suite ($bbob - largescale$) of the COCO platform for a 640-dimensional search space under the same default parameter settings as they were used in the previous experiments (see Table 5.10).

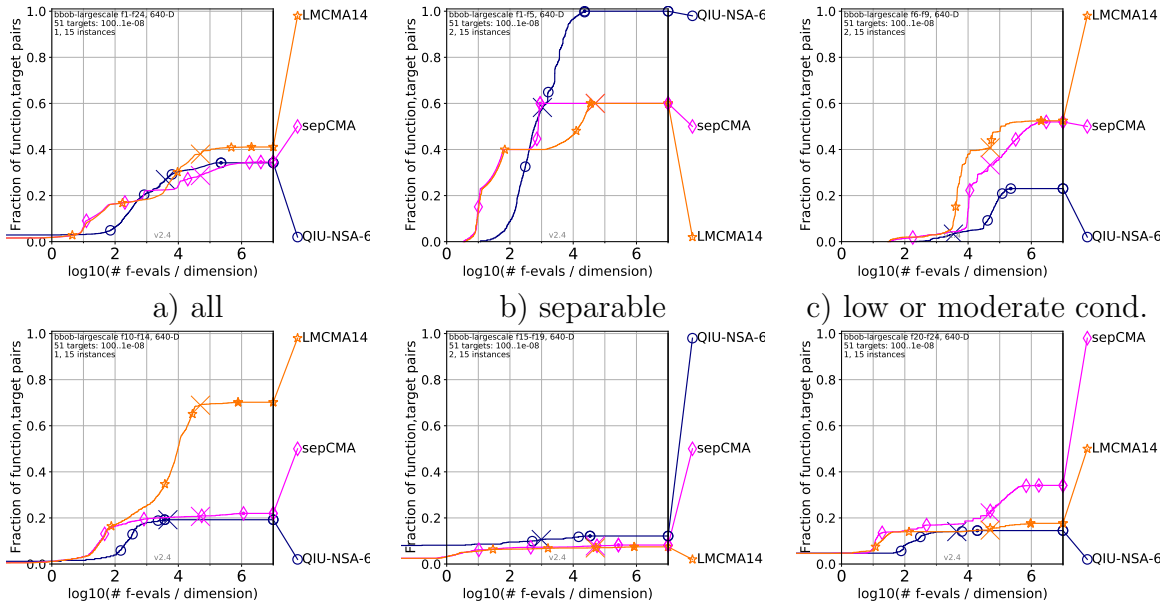
We compare the obtained performance with the reference results of separable CMA-ES (sepCMA) [Ros & Hansen (2008)] and Limited Memory CMA-ES (LMCMA-ES) [Loshchilov (2014)] provided by the COCO platform.

The results are presented in Figure 5.15. Looking at the aggregated Empirical Cumulative Distribution Functions (ECDFs) of all functions in Figure 5.15 (a), we observe that QIU-NSA and sepCMA solve around 30% of all problems. LMCMA-ES is the best one, which solves 40%.

A great performance is shown by QIU-NSA on separable functions (see Figure 5.15 (b)), where it solves 100% of the problems. It improves the BBOB records for a 640-dimensional search space on the Rastrigin separable and Skew Rastrigin-Bueche separable benchmark problems (see Figure 5.15 and Figure 5.16): according to the COCO reference results of different algorithms for optimization in large dimensions presented in [Varelas (2019)] on the same testbed of COCO, QIU-NSA shows the best results have ever been achieved on these functions in a 640-dimensional search space.

However, from Figure 5.15 (c, d) we notice that QIU-NSA starts to exhibit a non-invariance w.r.t. function rotations with a growing search space size, where it solves only 22% of the problems.

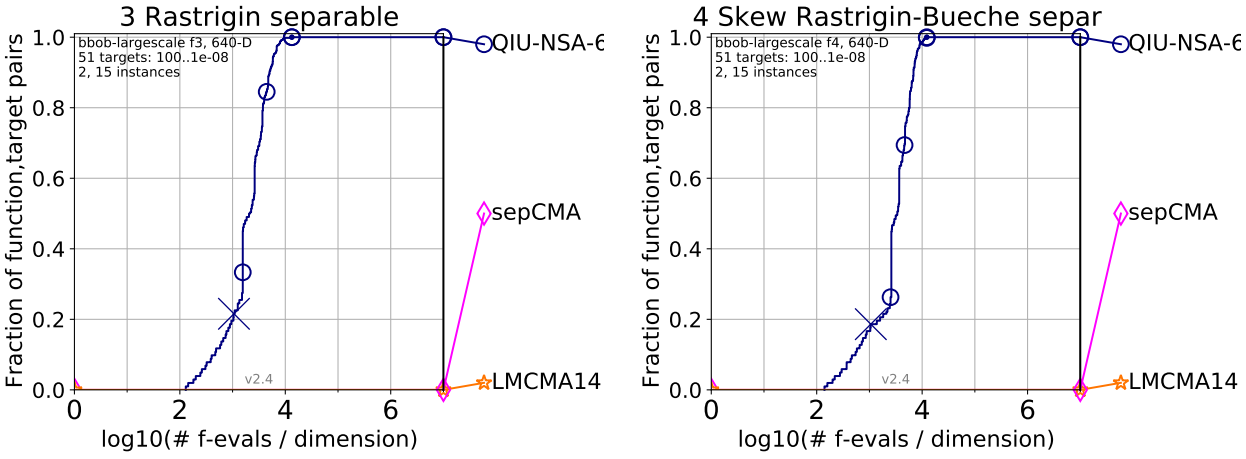
On multi-modal functions with adequate global structure (g.s.), QIU-NSA demonstrates a small, but not significant advantage (see Figure 5.15 (e)). On multi-modal



a) all b) separable c) low or moderate cond.

d) high cond./uni-modal e) multi-modal adeq. g.s. f) multi-modal weak g.s.

Figure 5.15: Empirical cumulative distribution of runtimes, summarized by function groups on a 640-dimensional search space.



a) Rastrigin Sep. b) Rastrigin-Bueche Sep.

Figure 5.16: Empirical cumulative distribution of runtimes on Rastrigin separable functions on a dimension 640-dimensional search space.

problems with weak global structure (g.s.) in Figure 5.15 (f), the improvements brought by QIU-NSA are less pronounced. These results are explained by the complicated landscapes of the multi-modal problems and the too small population size.

5.2.3 Summary and Discussions

In this work, we have put forward a unified algorithm, which endows a many-objective optimization algorithm, NSGA-III, with the ability to solve single-objective problems by employing the modified version of the QPSO algorithm via a fusion method. The advantages of the presented algorithm, called QIU-NSA, are the following:

1. The fusion method allows us to scale NSGA-III down to solve single-objective problems without any modifications in the NSGA-III structure, which excludes a risk to lose its effectiveness on multi-/many-objective problems.

2. QIU-NSA is a parameter independent algorithm, because it works with the default parameters of the modified version of QPSO and NSGA-III.

3. QIU-NSA demonstrates a stable average performance on different classes of problems without any additional tuning of the parameters.

4. QIU-NSA is very efficient to solve separable problems.

5. QIU-NSA shows a strong scalability w.r.t. the dimensions of the search space on separable problems.

Contribution:

1. A new successful application of the fusion method in order to constitute a scalable algorithms w.r.t. the dimensions of the objective and search spaces.

2. A new study, where a quantum-inspired algorithm, QPSO, is improved for working with a small population size;

3. The proposed fusion-based algorithm allows users to simplify their real-world research study by working with a single software for optimizing problems with different dimensions of objective and search spaces.

Discussion: The results obtained from the experimental study conducted on the single-objective Black-Box Optimization Benchmarking (BBOB) test suite, confirm that QIU-NSA retains the properties of the modified version of the QPSO algorithm and improves NSGA-III ability to solve single-objective problems with a restricted population size.

We observed during the experimental study that QIU-NSA performs perfectly on all separable functions: i.e., the functions, which can be optimized coordinate-wise. QIU-NSA solves all 5 provided problems with the highest target precision $\Delta f = 10^{-8}$ on all instances and in all dimensions: from $d = 5$ to $d = 640$. Among them, QIU-NSA solves multi-modal separable functions, without any premature convergence to a local optimum: i.e., the Rastrigin separable and Skew Rastrigin-Bueche separable functions. Moreover, on these functions, in a 640-dimensional search space, QIU-NSA shows the best results have ever been achieved, according to COCO reference data presented in [Varelas (2019)].

This result can be explained by the combination of the coordinate-wise variation operators of NSGA-III and an uni-variant Cauchy distribution-based component of QPSO, which is used for maintaining diversity. However, as already discussed in Section 4.2.5, the Cauchy distribution is heavily coordinate-dependent and consequently, can be efficient only on separable functions. The marginal performance on non-separable multi-modal functions appears when the search space dimension is larger than 10. Furthermore, as seen from the experimental results, QIU-NSA performs slowly on ill-conditioned functions, where its search cost (i.e., the number of function evaluations) depends on the number of conditions of the problem.

The learning of correlations between the solution variables needs to be included in order to fix this drawback. One of the possible solution to improve the performance on non-separable functions is to replace uni-variate by multi-variate Cauchy distribution, presented in [Lee & Park (2014)], which is demonstrating its usefulness for problems with correlated decision variables.

However the positive results on separable functions, especially in large dimensions, lead us to search a method to exploit this separability property: a transformation method is needed to "un-correlate" the variables and convert a non-separable problem into a separable one. We assume that the feed-forward neural network based method presented in [Lu & Ito (2003)], could be used for such kind of conversion and joined with QIU-NSA.

Limitations: The main limitation of QIU-NSA is its non-invariance w.r.t. the function rotations with an increase in the search space size. Even though it may be not currently critical from the point of view of the research problems of this thesis, this fact excludes its application on multi-modal non-separable single-objective problems.

Perspectives: A short term perspective is to extend our approach and to improve the performance of NSGA-III on bi-objective functions via the fusion method.

A long term perspective is to improve robustness of QIU-NSA w.r.t. non-separable functions.

Connection with research problems: QIU-NSA is developed as a unified algorithm, considered to solve from single- to many-objective problems and to be applied for solving the research problems of this thesis. Thus, the experimental results on the real-world problems are presented in the next chapter, where one of the problems of the Blume–Emery–Griffiths–Ising model of Magneto Caloric Materials (MCMs) is solved in Section 6.2.

.

Chapter 6

Optimization of Simulation Models

First, as an auxiliary work, in Section 6.1, we present a new extended version of the EASEA platform, which supports multi-objective optimization and an optimization tool based on this platform. Employing this tool, in Section 6.2, we introduce an optimization-based method for reproducing the temperature and magnetic-field dependence of magnetization and heat capacity for different Magneto Caloric Materials (MCMs). Since the first challenge is completed, in Section 6.3 we proceed to the second challenge, where we analyse the AMR model of the *Ubiblue* company as an optimization function and provide a flexible tool to explore innovative architectures of the AMR model, operating in the mode of Magnetic Refrigeration System (MRS) and in the mode of Thermo-Magnetic energy Generator (TMG).

6.1 Optimization Tool

In practice, users want to have a user-friendly software tool for quickly solving their complex real-world problem, such as the optimization of the simulation models of Magneto Caloric Materials (MCMs) and Active Magnetic Regenerator (AMR). Due to the fact that the best optimization algorithm for a specific user problem is a priory unknown, a framework of single and multi-optimization evolutionary algorithms is required to easily select the most efficient algorithm for quickly obtaining the best set of non-dominated solutions of this problem with the required accuracy.

On the other hand, developers of algorithms also need an instrument to design, test and compare novel high-performance MOEAs. Such kind of instrument has to provide many different benchmark suites and set of performance metrics for tests and comparative studies.

For both developers and users, a framework with different optimization algorithms and with a convenient user interface is desirable in order to quickly organize an experiment, obtain the results and make the comparative experiments,

For this reason, in this thesis, we extend the EASEA (EAsy Specification of Evolutionary Algorithms) platform for multi-objective optimization. This updated version of EASEA can be useful not only in the frame of this thesis, but also for other developers of optimization algorithms and users.

6.1.1 EASEA Version 2.20

The EASEA platform presented in presented in Section 3.6 was chosen as a host framework for the algorithms developed in this thesis, and as a software core for the optimization of the simulation models. For this purpose, a certain part of this thesis was spent for updating EASEA and extending the library version of EASEA, *LibEASEA*, mainly towards multi-objective optimization. It takes time and effort to implement different optimization algorithms well, due to the fact, that if they are not carefully implemented, the optimization algorithms can work quite slow or be inaccurate. In order to generalize the development of Multi-Objective Evolutionary Algorithms (MOEAs), a multi-objective optimization sub-library is implemented into *LibEASEA* (see Section 6.1.1.1).

The updated version of EASEA provides the following features:

1. **Support of the following new templates (**.tpl*)**:
   - Single-objective optimization algorithms:
     (a) CPU parallel version of QAES [*QAES.tpl*], presented in Section 5.1.
     (b) GPU parallel version of QAES [*CUDA-QAES.tpl*].
     (c) CPU parallel version of QES [*QIEA.tpl*], developed in accordance with [da Cruz et al. (2010)].
   - Multi-objective optimization algorithms:
     (a) CPU parallel version of FastEMO [*FastEMO.tpl*], presented in Section 4.
     (b) CPU parallel version of six state-of-the-art MOEAs, developed in accordance with the cited papers: NSGA-II (*NSGAII.tpl*) [Deb et al. (2000)], NSGA-III (*NSGAIII.tpl*) [Deb & Jain (2013)], ASREA (*ASREA.tpl*) [Sharma & Collet (2010*b*)], MOEA-D (*MOEAD.tpl*) [Zhang & Li (2007)], IBEA (*IBEA.tpl*) [Zitzler & Künzli (2004)], CDAS (*CDAS.tpl*)[Sato et al. (2007)].
   - Unified optimization algorithms:
     (a) CPU parallel version of QIU-NSA, presented in Section 5.2.
2. **Parallelism support**: For all MOEA templates, we currently provide two possible execution scenarios of MOP evaluation: sequential and multi-threaded on CPU. The multi-thread mode evaluation is organized by applying OpenMP[1] (the API specification for parallel programming).

   OpenMP was chosen, because it supports different compilers (GNU, Clang, Microsoft Visual *C++*) and does not require to significantly modify the sequential source code of the algorithm in order to add parallelism into it.

   If OpenMP is not supported by the compiler (or is not installed), the code of MOEAs will be compiled as executable and will have a correct behaviour, without breaking compatibility with compiler, but without any parallelism.

   By default, all templates work in sequential mode, and can be launched in CPU parallel mode by applying the special command line option *"–nbCPUThreads N"* (where N is a required number of CPU threads).

[1] `https://www.openmp.org/`

3. **Implementation of the following testbeds as problem-specific files (**.ez*)**:
   (a) Zitzler–Deb–Thiele (ZDT) [Zitzler et al. (2000)].
   (b) Deb–Thiele–Laumanns–Zitzler (DTLZ) [Deb et al. (2002)], Walking Fish Group (WFG) testbeds [Huband et al. (2005)].
   (c) Unconstrained functions [Zhang et al. (2008)].
   (d) Black-Box Optimization Benchmarking (BBOB) (described in Section 3.5).

   These testbeds are accessible in the catalog of examples, distributed as a part of the EASEA software and can be used to compare the effectiveness of the different MOEAs.

4. **Linux-based "*no-sudo*" [no-sudo] EASEA local installation tool**: is a *bash* script, developed as an auxiliary tool, to allow to locally install EASEA (with all package dependencies, except *cuda*) in a user's directory, without *sudo* permissions. This can be useful whenever user does not have root access, like in shared hosting service or in shared computer system of university. It works only with the Advanced Package Tool[2] (*apt*) and only for linux-based distributions (Debian, Ubuntu).

6.1.1.1 LibEASEA

The new version of *LibEASEA* is developed as an extension of the former version, without changing the object-oriented design and the structure of the library (all experiments (**.ez* files), developed for the previous version are supported by this one). The new sub-library (described below) provides new features (necessary for this thesis), while keeping the old functionalities. In order to be easily integrated into *LibEASEA*, these sub-libraries are developed as header-only, so nothing needs to be compiled separately from *LibEASEA*.

For the purpose of code reusability and flexibility, a powerful *C++* templates technique is used in the new sub-library of *LibEASEA*. Thus, various data types can arbitrarily be defined by the template parameter. It helps to realize generic programming, which allows the generic algorithms to process different data structures. This technology performs at compile-time and consequently can significantly increase efficiency. Furthermore, by using *C++* templates, users can easily change (in one line of source code) the types of decision variables according to the problem specifications, that can help to give a speedup benefit in many situations. For example, if a high precision is not necessary, user can define the type of decision variables as *float*, due to manipulating single precision floating point numbers is computationally cheaper than manipulating double precision ones (and this saving can add up when big population size is used).

The following sub-libraries are implemented in the version 2.20 of EASEA:

1. **Multi-objective optimization**: is a tiny header-only sub-library, the main purpose of which is to provide the class and function templates to offer fast implementations for a wide range of MOEAs, which can all be used for optimization of the simulation models or other optimization problems. It consists of the following modules:

[2] `https://guide.ubuntu-fr.org/server/apt.html`

(a) Classes of multi-objective algorithms in [*/libeasea/include/algorithms/*].

(b) Auxiliary classes and functions in [*/libeasea/include/shared*].

(c) Classes and functions of operators in [*easea/libeasea/include/operators/*]:

- Crossover: Simulated Binary Crossover (SBX) [CsbxCrossover.h], New adaptive crossover (presented in Section 4) [CbetaCrossover.h];
- Mutation: Polynomial [CPolynomialMutation.h], Cauchy [CCauchyMutation.h], Gaussian [CGaussianMutation.h];
- Selectors: Nondominate [nondominateSelection.h], Random [randomSelection.h], Tournament [tournamentSelection.h].

(d) Base class for all Multi-objective Optimization problems (MOPs) [*/libeasea/include/problems*].

Figure 6.1 illustrates the Unified Modeling Language (UML) diagram of the architecture of this sub-library. This diagram is composed of the above-mentioned modules (a - yellow, b - light green, c - light and dark blue, d - light brown), that belong to different namespaces. Each namespace has at least one abstract class, with a set of abstract methods to create a unified interface between these namespaces. Thus, the derived classes in different namespaces are combined through this interface.

Due to the *C++* template design, the types of decision variables and of objective functions can be defined by user in the problem-specification file. In the same file, the user has to define the optimization functions in the method, called *evaluator* of class *GenomeClass* (see Block "EZ Files" in Figure 6.1). The *GenomeClass* will be associated with the class *CIndividual* in the algorithm template (*.tpl*) by the EASEA compiler (*easena*) (see Block "Templates" with ASREA algorithm example in Figure 6.1). Thus, the content of *GenomeClass::evaluator* will be automatically integrated into the method *evaluate()* of *CIndividual*.

If optimization functions are not *C++* code, the programming language independent interface (described below) is called from the method *GenomeClass::evaluator*, in order to execute the code with needed programming software as a shell command.

To use one of the Multi-Objective Evolutionary Algorithms (MOEAs), provided by EASEA, one only needs to be able to specify the parameters of his problem and optimization functions in the problem-specific file. The implementational details of each algorithm are abstracted away. As seen from the UML diagram, this sub-library provides a convenience for the development of multi-objective optimization algorithms templates (**.tpl*) and their applications in (Multi Objective Problems) MOPs (**.ez*).

2. **Performance indicators for multi-objective optimization** [Audet et al. (2020)]: Pareto non-compliant indicators - Generational Distance (GD) [CQMetricsGD.h] and Inverted Generational Distance (IGD) [CQMetricsIGD.h]; Pareto compliant indicator - hypervolume (I_{HV}) [CQMetricsHV.h].

3. **Application Programming Interface (API)** [CApiSubProcess.h]: it is the programming language independent user interface developed as a tiny one-header-only

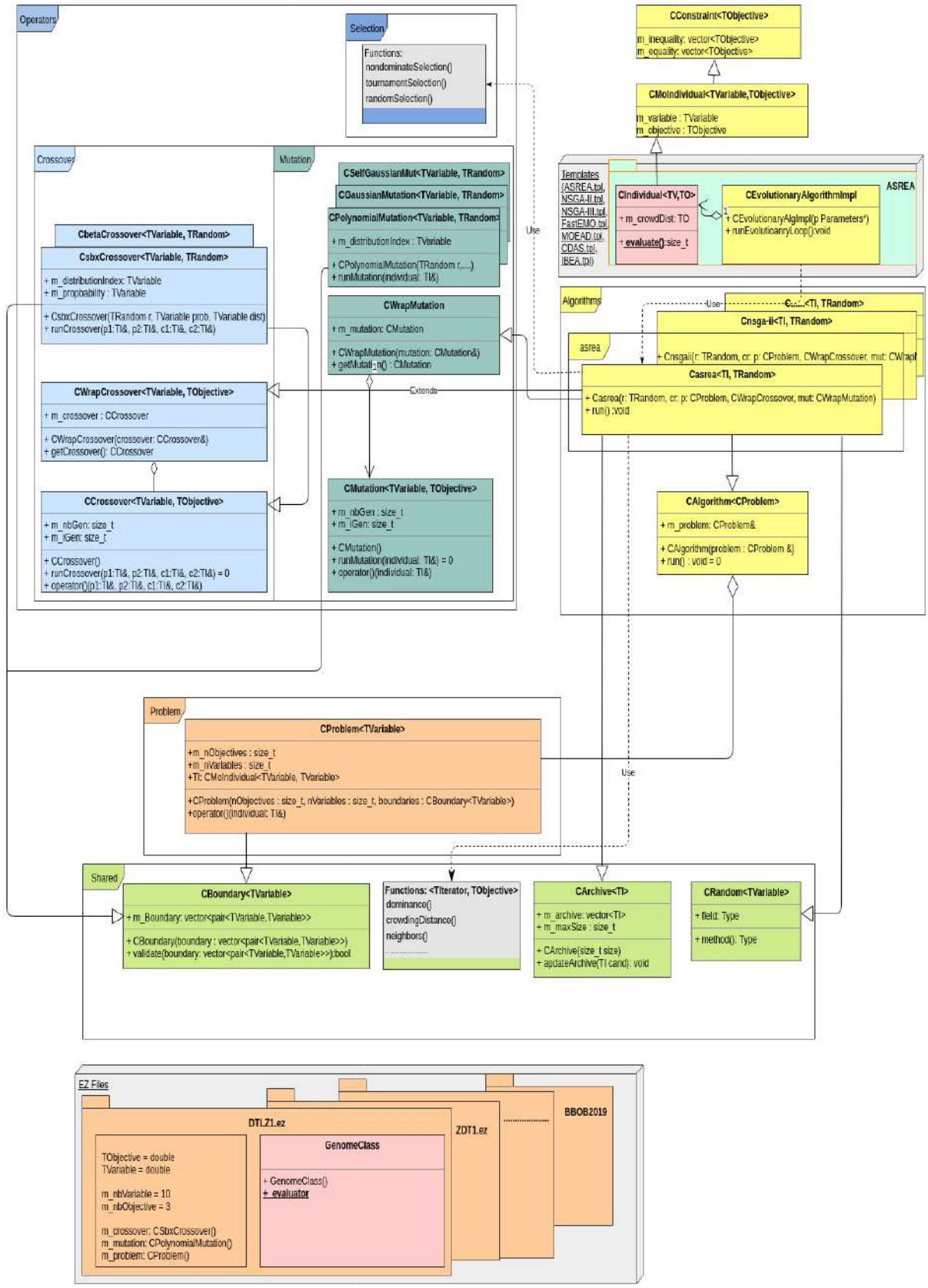


Figure 6.1: UML diagram of the basic architecture of multi-objective optimization sub-library of *LibEASEA*.

sub-library, which provides *C++*-based function(s) to launch the external problems (e.g., simulation models) implemented on different programming languages or compiled as an executable file. as a shell commands from the problem-specific file (**.ez*).

The working principle of the proposed API is similar to *popen()*[3]: it creates a new process and executes the command (specified by the string command, which calls external SCILAB code or binary executable file of AMR model) by invoking shell scripts. The difference compared to *popen()* is that *stdin*, *stdout* and *stderr* of the child process are associated to *iostream* objects which can be accessed by the parent process to easily take the output values, simulated by AMR model. To construct these *iostream* objects from file descriptors, the GNU extension *_gnu_cxx::stdio_filebuf*[4] is used. It works with GNU *C++* and can work with Clang, as long as GNU *C++* standard library is used.

In this work, this feature is necessary, due to the AMR simulation model being developed as a SCILAB code, which has to be executed in parallel on CPU from the problem-specific part of EASEA.

4. **Customised logger, *logg*** [CLogFile.h]: is a tiny one-header-only sub-library. It provides customised and extended preservation of experimental settings and experimental data, which is enough for reproducibility of research. By default, it creates the "*.log" file in the current directory, where all settings information about experiment is saved: run configuration (date and time of experiment, seed, population size, number of generations, number of CPU threads), special options of run (all settings from *.prm file), results (the best solution and its objective function value). User has an access to the logging system, *logg*, from his problem-specific file ("**.ez*"), in order to add any other needed information.

The presented version of EASEA is available in the official GitHub repository of EASEA[5].

6.1.2 Architecture of the Optimization Tool

The optimization tool, proposed in this section, aims at providing researchers with a comfortable, flexible and user-friendly software instrument for single/multi-objective/unified optimization of complex simulation models, with the help of the updated version of the EASEA platform described in the previous subsection. The EASEA-based architecture of the tool is depicted in Figure 6.2.

As explained in Section 3.6 and Section 6.1.1, an optimization problem has to be formulated in its problem specific file (*.ez*), where the following attributes of the problem have to be defined:

1. Configuration attributes: (i) problem configuration (types of decision variables and objectives, dimension of search space and of target space, boundaries of each variable); (ii) algorithm configuration (types of crossover and mutation operators and their parameters); (iii) optional (reference values or needed constants).

[3] `https://man7.org/linux/man-pages/man3/popen.3.html`
[4] `https://gcc.gnu.org/onlinedocs/gcc-4.6.2/libstdc++/api/a00069.html`
[5] `https://github.com/EASEA/easea`

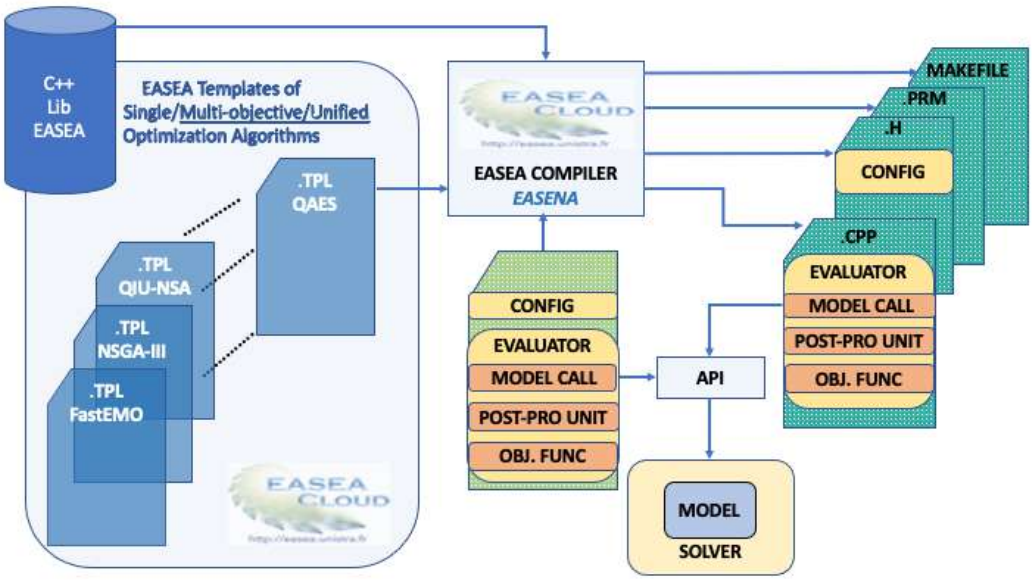


Figure 6.2: The EASEA-based architecture of proposed optimization tool.

2. Evaluator: (i) call of model *C++*-based function (if the source code of the model is not *C++*-based or is not available, the programming language independent interface of EASEA is used to launch an executable file or an external programming environment as a child process); (ii) post-processing unit, aimed at transforming the output of model to the input parameters of the objective functions for each solution candidates; (iii) evaluation functions, where the mathematical equations of the objective functions must be defined.

An example of the problem specific file definition is provided in Appendix 21.

In this thesis, in the frame of the *CoolMagEvo* project, we deal with two simulation models: the generalized Blume–Emery–Griffiths–Ising (BIG-I) Hamiltonian model of Magneto Caloric Materials (MCMs) and the dual-mode operating Active Magnetic Regenerator (AMR) model presented in Section 2.2.2 and Section 2.3.3 respectively. To be optimized, both optimization problems have to be specified in *.ez* file. These optimization problems of the Hamiltonian model and the AMR model are respectively formulated as:

1. minimization of several differences between simulated and reference physical properties of MCMs for finding optimal free parameters of BEG-I model;

2. maximization of efficiency and power density for a Magnetic Cooling System (MRS) and a Thermo-Magnetic energy Generator (TMG) operating modes for finding optimal control and design parameters of the AMR model.

Taking into account that the EASEA-based optimization tool considers every MOP as minimization optimization problem, in the Evaluator, we convert the maximization objectives into their opposite values and then restore them to get the original objective values, which helps to simplify the tool architecture.

Once the problem specific file is done, the tool can be run by doing the next three steps:

1. to call compiler *easena* for automatically merging the optimization problem of the model (specified in **.ez* file) with the code of selected optimization algorithm (provided in its template file (**.tpl*) into a CPU parallel C++ code (Example: `easena -fastemo mcm-optimiser.ez`);

2. to compile the obtained C++ code into an executable code by generated *Makefile*: `make;`

3. to run the executable file with the desired number of parallel threads (Example: `./mcm−optimiser --nbCPUThreads 50`)

It can be seen from Figure 6.2 that the EASEA-based design significantly simplifies a code-coupling process among a selected optimisation algorithm and a problem of simulation model, where the model can be a source code on different programming languages or an executable file.

The presented tool gives a large degree of freedom for users: a source code of the models can be easily and separately modified, as well as a template of selected algorithm can be replaced by another one, depending on the specific requirements of optimization problems.

To conclude, the proposed software tool allows to quickly organize an experiment and to obtain the results, because of EASEA-based user friendly interface and new implemented features.

6.1.3 Summary and Discussions

In this Section, first, we present an extended version of the EASEA (EAsy Specification of Evolutionary Algorithms) platform for single- multi-, and many-objective problems. Note that the algorithms developed in the frame of this thesis and presented in previous Sections, are hosted in the EASEA platform. Then, we introduce the optimization tool based on this platform, which contributes as a software for flexible and reliable implementation of the real world research experiments defined in the frame of this thesis. In principle, the proposed tool can be used in the different research areas where an optimization process is needed.

In the next two sections, Section 6.2 and 6.3 we present in details the investigation and experiments addressed our research problems: (i) the problem of reproducing theoretical data of MCMs by optimization of the free parameters of the the generalized Blume–Emery–Griffiths–Ising (BIG-I) Hamiltonian model; (ii) the problem of exploring innovative architectures of the AMR model, operating in the mode of a Magnetic Refrigeration System (MRS) and in the mode of a Thermo-Magnetic energy Generator (TMG).

6.2 Novel Evolutionary Method for Modeling Physical Properties of MCMs

The simulation of physical properties of Magneto Caloric Materials (MCMs) w.r.t. the measured data, is one of the major scientific and technical challenge on the way to magnetic device development.

The physical properties involved in the simulation of MCMs are presented in Section 2.2.2.2: the temperature and magnetic-field dependence of magnetization ($M(H,T)$) and of heat capacity ($Cv(H,T)$).

As it discussed in Section 2.2.1, the indirect state-of-the-art techniques for studying and simulating physical properties of MCMs have many serious disadvantages: a lack of reliability and generalization, a dependency on the databases and high time consumption.

In an effort to overcome the drawbacks of existing methods, we introduce a new reliable indirect method. In the current Section, we employ single- and multi-objective approaches for optimizing the free parameters of the generalized Blume–Emery–Griffiths–Ising (BIG-I) Hamiltonian model of MCMs detailed in Section 2.2.2.

The main idea of our method is to employ an efficient optimization algorithm to automatically find a combination of free parameters of BEG-I Hamiltonian, the values of those are required in order to reproduce the temperature dependence of magnetization and temperature dependence of heat capacity of desirable MCM. Consequently, the optimization process adjusts BEG-I model to different MCMs.

To the best of our knowledge, the simulation of MCM physical properties has never been formulated as an optimization problem before and this method is a new contribution.

This Section, first, presents a description of the proposed method for reproducing/studying physical properties of MCMs. Second, an implementation of this method through the EASEA-based software tool detailed in Section 6.1.2, is explained. Then, the applicability of the proposed method is experimentally validated on two First Order phase Transition (FOT) MCMs ($LaFe_{13-x}Si_x$, $GdSiGe$) and one Second Order phase Transition (SOT) MCM ($LaFeCoSi$) by testing the optimization algorithms presented in previous sections (FastEMO, QAES and QIU-NSA). We do not provide the precise formulas of the MCMs used in the validation experiments, because it is confidential information.

It is shown that the results obtained by the proposed method are in good qualitative agreement with the available experimental data.

A redacted version of this Chapter has been accepted and published in IEEE Congress on Evolutionary Computation (CEC) 2021.

6.2.1 Proposed Method

Our method for reproducing/studying physical properties of MCMs is schematically depicted in Figure 6.3. This simplified scheme consists of two main parts: a classical Monte Carlo simulation solver with the the generalized Blume–Emery–Griffiths–Ising (BEG-I) Hamiltonian model detailed in Section 2.2.2 and an optimization algorithm integrated in the platform EASEA. The free parameters of the Hamiltonian model detailed in Section 2.2.2.1 (Table 6.6) present the decision variables of the optimization problem, because they have a strong impact on the shape of magnetization and of heat capacity curves, which determine the order of phase transition, and on the Curie temperature field dependency.

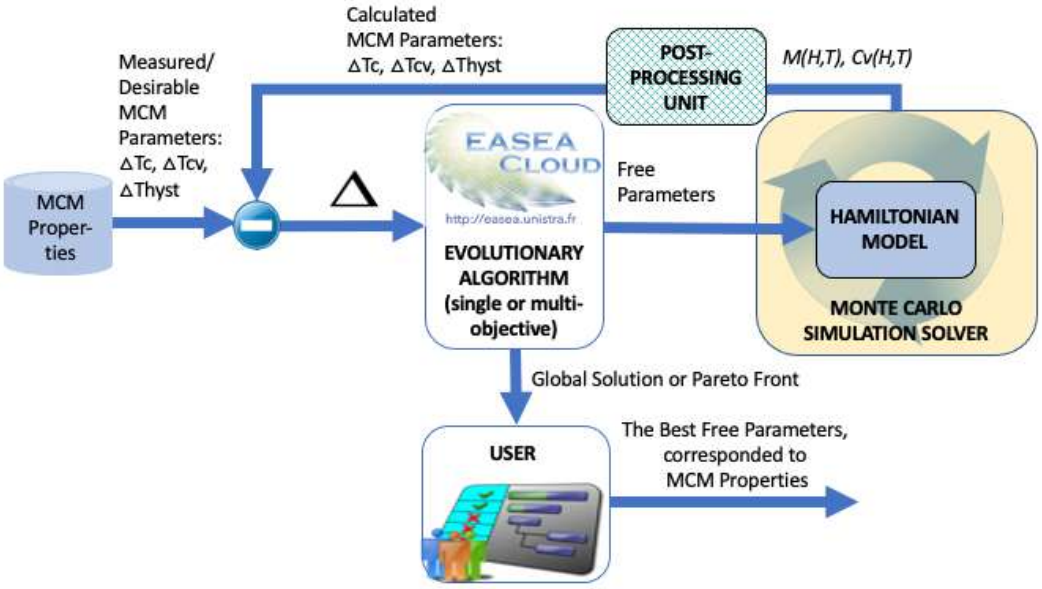


Figure 6.3: Simplified scheme of the proposed method.

The essence of this method is to define a set of optimal free parameters of the Hamiltonian model, i.e., the best solution or the Pareto Front in the multi-objective optimization case, by minimizing the difference(s) between measured/desirable parameters of Magneto Caloric Materials (MCMs) and the parameters calculated from the temperature and magnetic-field dependence of magnetization ($M(H,T)$) and of heat capacity ($Cv(H,T)$) in the post-processing unit.

The work of this scheme is emulated into five steps, performing as a cycling process:

1. The selected optimization algorithm generates a new set of solutions: combinations of the free parameters.

2. For each solution, a Monte Carlo solver solves the BEG-I Hamiltonian model to obtain the temperature and magnetic-field dependence of magnetization ($M(H,T)$) and of heat capacity ($Cv(H,T)$) for the required MCM.

3. For each solution, the post-processing unit calculates the required parameters for the objective function(s). These parameters are defined by the user and formulate the optimization problem: if some of them are conflicting parameters, the optimization problem becomes multi-objective. For different configurations of the optimization problem, the parameters used in this research are as follows: the interval width of heat capacity curve peak (ΔT_{cv}), Curie temperature field dependency (ΔT_c) and hysteresis loss (ΔT_{hyst}, only for the alloys with First Order phase Transition (FOT)).

4. For each solution, the objective functions are calculated as the differences between the values of calculated parameters in post-processing unit from simulated data and expected/measured ones.

5. The selected optimization algorithm performs the optimization process and goes to the first step, until the defined number of cycles is reached.

At the end of this process, the optimization algorithm provides the best solution in case of single-objective optimization or the Pareto Front of solutions in case of multi-objective optimization. The obtained solution(s) is/are a combination of free parameters, that tunes the energy terms intensities of the Hamiltonian to correspond to the physical properties of the expected MCM as close as possible.

In the case of multi-objective optimization, the user has to select the most appropriate combination for his needs from the Pareto Front.

In principle, this method is versatile. The different physical properties can be used as optimization objectives.

In the next subsection, the post-processing unit of the proposed method is detailed.

6.2.1.1 Post-Processing Unit

As seen in Figure 6.3, in this thesis we assume the computation of the following parameters:

- ΔT_{cv} - the temperature interval width of heat capacity curve ($Cv(H,T)$) peak under different magnetic fields (H_1, H_2).
- ΔT_c - the Curie temperature field dependency.
- ΔT_{hyst} - the hysteresis loss (only for the alloys with First Order phase Transition (FOT)) under different magnetic fields (H_1, H_2).

These parameters are used as the inputs for the objective functions. The number and combination of these parameters can be changed accordingly to the problem configuration. For example, the hysteresis loss is not taken into account for Magneto Caloric Materials (MCMs) of Second Order phase Transition (SOT), due to its absence across SOT.

In the next paragraphs, we explain how these parameters can be calculated.

Temperature interval Width of Heat Capacity Curve Peak: ΔT_{cv}

The temperature interval width of heat capacity curve peak, ΔT_{cv}, is proposed as one of the most significant parameters of the optimization process, because its value reveals the sharpness of the phase transition. Thus, ΔT_{cv} determines the transition order. In fact, it can be a more reliable parameter than the height of heat capacity peak, especially in case of Magneto Caloric Materials (MCMs) of First Order phase Transition (FOT), where the peak tends to infinity.

In this thesis, we propose the following method for calculating ΔT_{cv}: the value of ΔT_{cv} is evaluated through an approximation of heat capacity peak by two intersecting lines, as it is presented in Figure 6.4.

A point of intersection of these lines ($B(T_B, C_B)$) corresponds to the maximum heat capacity value and the temperature, related to it. To find the position of the intersected lines on the coordinate plane, the coordinates of the intermediate points $A(T_A, C_A)$ and $C(T_C, C_C)$ are defined (like the average coordinates of n left points and of n right points from the extreme point B) in Eq. 6.1 and Eq. 6.2:

$$T_A = \frac{1}{n}\sum_{i=N-n}^{N} T_i, \quad T_C = \frac{1}{n}\sum_{i=N}^{N+n} T_i \tag{6.1}$$

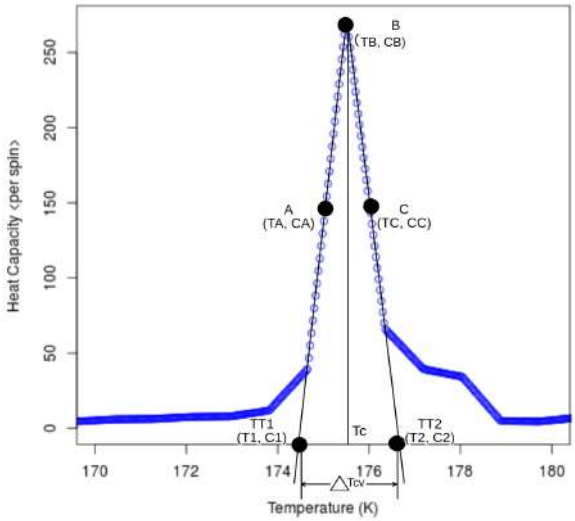


Figure 6.4: Determination of the temperature interval width of heat capacity curve peak.

$$C_A = \frac{1}{n}\sum_{i=N-n}^{N} C_i, \quad C_C = \frac{1}{n}\sum_{i=N}^{N+n} C_i \tag{6.2}$$

where N is the position in the array of the maximum heat capacity value in the array $Cv(T)$, n is the number of points, taken on the left and on the right from the peak value (it was determined experimentally and the suggested value is $n = 50$), C_i is the heat capacity value on the position i in the array $Cv(T)$ and T_i is the temperature value corresponding to C_i.

The distance between points $TT1$ and $TT2$ is assumed as the expected interval ΔT_{cv}. Thus, the equations to calculate the coordinates of points $TT1(T_1, C_1)$ and $TT2(T_2, C_2)$ have to be formulated. As the coordinates of points $A(T_A, C_A)$, $B(T_B, C_B)$, and $C(T_C, C_C)$ have been already found above, the equation for lines AB and BC can be defined as follows:

$$\frac{C_1 - C_A}{C_B - C_A} = \frac{T_1 - T_A}{T_B - T_A} \tag{6.3}$$

$$\frac{C_2 - C_B}{C_C - C_B} = \frac{T_2 - T_B}{T_C - T_B} \tag{6.4}$$

Since the points $TT1$ and $TT2$ lie on the abscissa axis, the coordinates T_1 and T_2 are evaluated by substituting $C_1 = 0$ and $C_2 = 0$ into Eq. 6.3 and Eq. 6.4 according to the obtained formulas:

$$T_1 = \frac{T_A \times (C_B - C_A) - C_A \times (T_B - T_A)}{C_B - C_A} \tag{6.5}$$

$$T_2 = \frac{T_B \times (C_C - C_B) - C_B \times (T_C - T_B)}{C_B - C_C} \tag{6.6}$$

Finally, the equation of the temperature interval width of the heat capacity peak is:

$$\Delta T_{cv} = T_2 - T_1 \tag{6.7}$$

In this work, the following notations are used: $\Delta T_{cv_{H1}}$, $\Delta T_{cv_{H2}}$ are the values of the temperature interval width of heat capacity peak under the external magnetic fields H_1 and H_2, calculated from the Hamiltonian model output arrays $Cv_{H1}(T)$ and $Cv_{H2}(T)$ respectively.

Curie Temperatures: T_c

The Curie temperature, T_c, is a phase transition temperature and it corresponds to the maximum value of heat capacity (Cv) from the array of the temperature and the magnetic field dependency of heat capacity ($H-T$ surface). In this work, the following Curie temperatures are taken into account:

- $T_{c_{cool}}(H_1)$, $T_{c_{cool}}(H_2)$: The Curie temperature under different magnetic fields (H_1 and H_2) upon cooling process (from arrays $Cv_{H1}(T)$ and $Cv_{H2}(T)$) ;
- $T_{c_{warm}}(H_1)$, $T_{c_{warm}}(H_2)$: The Curie temperature under different magnetic fields (H_1 and H_2) upon warming process (from arrays $Cv_{H1}(T)$ and $Cv_{H2}(T)$);

Curie Temperature Field Dependency: ΔT_c

The value of the field dependency is calculated as the difference between the Curie temperatures under different magnetic fields upon the cooling and warming processes:

$$\Delta T_{c_{cool}} = T_{c_{cool}}(H_2) - T_{c_{cool}}(H_1) \tag{6.8}$$

$$\Delta T_{c_{warm}} = T_{c_{warm}}(H_2) - T_{c_{warm}}(H_1) \tag{6.9}$$

Hysteresis Loss: ΔT_{Hyst}

The values of hysteresis loss are found as the difference between $T_{c_{warm}}$ and $T_{c_{cool}}$ for each magnetic field:

$$\Delta T_{hyst_{H1}} = T_{c_{warm}}(H_1) - T_{c_{cool}}(H_1) \tag{6.10}$$

$$\Delta T_{hyst_{H2}} = T_{c_{warm}}(H_2) - T_{c_{cool}}(H_2) \tag{6.11}$$

6.2.2 Software Implementation

For the experimental validation, the proposed method is implemented as an EASEA-based optimization tool, where the C++-based code of the Monte Carlo Solver with the Blume–Emery–Griffiths–Ising (BEG-I) model, the post-processing unit and the equation(s) of optimization function(s) are included in the *evaluator* part of the problem specific file of the EASEA platform. An example of the problem specific file definition is provided in Appendix A.1. For more details, one can see Section 6.1.2.

The structure of the software implementation is schematically depicted in Figure 6.5. This figure details the programming relation between the two main parts of the method: (i) the optimization core of the EASEA platform: e.g., the FastEMO algorithm, which is shown in the left side of the picture, and (ii) the EASEA CPU-parallel evaluator with the BEG-I simulation model is depicted in the right side.

In fact, this structure is an example of the method implementation: any other algorithms from the EASEA platform, e.g., QAES, QIU-NSA, single-objective Genetic Algorithm, NSGA-II, NSGA-III and etc., can be easily applied instead of FastEMO as an

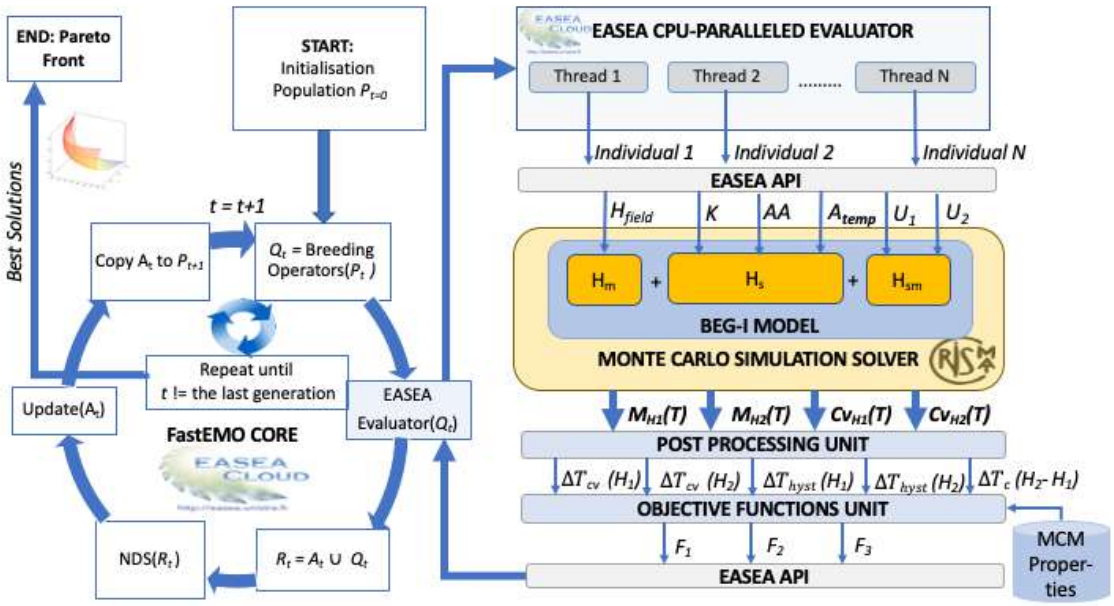


Figure 6.5: Structure of the method, implemented in the EASEA-based optimization tool.

optimization core of this method. The configuration of the problem can be also easily changed: e.g., the number of objective functions, F, can vary from one to many, according to the conflicting parameters of interest.

Furthermore, the Hamiltonian model can be modified without changing other parties of the code, due to very flexible EASEA-based structure. An example of the problem specific file definition for three-objective optimization problem, as shown in Figure 6.5, is provided in Appendix 21.

The following steps explain how this software performs:

1. The selected optimization algorithm, e.g., FastEMO (*FastEMO CORE* in Figure 6.5), starts by randomly generating the initial set of solutions, where each solution consists of free parameters of the Hamiltonian defined in Table 6.6.

2. The main cycle of the optimization algorithm is executed and is repeated until the last generation is achieved (see Figure 6.5, left part).

3. The optimization algorithm creates the new child population Q_t from the parent population P_t by breeding operators: the SBX crossover and the Cauchy-based mutation operator.

4. The *EASEA Evaluator* executes several threads with *Monte Carlo Simulation Solvers* for solving BEG-I model in parallel (see the right part of Figure 6.5).

5. Each executed solver works out the *BEG-I Model* code with its own combination of free parameters (Individuals in Figure 6.5), obtained via *EASEA API*.

6. Each *Monte Carlo Simulation Solver* returns the output values, the temperature and magnetic-field dependence of magnetization ($M(H,T)$) and of heat capacity ($Cv(H,T)$), to the *Post-Processing Unit*.

7. The *Post-Processing Unit* calculates the parameters defined in Section 6.2.1.1, which allow the program to compute the objective function(s) according to the specific requirement of the user.

8. *Objective Functions Unit* performs the evaluation of the defined number of objective functions from the post-processing data for each individual.

9. Each evaluated individual (potential solution) is sent back to the optimization algorithm via the *EASEA API*.

10. The optimization algorithm performs the current generation and goes to the step 3.

As long as the defined maximum number of generations was not reached, the loop of 8 steps (from 3 to 10) repeats.

6.2.3 Experimental Validation

In this section, we present an experimental study.

Experimental Objectives:

1. Validate the efficiency of the proposed method for finding a set of free parameters of the model, which corresponds to the physical properties of a given Magneto Caloric Material (MCM). For this purpose, the following MCMs are selected:

   (a) one Second Order phase Transition SOT (MCM): $LaFeCoSi$ ($LaFeSiA$);

   (b) two First Order phase Transition (FOT) MCMs: $GdSiGe$ and $LaFe_{13-x}Si_x$ ($LaFeSiB$).

   If the experimental results confirm an ability of the proposed method to find a set of free parameters of the Hamiltonian for the selected materials, which ensures an accurate simulation result of the model, this method can be employed for many different MCMs for further performing a numerical simulation of their properties. The latter is important, because the numerical simulation of the physical properties of MCMs can be a rich source of information.

2. Confirm the applicability of the following developed algorithms in the frame of the proposed method: FastEMO, QAES, QIU-NSA.

Algorithms and Parameters:

For a comparative study, we use the following optimization algorithms:

1. Single-objective algorithms: QAES, QIU-NSA.

2. Multi-objective algorithms: FastEMO.

The C++-based versions of the mentioned above algorithms are open-source and publicly available in the EASEA platform, version 2.20.

All algorithms are employed with their default parameter settings. One can find the default parameter settings for QIU-NSA in Tables 5.10, for QAES in Tables 5.4 and for FastEMO in Tables 4.5.

Experimental Procedure:

To reach both objectives, we use the same experimental procedure for all experiments, which consists of the following three steps for each material:

1. Employ the proposed method with the selected algorithm for finding the optimum combination of free parameters for the Blume–Emery–Griffiths–Ising (BEG-I) model w.r.t. the physical properties of interest.

2. Run the Monto Carlo simulation for solving the BEG-I model with the obtained free parameters of the Hamiltonian in order to reproduce the curves of magnetization, of heat capacity and of magnetic entropy change.

3. Compare the obtained curves of magnetization, of heat capacity and of magnetic entropy change with those retraced from the measured data provided by the *Crismat* laboratory.

Equations of Objectives:

According to the proposed method, the objective function(s) is/are defined as the minimization of differences between reference (measured or desirable) physical properties of Magnetoc Caloric Material (MCM) and those, calculated by the post-processing unit (see Figure 6.3). The number of objectives are defined according to the required physical properties of the MCM.

The possible objectives, which are used in these experiments in different combinations, are presented below:

1. The sum of differences between simulated and reference values of ΔT_{cv} under H_0 and H_1 upon cooling process:

$$\begin{aligned} F_1 = &\left|(\Delta T_{cv_{cool}Ref}(H_0) - \Delta T_{cv_{cool}}(H_0))\right| + \\ &\left|(\Delta T_{cv_{cool}Ref}(H_1) - \Delta T_{cv_{cool}}(H_1))\right| \end{aligned} \tag{6.12}$$

2. The sum of differences between simulated and reference values of ΔT_{cv} under H_0 and H_1 upon heating process:

$$\begin{aligned} F_2 = &\left|(\Delta T_{cv_{warm}Ref}(H_0) - \Delta T_{cv_{warm}}(H_0))\right| + \\ &\left|(\Delta T_{cv_{warm}Ref}(H_1) - \Delta T_{cv_{warm}}(H_1))\right| \end{aligned} \tag{6.13}$$

3. The sum of the differences between simulated and reference value of ΔT_c under H_0 and H_1 upon cooling process:

$$\begin{aligned} F_3 = &\left|(\Delta T_{c_{cool}Ref}(H_0) - \Delta T_{c_{cool}}(H_0))\right| + \\ &\left|(\Delta T_{c_{cool}Ref}(H_1) - \Delta T_{c_{cool}}(H_1))\right| \end{aligned} \tag{6.14}$$

4. The sum of the differences between simulated and reference value of ΔT_c under H_0 and H_1 upon heating process:

$$\begin{aligned} F_4 = &\left|(\Delta T_{c_{warm}Ref}(H_0) - \Delta T_{c_{warm}}(H_0))\right| + \\ &\left|(\Delta T_{c_{warm}Ref}(H_1) - \Delta T_{c_{warm}}(H_1))\right| \end{aligned} \tag{6.15}$$

In our preliminary research, we investigated some other possible objective functions. Finally, in the current version, the listed above objectives are taken under consideration, where the different combinations of these objectives are possible, depending on the studied properties of the MCM.

At the beginning of the research, we had no intuition whether some of the objectives were conflicting w.r.t. each other or not. By our preliminary experiments with multi-objective algorithms, i.e., NSGA-III and FastEMO, we revealed that objectives F_1 and F_2, as well as F_3 and F_4 are non-conflicting and can be aggregated to reduce the number of objectives. In this case, when only two non-conflicting criteria are included in the configuration of the problem, an application of multi-objective algorithms is possible, but not very meaningful, because all objectives can be aggregated.

Experimental Settings:

The general parameter settings are summarized in Table 6.1, Table 6.2 and Table 6.3.

The experimentally measured parameters for each MCM used as reference values and provided by the Crismat laboratory are presented in Table 6.1.

Table 6.1: Reference parameters for the objective functions provided by the *Crismat* laboratory.

Parameter [K]	$T_{c_{cool}Ref}$	$T_{c_{warm}Ref}$	$\Delta T_{cv_{cool}Ref}$	$\Delta T_{cv_{warm}Ref}$
$GdSiGe$ $H_0 = 0[T]$	84.0	87.0	5.0	5.0
$GdSiGe$ $H_1 = 2[T]$	92	94	8.0	8.0
$LaFe_{13-x}Si_x$ $H_0 = 0[T]$	175.8	179.2	2.0	2.0
$LaFe_{13-x}Si_x$ $H_1 = 2[T]$	187.7	187.8	3.5	3.5
$LaFeCoSi$ $H_0 = 0[T]$	238.5	239.3	6.0	6.0
$LaFeCoSi$ $H_1 = 2[T]$	245.5	246.0	12.0	12.0

The number of decision variables of the experimental optimization problems depends on the studied material, which defines the configuration of the Hamiltonian and consequently, the number of its free parameters. In Table 6.2, the free parameters with their boundary values are detailed for each material.

Due to the fact, that the Blume–Emery–Griffiths–Ising (BEG-I) model is under active development, we conducted many tests for different configurations: (i) with different algorithms, (ii) with real/imaginary data of materials, (iii) normalized/non-normalized BEG-I model, (iv) different number of objectives, etc.

In this Section, in Table 6.3, we present the configurations, which summarize the most significant experiments from our point of view, which are conducted in this study. They are presented below. In all presented experiments, a normalised version of BEG-I model is used.

Table 6.2: Boundary values of free parameters of the Hamiltonian for different materials.

Parameter	$GdSiGe$	$LaFe_{13-x}Si_x$	$LaFeCoSi$	Unit
H_{field}	0.1 - 2.5	0.1 - 2.5	0.1 - 2.5	[T]
K	0.1 - 2.0	0.1 - 2.0	0.1 - 2.5	[-]
U_1	0.1 - 2.0	0.1 - 2.0	0.1 - 2.5	[-]
U_2	0.1 - 2.0	0.1 - 2.0	$= U_1$	[-]
AA	-	0.0 - 1.0	-	[-]
A_{temp}	-	0.0 - 2.0	0.0-2.5	[-]

Table 6.3: Configurations of the experiments.

Experiment	MCM	Algorithm	Obj.	Pop. size	Gen. nb.
Exp. 1	$LaFe_{13-x}Si_x$	FastEMO	F_1, F_3+ F_4	1000	12.5
Exp. 2	$GdSiGe$	QAES	$F = F_1 + F_2$	[2,50]	250
Exp. 3	$LaFeCoSi$	QIU-NSA	$F = F_1 + F_2$	4	3125

We set the overall computational budget to 12500 function evaluations, trying to select the most appropriate population size for each algorithm, taking into account its specific properties: e.g., for FastEMO, the population size is relatively large in order to exploit its scalability w.r.t. the population size and to achieve a speedup by reducing the number of generations.

The results of each experiment are evaluated in terms of similarity between the simulated and measured curves of magnetization $M(H,T)$, heat capacity $Cv(H,T)$ and entropy change $\Delta S(T)$.

In order to investigate the impact of the optimization algorithm on performance of the proposed method, we conduct the first three experiments for FOT MCM - $LaFe_{13-x}Si_x$ and the last three experiments for SOT MCM - $LaFeCoSi$, by using single and multi-objective optimisation algorithms (see Table 6.3).

6.2.3.1 Experimental Results

Experiment 1: FastEMO runs to reproduce the properties of $LaFe_{13-x}Si_x$

First, we employ the multi-objective algorithm developed in the frame of this thesis, FastEMO, to reproduce the properties of First Order phase Transition (FOT) Magneto Caloric Material (MCM) $LaFe_{13-x}Si_x$.

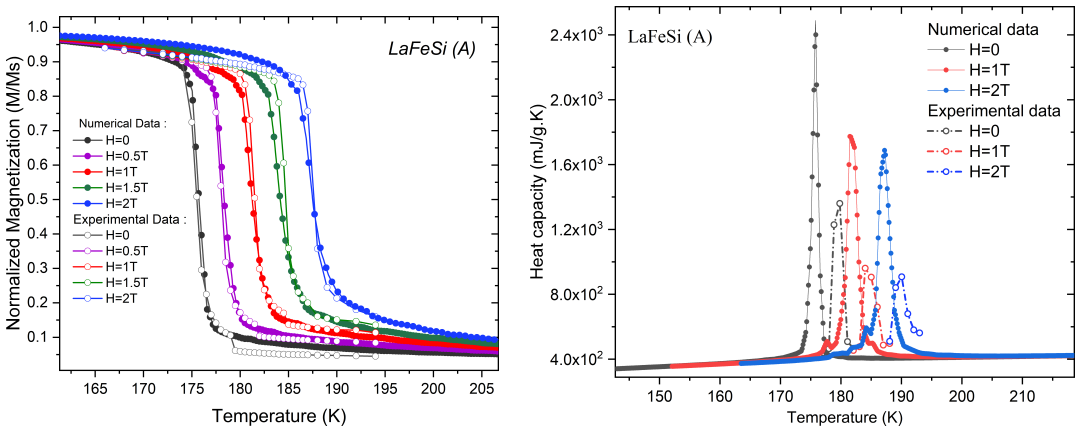
Since F_3 and F_4 objectives are non-conflicting we aggregate them and formulate the problem as bi-objective problem: $m = 2$. According to the free parameters of the Hamiltonian taken under consideration in this experiment and shown in Table 6.2, the optimization problem is configured as a partially separable problem with the number of decision variables $d = 6$.

The resulting Pareto Front is quite small (9 solutions), and the best balanced solution for all objectives is easily chosen, which provides the following free parameters for FOT

MCM $LaFe_{13-x}Si_x$: $H_{field} = 1$ [T], $K = 1.419$, U_1=0.827, U_2= 1.562, AA=0.255, A_{temp}= 0.259.

In order to validate the results, we make a control run of the Blume–Emery–Griffiths–Ising model (BEG-I) simulation with the selected combinations of free parameters.

Figure 6.6 and Figure 6.7 depict the results of the control simulation.



a) b)

Figure 6.6: $LaFe_{13-x}Si_x$: the BEG-I simulation (numerical) (black symbols) and experimental (open symbols) a) curves of temperature dependence of magnetization and b) curves of temperature dependence of the heat capacity.

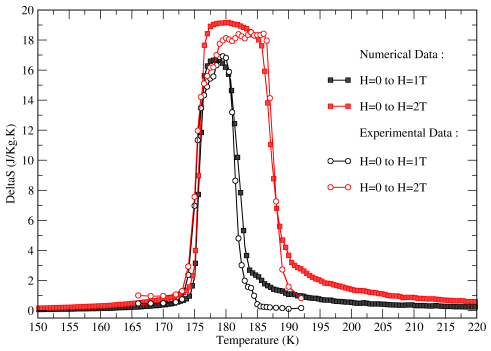


Figure 6.7: $LaFe_{13-x}Si_x$: The BEG-I simulation and experimental temperature dependence of magnetic entropy change.

Figure 6.6 (a) represents the temperature and magnetic field dependence of the simulated and experimentally measured magnetization at magnetic field 0T, 0.5T, 1T, 1.5T, 2T upon warming. The overlapping experimental magnetization curves together with the theoretical curves are very similar with an average error of $\pm 0.3K$.

Figure 6.6 (b) shows the temperature and magnetic field dependence of the heat capacity upon warming at 0T, 1T and 2T. We observe a shift between the experimental and theoretical heat capacity peaks. We suppose that the origin of this shift can be connected with the problems of the experimental measurement of heat capacity in the case of FOTs (in our study case, it is measured by the semi-adiabatic relaxation method by the *Crismat* laboratory). During the temperature stabilization period around FOT, part of the latent

heat may be consumed, and will not be taken into account during the measurement. The value of the peak of heat capacity is considered as an unreliable value and the differences in this value among simulated and measured data are not taken into account.

In Figure 6.7 we show the temperature dependence of magnetic entropy change, calculated according to Maxwell's equation, in the magnetic field ranges of $0-1$T and $0-2$T. We see a good agreement between the simulated and experimentally measured data: the mean absolute percentage error MAPE for ΔS is 0.2%.

To summarize, the obtained free parameters of the Hamiltonian by FastEMO allow us to simulate the curves of temperature dependence of magnetization, temperature dependence of heat capacity and temperature dependence of magnetic entropy change for $LaFe_{13-x}Si_x$ material, which are in a good agreement with those from the experimental data provided by the *Crismat* laboratory.

Experiment 2: QAES runs to reproduce the properties of $GdSiGe$

Now we employ a single-objective algorithm, QAES, which was developed to solve computationally intensive separable or partially separable optimization problems. The goal of this experiment is dual: (i) to confirm the robustness and applicability of the proposed method by applying to one more First Order phase Transition (FOT) material; (ii) to confirm the ability of QAES to efficiently solve the partially separable problem of the Blume–Emery–Griffiths–Ising (BEG-I) model.

For this purpose, we conduct an experiment for another FOT material - $GdSiGe$, aimed at finding the optimal combination of free parameters of the Hamiltonian of BEG-I. According to the free parameters of the Hamiltonian taken under consideration in this experiment and shown in Table 6.2, the problem is defined as partially separable with the number of decision variables $d = 4$.

Since F_1 and F_2 objectives are non-conflicting, we aggregate them and formulate the problem as single-objective problem: $m = 1$.

The best obtained solution for $GdSiGe$ is as follows: $H_{field} = 1$ [T], $K = 0.995$, U_1= 0.525, U_2=1.00

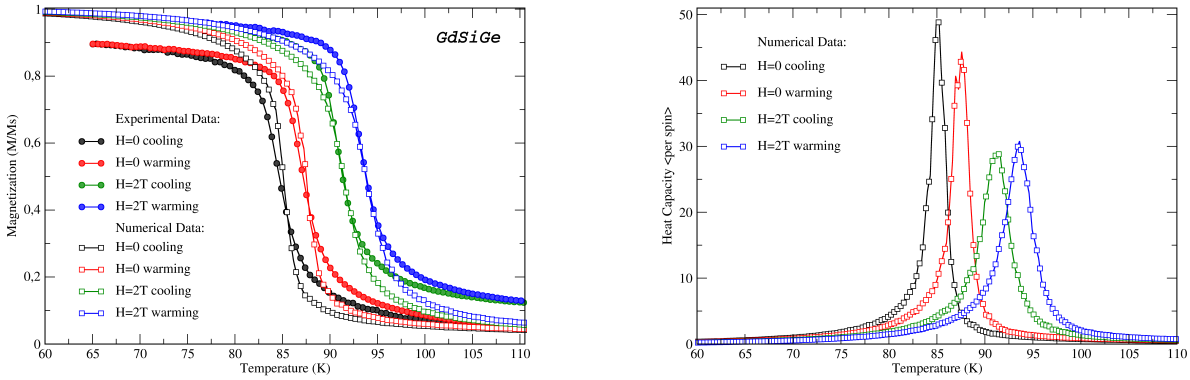
In order to validate the results, we make a control run of the BEG-I simulation with the selected combinations of free parameters.

Figure 6.8 and Figure 6.9 depict the results of the control simulation.

Figure 6.8 displays the comparison between simulated and experimentally measured magnetization and heat capacity temperature dependency for $GdSiGe$ under magnetic fields $H_0 = 0$ [T], $H_1 = 2$ [T].

As seen from Figure 6.8 (a), all curves of magnetization temperature dependency of simulated result fit the experimental values perfectly with an average error $\pm 0.32K$.

Due to the fact that the measured values of the peak of the heat capacity are not provided for $GdSiGe$ by the *Crismat* laboratory, we analyse the curves of temperature dependency of heat capacity only by the temperature interval width of this peak and the Curie temperature, which corresponds to this peak. From Figure 6.8 (b) it is clear, that the simulated behaviour of the heat capacity temperature dependency corresponds to the experimental reference parameters, defined in Table 6.1. Thus, upon the cooling and warming processes, the heat capacity peaks correspond to the reference Curie temperatures (for H_0: $T_{c_{cool}Ref} = 84[K]$, $T_{c_{warm}Ref} = 87[K]$ and for H_1: $T_{c_{cool}Ref} = 92[K]$, $T_{c_{warm}Ref} = 94[K]$) and the values of the temperature interval width of this peak are also correct with respect to the reference data in Table 6.1 (for H_0: $\Delta T_{cv_{cool}Ref} = \Delta T_{cv_{warm}Ref} = 5.0[K]$ and



a) b)

Figure 6.8: *GdSiGe*: the BEG-I simulation (black symbols) and experimental (open symbols) a) curves of temperature dependence of magnetization and b) curves of temperature dependence of heat capacity.

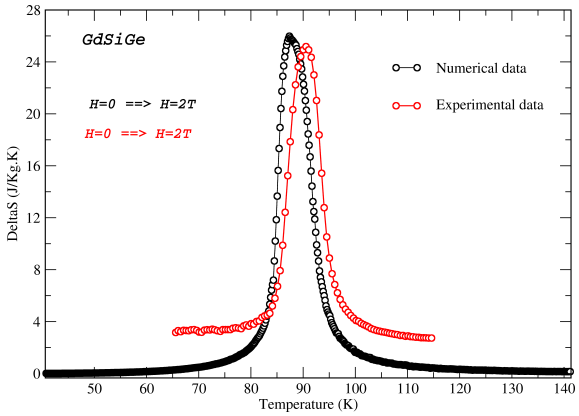


Figure 6.9: *GdSiGe*: The BEG-I simulation and experimental temperature dependence of magnetic entropy change in the magnetic field range of 0-2T.

for H_1: $\Delta T_{cv_{cool}Ref} = \Delta T_{cv_{warm}Ref} = 8.0[K]$). The trend of the heat capacity behaviour is also defined correctly: the peak value of the heat capacity decreases when the magnetic field increases and it shifts towards higher temperatures.

The temperature dependencies of magnetic entropy change, calculated from BEG-I simulation output and from experimental measurements are presented in Figure 6.9 and show a good agreement: the peak values of magnetic entropy change of experimental and simulated data are around the same and are founded near the same Curie temperature.

This experiment was repeated 3 times and the best results were found by QAES in about 500 function evaluations for all runs. The obtained result confirms the ability of QAES to efficiently solve the partially separable computationally intensive problem of the Hamiltonian model.

We can conclude that the free parameters of the Hamiltonian obtained by QAES allow to simulate the properties of FOT material ($LaFe_{13-x}Si_x$) in a good agreement with the experimental data provided by the *Crismat* laboratory.

Note that the experimental data of the temperature dependence of the heat capacity was not measured by the *Crismat* laboratory, which demonstrates the independence of our method w.r.t. experimental data and its ability to reproduce missing data.

Experiment 3: QIU-NSA runs to reproduce the properties of $LaFeCoSi$

For our last experiment we use QIU-NSA, which was developed for the unified optimization. The goal of this experiment is also dual: (i) to confirm the robustness of the proposed method to find the free parameters of the Hamiltonian for SOT materials; (ii) to confirm the ability of QIU-NSA to solve a single-objective partially separable problem of the Hamiltonian model.

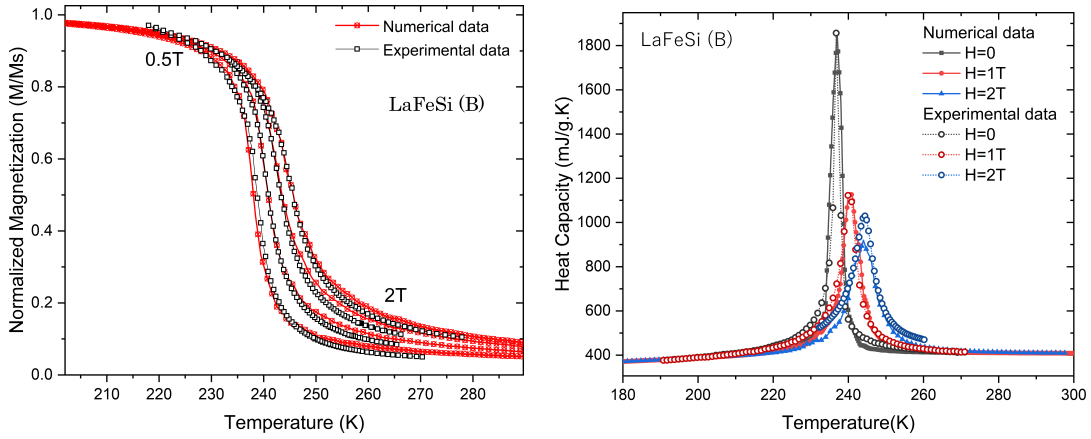
For this purpose, we conduct an experiment for the Second Order magnetic phase Transition (SOT) material - $LaFeCoSi$, aiming at finding the optimal combination of free parameters of the Hamiltonian of Blume–Emery–Griffiths–Ising (BEG-I) model. According to the free parameters of the Hamiltonian taken into consideration in this experiment and shown in Table 6.2, the problem is defined as partially separable with the number of decision variables $d = 4$.

Since the optimization criteria are non-conflicting, we can aggregate F_1 and F_2 objectives, and formulate the problem as a single-objective problem ($m = 1$).

The values of the free parameters obtained by the optimization for $LaFeCoSi$ are as follow: $H_{field} = 1$ [T], $K = 0.80$, $U_1 = 0.85$, $A_{temp} = 1.00$

In order to validate the results, we make a control run of the BEG-I simulation with the selected combinations of free parameters.

Figure 6.10 and Figure 6.11 present the results of the control simulation.



a) b)

Figure 6.10: $LaFeCoSi$: the BEG-I simulation (numerical) (black symbols) and experimental (open symbols) a) curves of temperature dependence of magnetization and b) curves of temperature dependence of the heat capacity.

In Figure 6.10 (a) we show several magnetization curves in different magnetic fields upon warming in order to follow the evolution of the intermediate magnetization shapes. The overlapping experimental magnetization curves together with the theoretical curves is very similar with an average error of $\pm 0.15K$.

In Figure 6.10 (b) we represent the heat capacity peaks in $H = 0$T, $H = 1T$, and $H = 2$T magnetic fields, compared with the experimental data in the temperature interval from 210 to 280K. Notice that the experimental heat capacity peaks are obtained from

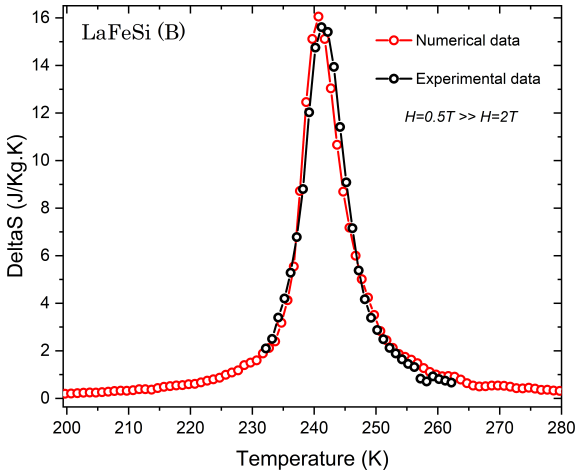


Figure 6.11: *LaFeCoSi*:The BEG-I simulation and experimental temperature dependence of magnetic entropy change in the magnetic field range of 0-2T.

a semi-adiabatic relaxation method by the *Crismat* laboratory. The simulated data fits experimental data with regard to the width and magnitude of the peaks.

In Figure 6.11, we report the temperature dependence of magnetic entropy change calculated by using the Maxwell's equation from the simulated (theoretical) and measured data. The agreement between our theoretical and experimental data is excellent: the Mean Absolute Percentage Error (MAPE) of ΔS is 0.1%.

This experiment confirms the efficiency of the proposed method to find the free parameters of the Hamiltonian for SOT materials.

The optimization process was repeated 3 times and the best results were found by QIU-NSA in about 380 function evaluations for all runs. The obtained result confirms the ability of QIU-NSA to efficiently solve partially separable computationally intensive problem of the Hamiltonian model.

To summarize the results of all experiments, we can make to the following conclusions:

1. The applicability of the proposed method for reproducing/simulating/studying the physical properties of different MCMs is confirmed: all simulation results follow very closely the measurements, provided by the *Crismat* laboratory.

2. All tested algorithms, FastEMO, QAES and QIU-NSA, can be employed in the proposed method for reproducing/simulating/studying the physical properties of different MCMs.

6.2.4 Summary and Discussion

In this section we have introduced a new optimization-based method to study and to simulate the physical properties of First Order magnetic phase Transition (FOT) and Second Order magnetic phase Transition (SOT) Magneto Caloric Materials (MCMs). The main idea of the proposed method regards the optimization of the generalized Blume–Emery–Griffiths–Ising (BEG-I) Hamiltonian simulation model of MCMs by means of finding the free parameters of this model, taking into account *a priori* known properties of MCMs.

Contribution: The key contribution of this work is to apply efficient optimization algorithms for finding the set of free parameters of the Hamiltonian of BEG-I model, the values of those are required in order to reproduce the temperature dependence of magnetization and temperature dependence of the heat capacity of desirable MCM. The proposed method is described in details and validated by the experiments with two FOT MCMs: $LaFe_{13-x}Si_x$, $GdSiGe$ and one SOT MCM: $LaFeCoSi$.

The experiments were conducted by employing three optimization algorithms developed in the frame of this thesis for single- and bi-objective problems of the Hamiltonian model.

The main advantage of the proposed method regards its generalization: it can be employed for studying and reproducing the properties of different MCMs.

In contrast to existing methods based on the Monte Carlo simulation of the Hamiltonian model [Sokolovskiy et al. (2009)], our method allows to avoid complex *ab initio* calculation. Moreover, it automates the process of finding the free parameters of Hamiltonian model by employing an optimization algorithm.

In contrast to the methods based on artificial intelligence [Maiorino et al. (2019)], [de Castro et al. (2020)], the proposed method ensures better thermodynamic coherence, due to the use of a Hamiltonian model. It does not require a huge training set.

One more positive aspect of our method is its affordability: it can be easily adapted to various Hamiltonian models, to different number of solution variables and objectives. Moreover, the software optimization tool based on the EASEA platform, makes our method user-friendly, reliable and flexible.

We assume that the proposed method is a working technique for solving the first challenge defined in the frame of the *CoolMagEvo* project (see Introduction).

Discussion: We assume that this method based on the fusion-based unified method QIU-NSA algorithm can be invariant to the complexity of the Hamiltonian model and provides scalability w.r.t. the dimension of search space if the Hamiltonian is a separable or partially separable problem. As shown in Section 5.2.2, QIU-NSA solves 100% of separable functions with the highest accuracy on Black-Box Optimization Benchmarking (BBOB) test suite for the search space dimensions $d \in [5, 640]$. Moreover, based on QIU-NSA, the proposed method provides scalability w.r.t. the dimension of objective (target) space and efficiently solves single- and multi-objective problems of the Blume–Emery–Griffiths–Ising (BEG-I) model.

Developing this idea further, we suggest that if this method is applied to a complex and detailed Hamiltonian model, it will be possible to predict the composition of alloys corresponding to the given physical properties.

Concerning the results obtained during experiments, we observe the difference of heat capacity peaks among simulated and experimentally measured data for First Order phase Transition (FOT) materials. We suppose that the comparison of heat capacity curves from simulated and experimentally measured data is only approved, and allowed in the case of Second Order phase Transition (SOT) materials. In the case of FOTs, the experimental analysis of the heat capacity peak is subject to several measurement uncertainties due to the various difficulties: the intrinsic nature of the transition, and the limit of the measuring device.

Perspectives: The experimental results allow us to hope that the proposed method together with the software optimization tool have good perspectives. It is expected, that it can be useful for investigating new commercially applicable (Magneto Caloric Materials) MCMs.

Further application of proposed method is considered for the Active Magnetic Refrigerator model of the *Ubiblue* company[6] to obtain magnetic and calorimetric data set of MCMs in concordance with experimental measurements.

In long term, two alternative perspectives are considered. The first one implies an implementation of a GPU-based version of the BEG-I model in order to exploit the scalability of FastEMO w.r.t. the population size and, consequently, to significantly improve speedup by reducing the number of generations.

The second perspective concerns quantum computing, where the Hamiltonian of the Blume–Emery–Griffiths–Ising (BEG-I) model can be implemented on quantum hardware. We will discuss this perspective in more details in Section 7.2 in the General Conclusion.

6.3 Design Exploration of Dual-Mode Operating AMR model

Since we can reproduce physical properties of different Magneto Caloric Materials (MCMs), now we proceed to explore innovative architectures of the Active Magnetic Regenerator (AMR) model, operating in the mode of a Magnetic Refrigeration System (MRS) and in the mode of a Thermo-Magnetic energy Generator (TMG). For this purpose, we turn to a many-objective optimization approach.

Optimizing the performance of the AMR model is not a new idea. As discussed in Section 2.3.2, many research works have focused on optimization, because applying optimization to numerical modeling allows to improve and accelerate the design process of an AMR by doing an efficient search of parameters, such as the geometry of the regenerator matrix, the operating frequency of the AMR etc. However, despite of various recent research in this area, commercially relevant performance for real world application has not been demonstrated yet.

From an optimization point of view, this slow progress can be explained by a lack of analysis of the AMR model as an optimization function. In contrast to related works, we, first, provide a careful analysis of the AMR model as an optimization problem, which helps to select appropriate optimization techniques. Second, we do not focus on one exactly defined problem. Instead, summarizing our previous contributions and compiling all together, we present a many-objective optimization tool with its EASEA-based software implementation, which allows users to easily set up optimization experiments independently from any modifications of the model. Thus, it is expected that an efficient optimization tool, such as the presented one, can help to make a qualitative jump in optimizing AMR models.

Moreover, to the best of our knowledge, there is no research, which simultaneously investigates the behaviour of the AMR model for two operating modes.

In this research, we optimize the performance of a multi-physics and multi-scale numerical model of AMR provided by the *Ubiblue* company, which is briefly described in

[6] `https://ubiblue.com/`

Section 2.3.3.

This Section is organized as follows: in Section 6.3.1, we analyse the AMR model and present our optimization tool developed w.r.t. the results of this analysis. In Section 6.3.2, we conduct an experimental validation of the presented tool by optimizing energy efficiency and power density of the AMR model in two operating modes: MRS and TMG, for assessing whether it is possible or not to find a common commercially applicable design.

A redacted version of this Chapter has been accepted to International Conference on Artificial Life and Evolution (ICALE) 2021.

6.3.1 Proposed Approach

6.3.1.1 Analysis of the Dual-Mode Operating AMR Model

As we explained above, in this research, we are interested in the development of a flexible instrument, which allows to explore the Active Magnetic Regenerator (AMR) model design through different experiments, rather than to solve one precisely defined problem with a fixed configuration.

For this purpose, in this section, we introduce an analysis of the AMR model in order to define and explain some general and specific considerations and requirements for the optimization algorithm. For the sake of consistent and logical storytelling, some of these considerations were already briefly presented in the introduction as the requirements to the optimization algorithms.

The AMR is a complex multi-objective system, where multiple physical phenomena, i.e., magnetic, thermal, fluid, coexist and define its performance [Risser et al. (2013)]. Consequently, for comprehensively investigating the design of the dual-mode operating AMR, at least four objectives have to be taken into account: energy efficiency and power density of the AMR model in each operating mode (a Magnetic Refrigeration System (MRS) and a Thermo-Magnetic energy Generator (TMG)). Considering that potentially more and less number of objectives can be taken taken into account, the scalability w.r.t. the target space is required. In addition, it has been shown that the relationship between the internal operating conditions and the AMR performance is not linear [Aprea, Greco & Maiorino (2017)]. Due to the non-linearity of the problem and the required number of objectives, an appropriate many-objective optimization algorithm has to be chosen for investigating multiple parameters in the design space depending on their effect on the non-linear system performance.

As the different number of common control and design parameters of the AMR can be taken under consideration, an algorithm has to ensure the scalability w.r.t. the dimension of search space.

Taking into account that the AMR model of *Ubiblue* is under active development since 2013 [Risser et al. (2013)] and its properties can be changed, we consider the model as a black-box optimization problem. Moreover, the AMR model can be defined as a dynamic optimization problem, where the optimization function on each generation depends on some environmental factors.

Our preliminary experiments show that several non-dominated solutions can be identical in the objective space and different in the search space. It means that the same performance can be obtained by different combinations of the control and design parameters. If the optimization technique does not take it into account, some of these solutions

will be lost. The loss of such kind of solutions is undesirable from an investigation point of view.

During our preliminary experiments, it also was revealed that the AMR model is an optimization problem with dominance resistant solutions. It means that some solutions can have extremely good values for some objectives and extremely bad values for other objectives. According to our preliminary tests, it is shown, that extremely good solutions are often observed in the TMG mode and extremely bad in the MRS mode. Obviously, the presence such kind of solutions degrades the search ability of the Pareto dominance operator, because the extremely good values of objective functions make the dominance resistant solutions non-dominated by other solutions.

We consider the AMR model of *Ubiblue* as a computationally intensive problem. One simulation run of a single-mode operating AMR model can take up to 15h on an AMD EPYC 7371 16-Core Processor. As the model includes two successive simulations: MRS and TMG, the overall runtime for one run of the AMR is multiplied by 2.

Taking into account all these considerations, an optimization tool to efficiently set up various optimization experiments with different number of decision variables and number of objectives is required.

In this study, we conducted various preliminary experiments in order to investigate the behaviour of the AMR model of *Ubiblue*. During these experiments the following limitations were revealed:

- source code compatibility: the source code of the AMR model of *Ubiblue* is a SCILAB code, which is not compatible with GPU parallelization;
- significant variance of the RAM size: one simulation of the current version of the AMR model can require between 0.8Gb and 24Gb of RAM;
- significant variance of the computing time: 2h - 15h.

6.3.1.2 Optimization Tool

In this section, we present an optimization tool developed to clarify the relationship between the variable control and design parameters of the AMR model and its performance in both modes. For an efficient optimization process, we compile together all our previous contributions in order to satisfy the requirements defined in Section 6.3.1.1.

Using our previous experience in the development of optimization tools, we apply again the EASEA plateform (version 2.20) as a software support in the same way as it was done for the optimization of the Hamiltonian model in Section 6.2.2, in order to provide a flexible interface, which allows users to easily adapt an optimization algorithm to the different problems and to join them together.

First, the problem of interested has to be defined in its problem specification file ($*.ez$) w.r.t. the explanation in Section 6.1.2. Thus, the $*.ez$ file includes the following attributes: the types of decision variables and objectives, the dimension of the search space and the objective (target) space, the boundaries of each variable, as well as the types of crossover and mutation operators with their hyper parameters. In order to solve the compatibility problem, the AMR model has to be called in parallel as a Linux process *via* the EASEA API from the *evaluator* part defined in the *.ez* file. Thus, each run of *evaluator* will simulate the AMR model in both operating modes, a Magnetic Refrigeration System

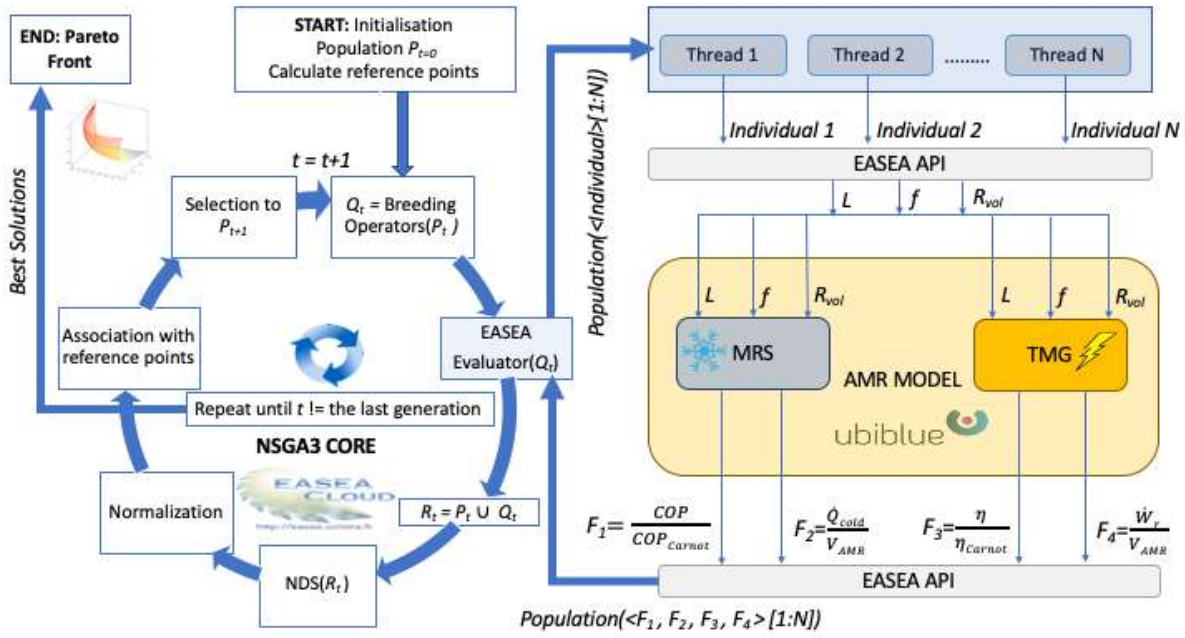


Figure 6.12: Structure of the proposed optimization tool for a dual-mode operating AMR model.

(MRS) and a Thermo-Magnetic energy Generator (TMG), with the different values of the control and design parameters.

Next, we use the EASEA compiler to automatically generate CPU parallel C++-based source code with a *Makefile* generated from the $*.ez$ file.

Last, once the code is compiled and run, the optimization tool performs w.r.t. the structure detailed in Figure 6.12. This structure of the proposed tool consists in two parts:

- a dual-mode operating AMR numerical model, provided by the *Ubiblue* company, the output parameters of which are used as evaluation functions in the optimization process (see the right part of Figure 6.12);
- a multi-objective optimization algorithm, e.g. NSGA-III, integrated to the EASEA platform(see the left part of Figure 6.12).

Aiming at tackling time consuming optimization problems, we propose to use FastEMO as an optimization core of this tool, which is presented in Section 4.2. FastEMO has been developed as a multi-objective algorithm for parallel implementation. The main feature of FastEMO is its scalability w.r.t. the population size, which is a source of speedup in terms of number of generations. However, the significant variance of the RAM size, which was revealed in the AMR model of *Ubiblue* after FastEMO had been developed, seriously restricts the number of models, which can be run in parallel. Consequently, FastEMO cannot exploit its main feature for the current version of the model, which is especially useful for large scale optimization problems.

Nevertheless, FastEMO has many other required features: (i) the technique of control the dominance area of solutions (CDAS) helps to ensure the scalability w.r.t. the dimension of the objective (target) space and solve the problems with dominance resistant

solutions (Section 4.2); (ii) the alternative crowding distance operator is used for solving the problem of non-dominated solutions, which can be identical in the objective space and different in the search space.

In this work, QIU-NSA is considered as an alternative algorithm for optimizing the AMR model. Being a fusion-based method, it employs NSGA-III in the many-objective optimization case (see the left part of Figure 6.12). NSGA-III does not use the classical crowding distance operator and can solve the problem of non-dominated solutions, which can be identical in the objective space and different in the search space, due to the niching operator. Moreover, NSGA-III performs well on many-objective functions. Thus, QIU-NSA based on NSGA-III and FastEMO are selected as many-objective optimization core in this study.

Taking into account that each simulation run of the AMR model is computationally intensive and that the same solutions can survive from generation to generation, we use a hash table to keep the already evaluated solutions, aiming at avoiding evaluating the same solutions several times. If the solution does not survive any more, the record associated with this solution is also deleted from the hash table in order to avoid memory leaks.

Hashing of solutions with the values of their objective functions is provided as an option rather than as the default EASEA behaviour because hashing is only valid when the values of the objective functions depend only upon the candidate being evaluated. E.g., this is not the case when candidates solutions are evaluated against the other members of the population. Hashing should not be used if it is possible for multiple evaluations of the same candidate solution to return different values.

The last concern regards the significant variance of computing time of the AMR model. This problem was revealed during experiments. The reason of it is the simulation time of the model with the different values of the controls and design parameters. This reason can also partially explain a presence of some memory leaks in the SCILAB code of the model, which are not fixed, because of the specific simulation way for some configurations not tested before. However, the mamory leaks has to be fixed in order to increase the population size for FastEMO. It becomes obvious that an asynchronous scheme is required [Harada & Takadama (2020)], in order to ensure a non-stop evaluation process: i.e., when a calculation in one process ends, its result is transmitted to the main part of the algorithm from the *evaluator* and the new evaluation process starts without waiting for the overall evaluation of all solutions in the generation.

In principle, this tool is versatile and can be applied to different AMR numerical models *via* the EASEA interface, i.e., the specification problem file.

6.3.2 Experimental Validation

Experimental Objectives:

- Validate the applicability of the proposed tool for solving the optimization problems of the Active Magnetic Regenerator (AMR) model of *Ubiblue*.
- Optimize the energy efficiency and power density in two operating modes: a Magnetic Refrigeration System (MRS) and a Thermo-Magnetic energy Generator (TMG), for assessing whether it is possible or not to find a common commercially applicable design.

To address these goals we present one experiment, which aims at finding the combination of parameters for an optimal AMR configuration in both modes. For this purpose, we define the optimization problem as a maximization of achievable performance by four objective functions, produced as the output of the AMR model and defined in Table 6.4, where COP is the Coefficient Of Performance, COP_{Carnot} is the Carnot Coefficient of Performance, η_{Carnot} is the Carnot yield, V_{AMR} the AMR volume ratio; $\dot{Q}_{cold}$ the refrigeration power and $\dot{W}_r$ the recovery power.

Algorithms and Parameters: In this study, we use two many-objective optimization algorithms: FastEMO (see Section 4.2) and QIU-NSA (see Section 5.2), which is NSGA-III on many-objective problems. Both algorithms are used with their default parameter settings. One can find the default parameter settings for QIU-NSA in Tables 5.10 and for FastEMO in Tables 4.5. In both cases, the AMR model is computed in parallel on CPU, where the maximum number of parallel executed models = 10, due to the high variance of the RAM size. For both algorithms, the population size is set at $N = 100$ and the maximum number of generation at $t = 50$.

Experimental Settings: Experiments are conducted with the fixed configuration parameters of the Active Magnetic Regenerator (AMR) model of *Ubiblue* defined in Table 6.5.

More precisely, Table 6.5 reports the common AMR configuration for the Magnetic Refrigeration System (MRS) and the Thermo-Magnetic energy Generator (TMG) modes, used for all simulations during the optimization process. The fixed default parameters of this configuration were chosen according to the feedback of previous *Ubiblue* prototypes and corresponds to a technico-economic optimum.

In this study, we use the data set of experimentally measured heat capacity and magnetization for several Curie temperatures for $FeSiLa$-based compounds, which are required for good energy conservation in the AMR simulations, and therefore for a reliable assessment of the energy conversion yield [Risser et al. (2012)].

Note that even though, in the current study, the numerical AMR model and the theoretical BEG-I model are not combined together, the BEG-I Hamiltonian model with the optimization-based method proposed in previous Section is employed in order to reproduce theoretical data on demand with a perfect thermodynamic consistency between heat capacity, magnetocaloric effect and magnetization on the entire Curie Temperature range required for the temperature span of the magnetic cooling system.

The experiment are conducted on an AMD EPYC 7371 16-Core Processor.

6.3.2.1 Experiment

For this experiment we select 3 control and design parameters presented in Table 6.6, as the decision variables of the optimization problem, because they can have an important influence on the AMR behaviour and its performance in both operating modes.

The value of these decision variables vary in the boundary ranges for each AMR solution.

In Table 6.6, R_{vol} and f are internal operating conditions, that can control the thermodynamic cycles of the AMR, where L presents the design parameters. The T_c of the

Table 6.4: AMR model: Optimization objectives.

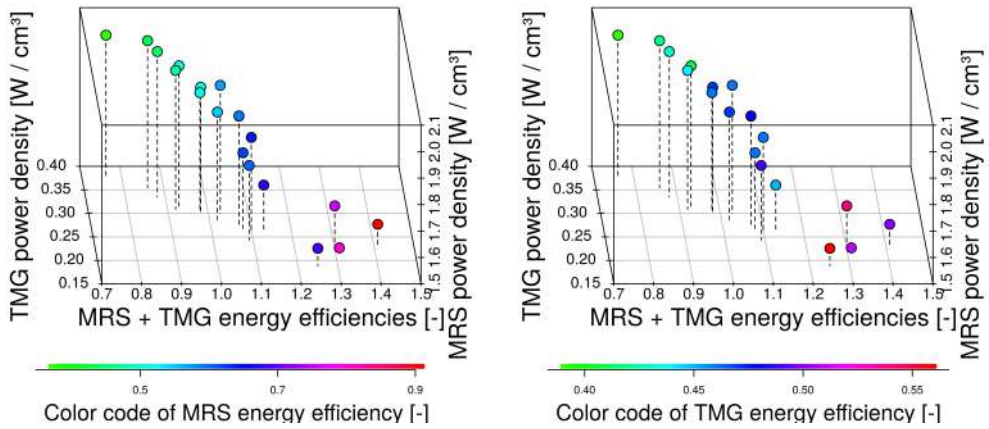
Objective	Unit	Mode	Description
$F1 = \eta = COP/COP_{Carnot}$	[-]	MRS	Energy Efficiency
$F2 = \eta/\eta_{Carnot}$	[-]	TMG	Energy Efficiency
$F3 = \dot{Q}_{cold}/V_{AMR}$	$[W/cm^3]$	MRS	Thermal Power Density
$F4 = \dot{W}_r/V_{AMR}$	$[W/cm^3]$	TMG	Mechanical Power Density

Table 6.5: Default parameters of AMR model configuration for both operating modes.

Parameter	Value	Unit
Geometry type	parallel plates	[-]
MCMs data set	$FeSiLaMnH$-based	[-]
Plate thickness	0.4	[mm]
Fluid channel thickness	0.15	[mm]
Magnetic field change value	0 - 0.8	[T]
Temperature span	$0 - 65$	$[^\circ C]$
T_c segment layering length	1	[cm]

Table 6.6: Variable (design and control) parameters of AMR configuration for both operating modes.

Parameter	Boundaries	Unit	Description
L	1-30	[cm]	AMR length along the direction of fluid motion
R_{vol}	0.05-1.5	[-]	Ratio of coolant volume transferred at each half AMR cycle on AMR fluid volume
f	0.1-10	[Hz]	AMR operating frequency



a) MRS energy efficiency in color b) TMG energy efficiency in color

Figure 6.13: Pareto Fronts.

cascade is selected relatively to the value of L within the value of the T_c segment length limitation (see Table 6.5).

Results:

The Pareto Front with non-dominated solutions obtained by NSGA-III is similar to that obtained by FastEMO. For the sake of avoiding repetition, we present the Pareto Front obtained by FastEMO. Figure 6.13 (a) and (b) depicts the Pareto front of the non-dominated solutions, showing in highlighted colour the values the MRS and TMG operating modes respectively. The Pareto front clearly reveals the conflict between the power densities and efficiency in both modes.

A parametric study is conducted to investigate the effects of the variable parameters on the cycle performance through power density and efficiency. To make it clear, Figure 6.14 reports the distribution of the Pareto-optimal solutions:

- Energy Efficiency (TMG): from Figure 6.14 (a) it seen that its maximum value $= 0.56$ $[-]$ is obtained for the following combination: $L = 17.4$ $[cm]$, $f = 0.69$ $[Hz]$ and $R_{vol} = 0.15$ $[-]$. One can notice that the TMG efficiency range is quite small $0.39 - 0.56[-]$. It can be explained by the fact that the pressure drop is directly proportional to L and consequently, a larger value of pressure drop greatly penalizes the energy efficiency in TMG mode.

- Recoverable Mechanical Power Density (TMG): Figure 6.14 (b) shows that its maximum value $= 0.37$ $[W/cm^3]$ is achieved with the following combination: $f = 1.45$ $[Hz]$ (which is a slightly larger than for the TMG energy efficiency), but the values of the other two parameters are almost the same: $L = 17.2$ $[cm]$, $R_{vol} = 0.15$ $[-]$. Thus, an increase of f leads to increasing the mechanical power density and efficiency decrease.

- Energy Efficiency (MRS): its maximum value $= 0.56$ $[-]$ is obtained with $L = 13.6$ $[cm]$, $R_{vol} = 0.11$ $[-]$ and $f = 1.18$ $[Hz]$ (see Figure 6.14 c). We notice that the value of L is smaller comparatively with the TMG mode, but the changing of the

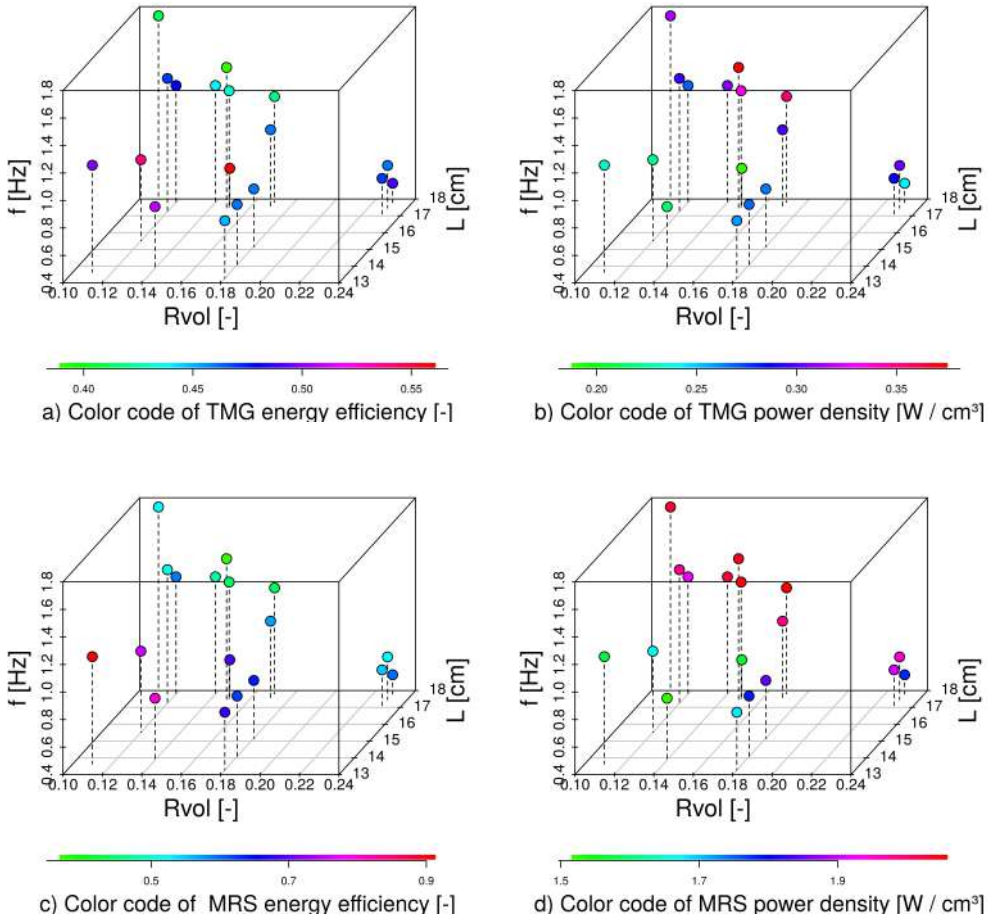


Figure 6.14: Distribution of decision variables in Pareto optimal solutions.

value of R_{vol} is not significant. the frequency is in 1.7 times higher than for the maximum of TMG energy efficiency.

- Thermal Power Density (MRS): Figure 6.14 d) shows that the maximum value = $2,05$ $[-]$ is achieved with the value of $L = 17.8$ $[cm]$ and $R_{vol} = 0.11$ $[-]$, which are almost the same as in the TMG mode, and a slightly smaller value of $f = 1.17$ $[Hz]$.

We can conclude that for all performance values, R_{vol} has less impact than the others. Also, we notice that according to Figure 6.14, there are several solutions, which ensure the balance between efficiency and power density for both modes.

From this experiment, we can also conclude that this tool allows users to obtain detailed information about the relationships between variable parameters and performance of the AMR.

The presented results allow us to confirm the applicability of the proposed tool for solving the different problems of the AMR model.

In this work, we do not estimate the speedup of the proposed tool, because of the small number of possible parallel runs, due to the high variance of the RAM size. Consequently, currently, we cannot use the FastEMO with large population sizes and obtain a significant speedup by reducing the number of generations.

6.3.3 Summary and Discussion

In this section, a user-friendly optimization tool based on the EASEA platform is presented. This tool is developed to study an influence of the control and design parameters of a Active Magnetic Regenerator (AMR) model on its performance in the mode of the Magnetic Refrigeration System (MRS) and a Thermo-Magnetic Generator (TMG).

The convenient interface allows users to quickly set up experiments and to obtain results in terms of the Pareto front of the non-dominated solutions.

Another positive aspect of the proposed optimization tool is its universality, because it can be employed for any other numerical models of AMR and can be easily adapted to different number of decision variables and objectives. Moreover, thanks to the EASEA API, the AMR model can be called as an executable file or as a C++ function.

The conducted experiment validates that a common design of the AMR, which ensures the balance between the efficiency and the power density can be found for the both modes. But further experiments are required for taking a larger number of parameters into account.

Discussion: Regarding the Pareto front, one should notice that it has a slightly discontinuous shape. Partially, this can be explained by a large number of rejected solutions, because of their nonexistence in the both modes simultaneously. Consequently, an evaluation of the energy conversion system is required by reconsidering some default parameters of the AMR model, e.g., the fluid channel thickness, which was set to optimally match a refrigeration system in this study.

The proposed tool can accelerate the design process of the AMR model by finding the common combination of the control and design parameters for the MRS and the TMG modes. Thus, it can accelerate the elaboration of a commercially available device, which will correspond to modern ecological and energy-saving requirements.

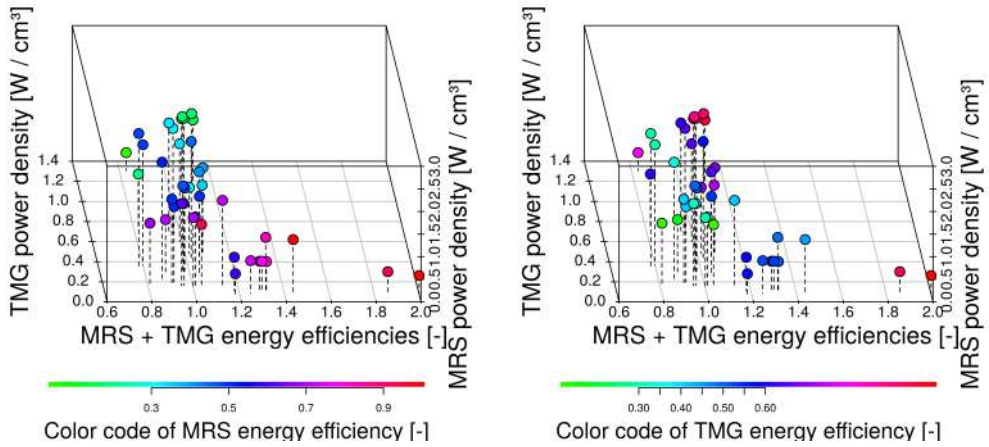


Figure 6.15: Pareto Fronts.

Perspectives: In a long term perspective, a programming language migration of the AMR model source code from SCILAB to C++ should be done, so that GPU-parallelization can be supported. In future work, a GPU-version of the AMR model will be joined with GPU-version of FastEMO through the EASEA platform in order to exploit the scalability of FastEMO w.r.t. the population sizes of the magnitude of 10000 individuals (potential solutions) and to obtain a high speedup by solving a problem in several generations.

In a short term perspective, the proposed tool has to solve dynamic problems, when the number of decision variables changes over time and depends on the result, obtained during previous generations.

Explicating this idea further, we are going to investigate the impact of a change of Curie temperature (T_c) cascade together with other design and control parameters on the performance in both operating modes simultaneously. In order to do this, the segments of T_c cascade must be a part of the solution variables of the optimization problem. They should be defined as a vector with varying size that depends on a value of another solution variable: i.e., the length of AMR (L) obtained on previous generation ($t-1$) of the evolutionary algorithm. It means that we have to deal with variable-length multi-objective optimization problem, which is intuitively difficult since the solutions from different search space have to share a common target space.

In our preliminary research, we fixed the size of the T_c segments of the cascade as the maximum possible size w.r.t. to the maximum value of L. Then, we use a basis spline function, which gives us the value of a typical thermal gradient of the T_c cascade: it helps to keep the T_c vector size fixed. Then, for each value of length (L), this function is sequenced according to its number of T_c segments. In addition to R_{vol}, f and L, we include the following decision variables: 9 segments of T_c cascade, plate thickness and fluid channel thickness. Figures 6.15 (a) and (b) depict the obtained Pareto front, showing in highlighted colour in the MRS and TMG operating modes respectively.

From Figure 6.15 we again observe a discontinuous shape of the Pareto front, but in contrast to the first experiment we obtain more solutions.

This preliminary experiment confirms the usefulness of the proposed tool for different research cases, however a dynamic optimization approach has to be investigated in order to solve more sophisticated optimization problems. E.g., it would be interesting to have

the possibility to find the solutions in the two modes by two search sub-spaces: one of them is common search space for both modes and the another one is different (i.e., the values of R_{vol} and f are found in different ranges).

Another short term perspective is to make more tests of the asynchronous scheme of CPU-parallelization and make it publicly available in a new version of the EASEA platform.

Part IV

General Conclusion

Chapter 7

Conclusion

7.1 Summary of Contributions

In the course of the presented thesis, the following scientific and practical results were achieved:

1. **The research and development of an evolutionary archive-based optimization algorithm, FastEMO, has been carried out**.

   FastEMO is designed to solve in parallel complex computationally intensive multi- and many-objective optimization problems of the Active Magnetic Regenerator (AMR) model provided by the *Ubiblue* company, but can be also applied to solve others real-world problems. The description of FastEMO is presented in Section 4.

   The main feature of FastEMO is the scalability w.r.t. the population size: i.e., an ability to find the optimal Pareto front by working with a large population size in a small number of generations. This feature is necessary for a parallel implementation, because the population scalability allows the algorithm to achieve a significant speedup, in terms of the overall runtime, especially for solving computationally intensive problems, when objective functions are evaluated in parallel.

   The experimental validation on different test suites confirmed that FastEMO provides the required population scalability and the scalability w.r.t. the dimension of the search space. However, its scalability w.r.t. the number of objectives has to be further improved.

   FastEMO has been successfully applied to solve the optimization problems of the AMR model and the model of Magneto Caloric Materials (MCMs).

2. **A quantum-inspired algorithm, QAES, based on the Diffusion quantum Monte Carlo (DMC) method with integrated the (1+1)-Evolution Strategy (ES), has been proposed, developed and investigated.**

   the Quantum-inspired Algorithm with Evolution Strategy (QAES) is designed to solve computationally intensive single-objective, low-dimensional separable and partially separable problems of the model of MCMs provided by the *Crismat* laboratory. The description of QAES is presented in Section 5.1.

   The motivation for this study is explained by the expediency of using a subject-oriented research tools for solve the problems of the model of MCMs, which has

a quantum nature. In this context, the Diffusion quantum Monte Carlo (DMC) method is a good candidate, since it can approximate the ground state energy for any kind of quantum systems.

The analysis of the DMC method as a baseline technique for an optimization algorithm has been carried out and its limitations to solve optimization problems have been found out. In order to overcome the detected issues, QAES has been proposed by integrating the (1+1)-ES algorithm in the structure of DMC.

The main feature of QAES is an ability to adjust a current number of solutions in each generation depending on the problem landscape, due to the control mechanism of the fluctuation of the population size of the DMC method. This feature can be useful to reduce the overall computational budget, which is necessary for computationally intensive problems. Another positive aspect in this context is that the population size does not play a role of the turning parameter of the algorithm any more, because the population size is not fixed and managed "automatically", fluctuating from generation to generation.

The experimental validation confirms:

(a) an applicability of the DMC method as a baseline technique for an optimization algorithm to solve continuous single-objective problems;

(b) an improvement of the optimization ability of the DMC method by integrating the (1+1)-ES;

(c) an efficiency of QAES to solve low-dimensional separable and partially separable optimization problems.

However, the experimental results revealed that QAES experiences the curse of dimensionality and has difficulty to solve non-separable problems, where a correlation among the decision variables exists. Despite these modest characteristics, QAES has good perspectives for further development.

3. **A unified algorithm, QIU-NSA, based on the fusion method, has been developed and explored.**

The Quantum-Inspired Unified Non-dominated Sorting Algorithm (QIU-NSA) is designed as a unified algorithm, capable to solve from single- to many-objective problems with a different number of decision variables. The description of QIU-NSA is presented in Section 5.2.

This research is caused by the necessity to simplify a process of real-world studies for a user by working with a single algorithm, which is flexible enough for solving the different optimization problems of the frequently modified models of MCMs and AMR.

In the result of the bibliographic research conducted for this thesis, it has been assumed that the required scalability can be achieved by employing the fusion method, which combines the solutions from several original algorithms with different properties in order to obtain the best results in comparison with each original algorithm separately.

Following this assumption, we scale a many-objective Non-dominated Sorting Genetic Algorithm III (NSGA-III) down, in order to solve single-objective problems by fusing its solutions with the solutions of the Quantum-inspired Particle Swarm Optimization (QPSO) algorithm. For this purpose, the QPSO algorithm was modified by integrating a control parameter of the population diversity in its structure for improving its ability to work with a very small population size.

The additional feature of QIU-NSA is a perfect scalability w.r.t. the dimension of the search space on the separable problems.

The experimental results confirm:

(a) a successful applicability of the fusion method for unified optimization;

(b) the positive impact of the integrated modifications into the structure of the QPSO algorithm on its ability to work with a small population size;

(c) a concordance of the properties of QIU-NSA with the requirements of the application domain.

4. **A software tool for solving real-world optimization problems has been developed and tested.**

This tool is developed as a flexible, user-friendly and open-source software based on the *EAsy Specification of Evolutionary Algorithms* (EASEA) platform, which allows users to easily set up, launch and conduct the experiments, aiming at solving various optimization problems of the both simulation models. The description of the proposed tool is presented in Section 6.1.

This elaboration is caused by the need of the scientists, who are studying the MCMs and AMR, to have an instrument with various optimization algorithms and a convenient user interface, in order to easily join any given problem with a suitable algorithm, quickly organise and reconstruct an experiment, obtain and compare results.

For this purpose, the EASEA platform was extended to a multi-objective optimization version: EASEA 2.20, which includes the following features:

(a) templates of state-of-the-art multi- and many-objective algorithms;

(b) templates of the algorithms developed in the frame of this thesis;

(c) an Application Programming Interface (API), which makes the proposed tool invariant w.r.t. the programming language of models;

(d) performance metrics;

(e) test suites;

(f) support of the old functionalities of EASEA.

The EASEA 2.20 platform is universal and can be employed for any kind of research in various domains.

The experimental validation confirms an efficiency and convenience of the proposed tool for solving different problems of the models of MCMs and AMR.

5. **An optimization-based method for reproducing and studying the physical properties of Magneto Caloric Materials (MCMs) has been developed and investigated.**

   This method is developed to find the set of the free parameters of a Hamiltonian model, which corresponds to physical properties of a studied MCM. In this study, the generalized Blume–Emery–Griffiths–Ising (BEG-I) model provided by the *Crismat* laboratory, is used as a Hamiltonian model. The description of the proposed method is presented in Section 6.2.

   The parameters obtained by the proposed method can be further employed in the Hamiltonian model:

   (a) for reproducing the physical properties of the required MCMs inside the Active Magnetic Regenerator (AMR) model;

   (b) for creating a database of physical properties for different MCMs.

   This research is caused by the necessity to simplify, accelerate and generalise the process of simulating the different properties of MCMs for materials scientists to easily reproduce and study them.

   In the proposed method, for each studied MCM, the optimal combination of the free parameters of the Hamiltonian model is found as a best solution by an efficient optimization algorithm through the developed user-friendly tool with a comfortable interface, where the differences between the required and calculated values of the properties of material are used as the objective function(s).

   The experimental validation confirms:

   (a) an effectiveness of all the algorithms developed in the frame of this thesis (FastEMO, QAES, QIU-NSA), as an optimization core of the proposed method: the mean absolute percentage error of the magnetic entropy change (ΔS) in about 0.2% by all algorithms;

   (b) an efficiency of the proposed method for finding an optimal combination of the free parameters of the Hamiltonian model for MCMs with First Order phase Transition (FOT) and with Second Order phase Transition (SOT);

   (c) an applicability and comfortability of the proposed method for conducting various experiments for different materials;

   (d) a possibility of reproducing the curves of temperature and magnetic field dependence of the heat capacity and the curves of temperature and magnetic field dependence of the magnetisation.

   The proposed method based on the reliable developed algorithms seems to be promising and should provide an interest in the scientific community, since it is able to ensure reliable reproduction of the properties of various materials, which is necessary to obtain thermodynamic consistency in the AMR model.

6. **The design of the dual-mode operating model of the Active Magnetic Regenerator (AMR) has been investigated.**

This study aims at investigating the AMR model, provided by the *Ubiblue* company, which operates in two modes: i.e., as a Magnetic Refrigeration System (MRS) and as a Thermo-Magnetic energy Generator (TMG). The description of this study is presented in Section 6.3.

The result of the bibliographic research shows a lack of research about AMR models, operating in the MRS and TMG modes. Thus, such kind of study is necessary in order to obtain detailed information about the relationship between the values of control and design parameters of the AMR model and its performance in both modes.

In the course of the study, the complex dual-mode operating AMR model was analysed as an optimization problem. Based on the results of this analysis, a parallel many-objective optimization algorithm, FastEMO, was selected and employed through the EASEA-based software tool.

The presented study case is centred on the four following optimization objectives: the energy efficiency and the power density for both modes.

The results of the experimental validation confirm that:

(a) the proposed EASEA-based software tool is successfully applicable for investigating different configurations of the AMR model;

(b) FastEMO is efficient, as an optimization algorithm, for solving the four-objective problems with the dominance resistant solutions, which have extremely good values for some objectives and extremely bad values for the others;

(c) the common combination of optimal parameters of the AMR model, which provides a good approximation to a commercially applicable design in both modes, was determined.

(d) the information about the relationship between the control and design parameters and the performance of the AMR model was obtained.

The presented study is assumed as an example, further research is required with a larger number of the control and design parameters taken into consideration.

The listed above scientific and practical results helps to successfully solve the following research problems, which were formulated earlier in Chapter 1:

1. reproducing the physical properties of various MCMs by the proposed reliable method, which allows to guarantee thermodynamic consistency and a perfect energy conservation in the AMR model;

2. investigating an impact of different combinations of the control and design parameters of the AMR model, operating in two modes: as a Magnetic Refrigeration System (MRS) and a Thermo-Magnetic energy Generator (TMG) on the performance of the AMR model.

The scientific and practical results of this thesis have been employed in the research work of the *Crismat* laboratory and the *Ubiblue* company in the framework of the *CoolMagEvo* ANR project.

The research objective of this thesis defined in Chapter 1 is completed: new evolutionary algorithm (FastEMO) and quantum-inspired optimization algorithms (QAES, QIU-NSA) were developed, investigated and applied to solve the various problems of the Hamiltonian models of Magneto Caloric Materials (MCMs) and the complex model of Active Magnetic Regenerator (AMR). A special software tool was developed on the basis of the new version of the EASEA 2.20 platform, which was also developed in the frame of this thesis, and tested on the different test suites and real-world problems.

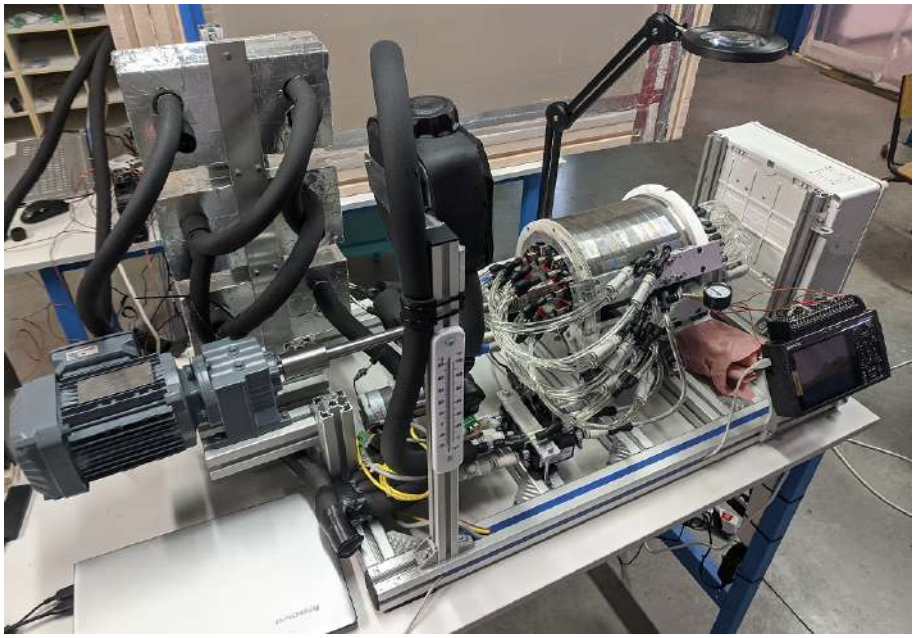


Figure 7.1: A photo of the new magnetic device prototype of the *Ubiblue* company (work in progress).

7.2 Future Directions

In this section we propose to consider several possible directions for future research, which, in our opinion, are promising. Some of these ideas have already been discussed in the corresponding chapters and sections.

7.2.1 Parallel Implementation

The presented archive-based multi-objective optimization algorithm, FastEMO, can significantly accelerate the optimization process by exploiting its scalability w.r.t. the population size. It means that FastEMO is capable to work with a population size of the magnitude of 10000 solutions in a relatively small number of generations (around 50). This property of FastEMO can be fully employed, if the objective functions are computed on GPU cards, like the RTX 2081 Ti NVIDIA card, which has 4352 cores, each running several threads.

Consequently, since the model of Active Magnetic Regenerator (AMR) is a part of the objective functions, the code of the AMR model has to be ported on the GPU cards, in order to evaluate the objective functions in parallel.

7.2.2 Problems Converter

As it was found out by the experimental study, the proposed unified fusion-based algorithm, QIU-NSA, ensures a perfect scalability w.r.t. the number of decision variables on separable problems: i.e., it is highly efficient for solving any kind of separable problems in the search space with any size.

We assume that this property of the algorithm can be extended to solve non-separable problems by converting non-separable problems into separable problems. More specifically, the main idea is to adapt the coordinate system during the search.

We assume to try one of the two following techniques in order to use it as a "problem-converter", which can make the QIU-NSA algorithm independent from any given coordinate system and consequently, can boost its optimization properties on more complex non-separable problems:

1. The Feed-Forward Neural Network based method presented in [Lu & Ito (2003)] that uses two following features:
   (a) an ability to approximate arbitrary continuous nonlinear functions with a desired degree of accuracy;
   (b) an ability to express a nonlinear function in terms of parameterized compositions of functions of single variables.
2. the Adaptive Encoding procedure based on Covariance Matrix Adaptation and presented in [Hansen (2008)]. This method allows to build a transformation of the coordinate system such that the new coordinates are as decorrelated as possible w.r.t. a given objective function. More precisely, being integrated in the structure of QIU-NSA, it can gradually lead the search toward a transformed coordinate system, where the objective function resembles more a separable function than in the original system.

Both methods can be considered as general converting methods, which can be used for the transformation of non-separable optimization problems into separable ones, even when the objective function is unknown.

7.2.3 Reinforcement technique

A natural perspective for the proposed quantum-inspired optimization algorithm, QAES, based on the Diffusion quantum Monte Carlo (DMC) method, is a reinforcement technique implementation, which allows to reinforce the QAES algorithm by an "informed" initialization scheme of the inital wave function. The natural idea is to employ the importance sampling presented in Section 3.4.2.2, in order to change the probability distribution of the particles in a controlled way by replacing a given sampling distribution with a different distribution that is more efficient, closer to a given optimization problem and provides a lower variance to the evaluation without changing the expectation value.

However, the implementation of the importance sampling technique in the context of the optimization algorithm for time-consuming problems is not obvious, as it was already discussed in this thesis, and requires further investigations. We suppose that if we succeed to use the simulated annealing algorithm as the reinforced technique, QAES would be boosted on more complex problems and demonstrate better search performance.

7.2.4 Implementation on Quantum Hardware

Since the Blume–Emery–Griffiths–Ising (BEG-I) model provided by the *Crismat* laboratory, is formulated as a Hamiltonian, the next logical step is to implement it on quantum hardware, in order to more quickly and accurately find its ground state. For this purpose, one of the existing hybrid quantum-classical algorithms has to be employed. There are several hybrid quantum-classical algorithms, which are currently used to solve a Hamiltonian in the fields of quantum chemistry and materials [McArdle et al. (2020)], [Cao et al. (2019)]. One of the most well-known example of a hybrid quantum-classical algorithm is the Variational Quantum Eigensolver (VQE) [Peruzzo et al. (2014)]. However, VQE

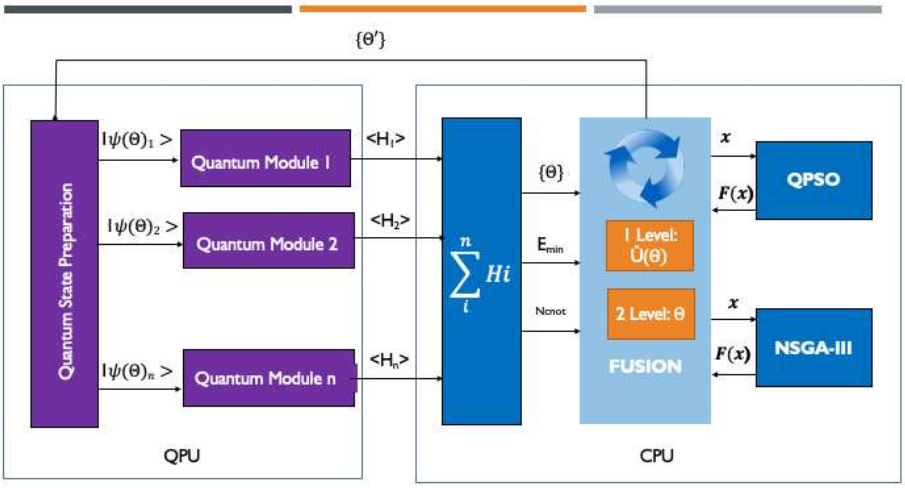


Figure 7.2: Multi-objective genetic VQE for the BEG-I model.

uses a fixed pre-constructed initial quantum circuit to solve the problem, called *ansatz*. Taking into account that the success of VQE mainly relies on this selected *ansatz*, which prepares an approximate ground state of a given Hamiltonian, the fixed structure of the quantum circuit limits the general applicability and accuracy of such algorithms. In order to efficiently find the free parameters of the Hamiltonian model, a novel noise-resilient hybrid algorithm, which will allow us to optimize the structure of quantum circuit simultaneously with the expectation value of a given Hamiltonian with a small computational budget are required.

Comparatively with the existing works, we plan to apply a more complex scheme, an approximate structure of which is presented in Figure 7.2, where the QIU-NSA algorithm is used:

1. for multi-/many-objective optimization: i.e., for finding hardware-efficient topology of a given quantum circuit by taking under consideration several conflicting criteria:

e.g., the energy of the Hamiltonian and some features of the topology of a quantum circuit, like the number of entangling operations of two qubits;

2. for single-objective optimization: i.e., for achieving a good convergence to the exact ground state energy of a given Hamiltonian.

Such a method is necessary to make a qualitative jump for Magnetic Refrigeration Systems development.

Appendices

List of Publications

National Journals

1. Abdulkarimova, U., Ouskova Leonteva, A., Rolando, C., Jeannin-Girardon, A., Collet, P. The PARSEC Machine: a Non-Newtonian Supra-Linear Supercomputer. *Azerbaijan Journal of High Performance Computing, Volume 2.*; 2019.

International Conferences

Articles

1. Ouskova Leonteva, A., Parrend, P., Jeannin-Girardon, A., Collet, P. Fast Evolutionary Algorithm for Solving Large-Scale Multi-objective Problems. *In International Conference on Artificial Evolution (Evolution Artificielle), Springer* (pp. 82-95); 2019.

2. Ouskova Leonteva, A., Abdulkarimova, U., Jeannin-Girardon, A., Risser, M., Parrend, P., Collet, P. Quantum-Inspired Algorithm with Evolution Strategy. *In International Conference on the Theory and Practice of Natural Computing (TPNC), Springer* (pp. 95-106); 2020.

3. Ouskova Leonteva, A., Hamane, R., Risser, M., Jeannin-Girardon, A., Parrend, P., Collet, P. New Evolutionary Method for Studying Physical Properties of Magneto Caloric Materials. *In International Conference on the IEEE Congress on Evolutionary Computation (IEEE CEC)*; 2021.

4. Ouskova Leonteva, Risser, M., Jeannin-Girardon, A., Parrend, P., Collet, P. A Multi-Objective Optimization Tool for Dual-Mode Operating Active Magnetic Regenerator Model. *In International Conference on Artificial Life and Evolution (ICALE)*; 2021.

Poster

1. Ouskova Leonteva, A., Risser, M., Hamane, R., Jeannin-Girardon, A., Parrend, P., Collet, P. A Hybrid Optimization Tool for Active Magnetic Regenerator. *In the Genetic and Evolutionary Computation Conference (GECCO)*; 2022.

2. Ouskova Leonteva, A., Abdulkarimova, U., Jeannin-Girardon, A., Parrend, P., Collet, P. A quantum simulation algorithm for continuous optimization . *In the Genetic and Evolutionary Computation Conference (GECCO)*; 2020.

3. Abdulkarimova, U., Santos Peretta I., Ouskova Leonteva A., Monjid Y., Amhaz R., Collet P., Rolando C. A GA for Non-Uniform Sampling Harmonic Analysis. *In the Genetic and Evolutionary Computation Conference (GECCO)*; 2020.

Abstracts

1. Ouskova Leonteva, A., Hamane, R., Risser, M., Jeannin-Girardon, A., Parrend, P., Collet, P. Evolutionary optimization of Hamiltonian model for the study of magneto-caloric materials. *In 9th IIR International Conference on Caloric Cooling and Applications of Caloric Materials Computation (THERMAG IX)*; 2020.

2. Hamane, R., Risser, M., Ouskova Leonteva, A., Hardy, V., Fast computational model for the magnetocaloric properties of *LaFeSi* materials. *In 9th IIR International Conference on Caloric Cooling and Applications of Caloric Materials Computation (THERMAG IX)*; 2020.

3. Risser, M., Lionte, S., Hamane, R., Ouskova Leonteva, A., Brochard, N., Muller, C., Numerical and experimental study of a thermomagnetic generator for energy conversion *In 9th IIR International Conference on Caloric Cooling and Applications of Caloric Materials Computation (THERMAG IX)*; 2020.

Presentations

1. Ouskova Leonteva A. From Quantum-inspired Evolutionary Algorithms to Quantum Computational Intelligence; Machine Learning for Quantum X conference, July, 2021.

International Seminars

1. Ouskova Leonteva, A., Jeannin-Girardon, A., Parrend, P., Collet, P. Fast evolutionary algorithm for multi-objective industrial problems. *International Workshop on Stochastic Local Search Algorithms (SLS)*; 2019.

Example of Problem Specific File *.ez for Finding Free Parameters of the Hamiltonian Model of MCMs

```
#define NB_VARIABLES 6    /* number of elements */
#define NB_OBJECTIVES 3   /* number of objectives */
/* Reference values */
#define DELTA_T_CV_H0_REF 3.0
#define DELTA_T_CV_H1_REF 3.5
#define DELTA_T_HYST_H0_REF 3.4
#define DELTA_T_HYST_H1_REF 0.1
#define DELTA_T_COOL_REF 11.9
#define DELTA_T_HEAT_REF 8.6
/* Boundary of search space for each element of solution */
std::vector<std::pair<TT, TT>> Boundary{
    std::make_pair<TT,TT>(lB_H,     uB_H),
    std::make_pair<TT,TT>(lB_U1,    uB_U1),
    std::make_pair<TT,TT>(lB_U2,    uB_U2),
    std::make_pair<TT,TT>(lB_K,     uB_K),
    std::make_pair<TT,TT>(lB_AA,    uB_AA),
    std::make_pair<TT,TT>(lB_Atemp, uB_Atemp)
};
/* Define the problem to be optimised */
TP m_problem(NB_OBJECTIVES, NB_VARIABLES,
                TBoundary(Boundary));

\GenomeClass::evaluator :
/* Objective Functions Unit - CPU-Paralleled Evaluator  */
        double In[NB_VARIABLES];
        double Out[NB_OBJECTIVES];

        for ( int i = 0; i < NB_VARIABLES; i++)
                In[i] = TI::m_variable[i];

        MONTE_CARLO_SOLVER(In);
        POST_PROCESSING();

        /* F1:  deltaTcv */
```

```
        Out[0] = abs(deltaTcv_H0 - DELTA_T_CV_H0_REF) +
                 abs(deltaTcv_H1 - DELTA_T_CV_H1_REF);

        /* F2:  deltaThyst */
        Out[1] = abs(deltaThyst_H0 - DELTA_T_HYST_H0_REF) +
                 abs(deltaThyst_H1 - DELTA_T_HYST_H1_REF);

        /* F3:  deltaTc */
        Out[2] = abs(deltaTc_cool - DELTA_T_COOL_REF) +
                 abs(deltaTc_heat - DELTA_T_HEAT_REF);

        for (int i = 0; i < NB_OBJECTIVES; i++)
                TI::m_objective[i] = Out[i];
        return;

\end
```

Résumé en Français de la Thèse

Introduction

Contexte

Actuellement, la demande mondiale de refroidissement augmente considérablement en raison de la croissance de la population et de l'urbanisation progressive. Selon le rapport de synthèse sur les émissions et les politiques de refroidissement du Programme des Nations Unies pour l'environnement (PNUE) et de l'Agence internationale de l'énergie (AIE), cette demande croissante de refroidissement contribue de manière significative au changement climatique et à la consommation de l'électricité. En effet, de nos jours, plus de 17% de l'électricité consommée dans le monde est destinée aux systèmes conventionnels de la production de froid [Coulomb et al. (2015)] basées sur un réfrigérant volatil (qui circule de manière continue par évaporation, compression, condensation et expansion) dans un processus de capture et de libération de la chaleur. En outre, ces systèmes conventionnels représentent un pourcentage important des émissions mondiales de dioxyde de carbone [Blowers & Lownsbury (2010)].

La réfrigération magnétique est une technologie alternative répondant à ces enjeux environnementaux et énergétiques, car elle possède plusieurs avantages par rapport aux réfrigérateurs à compression de vapeur, tels qu'une moindre pollution de l'environnement du fait de l'absence des gaz aggravant le réchauffement climatique, un faible niveau de bruit, la recyclabilité totale des matériaux et un rendement énergétique potentiellement supérieur à celui d'un réfrigérateur classique.

La réfrigération magnétique se base sur l'effet magnétocalorique (noté EMC par la suite), qui est une propriété des matériaux magnétiques et se traduit par une variation instantanée et réversible de leur température adiabatique ou l'entropie isotherme, sous l'effet de la variation d'un champ magnétique appliqué. L'EMC est maximal autour de la température de Curie qui marque le changement entre l'état ferromagnétique (moments magnétiques ordonnés) et l'état paramagnétique (moments magnétiques désordonnés) du matériau magnéto-calorique (noté MMC par la suite).

La plupart des MMCs purs (tels que le gadolinium) sont des terres rares et, par conséquence, sont très coûteux. D'autres MMCs sont développés par les métallurgistes pour répondre aux besoins commerciaux à venir avec un coût réduit et une différence adiabatique de température des matériaux plus significative, qui est insuffisante à l'heure actuelle (par exemple, pour gadolinium, cette différence de température est d'environ 3K dans un champ magnétique de 1T) [Lionte et al. (2015)]. Pour atteindre les écarts de température qui soient suffisants pour les applications usuelles, est utilisé le Régénérateur Magnétique Actif (noté AMR par la suite). L'AMR est représenté par une matrice poreuse des MMCs qui est traversée par un fluide caloporteur en mouvement alternatif, dans un

procédé cyclique.

Malgré les recherches récentes prometteuses, le développement des AMRs se heurte encore à plusieurs obstacles liées à l'estimation et simulation des caractéristiques physiques des MMCs, et aux méthodes analytiques non-efficaces, ainsi qu'à un coût de tests expérimentaux très élevé pour trouver un design du système d'AMR adapté pour les applications commerciales.

Les recherches menées dans cette thèse s'inscrivent dans le cadre du projet ANR *CoolMagEvo*, qui est axé sur un développement du système de refroidissement magnétique et de conversion thermomagnétique d'énergie à l'aide d'alliages magnétocaloriques. Le consortium est composé des participants suivants: l'entreprise *Ubiblue* - un spécialiste de la technologie du froid magnétique et le coordinateur du projet, *Crismat* - le laboratoire de sciences des matériaux et *ICube* - le laboratoire des sciences de l'ingénieur, de l'informatique et de l'imagerie.

Problèmes scientifiques

Dans le contexte décrit ci-dessus, les problèmes scientifiques sont définis comme suit:

1. De modéliser/reproduire les propriétés physiques des MMCs pour leur application dans un modèle de l'AMR ou pour la création de bases de données. Une reproduction fiable des propriétés physiques des MMCs est nécessaire afin de garantir la cohérence thermodynamique et la parfaite conservation de l'énergie dans le modèle de l'AMR.

2. D'étudier l'impact de différentes combinaisons de paramètres de contrôle et de design du modèle de l'AMR (fonctionnant en deux modes: en tant qu'un système de refroidissement magnétique et en tant qu'un générateur d'énergie thermomagnétique) sur la performance de l'AMR.

Objets de recherche

Chaque problème de recherche est en corrélation avec un modèle numérique correspondant. Par conséquence, les objets de recherche de ce projet de thèse sont les deux modèles numériques suivants qui ont un débit de calcul très élevé et qui sont en cours de développement:

1. Un modèle des propriétés physiques des MMCs (notamment la température de Curie, la dépendance de l'aimantation et la capacité thermique en champ magnétique, la largeur de l'hystérésis thermique et la forme des courbes d'aimantation) fourni par le laboratoire *Crismat*.

   Ce modèle magnétique nommé Blume–Emery–Griffiths–Ising (noté BEG-I par la suite), est décrit par un hamiltonien qui est un opérateur correspondant à l'énergie totale du système [Sokolovskiy et al. (2009)]. Pour faire une approche généralisée et pour un bon ajustement des courbes expérimentales, *Crismat* autorise un maximum de degrés de liberté dans l'hamiltonien, que nous dénoterons "les paramètres libres", où la contribution de chaque paramètre dans le modèle est utilisée pour ajuster les propriétés physiques simulés par rapport de données expérimentales d'un MMCs (notamment la capacité thermique et l'aimantation). Ces paramètres libres contrôlent la barrière énergétique des phases paramagnétiques et ferromagnétiques

et ont un impact sur la dépendance au champ de la température de Curie et sur la forme de la courbe d'aimantation. Le code du modèle est présenté sous C++.

Mais, le problème principal est une impossibilité de trouver d'une façon manuelle la combinaison de bonnes valeurs des paramètres libres permettant de reproduire les données expérimentales mesurées par *Crismat*, car ces paramètres sont différents pour chaque MMC. En plus, les simulations du modèle sont effectuées avec 10^5 cycles Monte Carlo en utilisant l'algorithme de Metropolis.

Donc, nous proposons d'utiliser une approche basée sur l'optimisation pour trouver une combinaison de paramètres libres du modèle BEG-I appropriée à chaque MMC requis.

2. Un modèle numérique multi-physique et multi-échelles de l'AMR, développé par l'entreprise *Ubiblue* [Risser et al. (2013)]. Ce modèle simule les étapes du cycle de l'AMR et évalue les performances en termes de l'efficacité énergétique et de la densité de puissance pour les deux modes fonctionnels: en tant qu'un système de réfrigération magnétique et un générateur d'énergie thermomagnétique. Le code du modèle est présenté sous Scilab.

   Mais, afin de trouver une combinaison optimale de paramètres de contrôle et de design du modèle, nous proposons une optimisation paramétrique de performance du système en ces deux modes fonctionnels.

Compte tenu de l'intensité de calcul et des multiples modifications des modèles, il est évident qu'un grand nombre de cas à rechercher ne peut être exploré à la main. Pour accélérer ce processus de recherche, une technique spéciale basée sur l'optimisation et un logiciel est nécessaire afin de résoudre les problèmes scientifiques définis ci-dessus.

Objectif de recherche

L'objectif de cette thèse est d'explorer, de développer et d'appliquer les algorithmes d'optimisation appropriés pour résoudre les problèmes scientifiques définis ci-dessus.

Sujets de recherche

Pour atteindre cet objectif défini ci-dessus, les sujets de recherche de cette thèse sont les algorithmes d'optimisation évolutionnaires et inspirés du quantique.

Un algorithme évolutionnaire met en œuvre des mécanismes inspirés de la nature et s'appuie sur les concepts de la théorie de l'évolution où les solutions jouent le rôle des organismes individuels d'une population [Collet & Rennard (2008)].

Les algorithmes inspirés du quantique utilisent les principes quantiques pour améliorer les performances, c'est-à-dire augmenter l'accélération du processus de l'optimisation et la précision de la solution finale. De plus, ils ne nécessitent pas de matériel quantique pour leur mise en œuvre et leur exécution. Donc, ces algorithmes permettent aux développeurs de tirer parti de la puissance des nouvelles techniques quantiques d'aujourd'hui sans attendre le fait que le matériel quantique soit disponible.

Pour appliquer ces algorithmes à nos problèmes, dans le cadre de cette thèse, nous avons formulé les problèmes de modélisation/reproduction de propriétés physiques des

MMCs et les problèmes de la conception du modèle de l'AMR en tant que des problèmes d'optimisation.

Chaque problème d'optimisation doit être défini par:

1. un ensemble de solutions – un espace de recherche ;
2. une ou plusieurs fonction(s), dite objectif(s), à optimiser (c'est-à-dire à minimiser ou à maximiser). Dans le cas de plusieurs objectifs il s'agit de l'optimisation simultanée de plusieurs fonctions (les critères) qui sont généralement conflictuelles ;
3. un ensemble de contraintes à respecter qui définit les conditions sur l'espace de recherche que les solutions doivent satisfaire.

La résolution optimale du problème consiste à trouver le point ou un ensemble de points de l'espace de recherche (dans le cas multi-objectif) qui satisfait au mieux la/les fonction objectif(s).

De plus, ces algorithmes doivent faire parti d'un outil d'optimisation pour automatiser le processus de couplage entre les modèles et l'algorithme d'optimisation choisi. Il est important de remarquer que des modifications de nos modèles peuvent introduire certaines erreurs de codage. Ces erreurs doivent être rapidement déboguées et corrigées. Pour réduire une zone de recherche d'erreurs, un outil d'optimisation confortable est nécessaire. Par conséquence, cet outil d'optimisation proposé dans cette thèse, nous permettra non seulement de simplifier le processus de couplage, mais aussi d'éviter les erreurs supplémentaires lors du processus de couplage après chaque modification des modèles.

Contribution

Afin de résoudre les problèmes scientifiques définis, dans cette thèse, nous avons proposé une approche basée sur les algorithmes évolutionnaires et inspirés du quantique pour l'optimisation de la performance du système de l'AMR.

Pour développer des algorithmes d'optimisation appropriés aux modèles, nous avons exploré ces deux modèles d'un point de vue de l'optimisation et considéré les caractéristiques des problèmes comme suit :

1. les problèmes d'optimisation pour le modèle des MMCs :
   - avec un ou plusieurs objectifs ;
   - à petite échelle ;
   - séparable (ou partiellement séparable) : c'est-à-dire qu'une fonction peut être exprimée sous la forme d'un produit ou d'une somme de sous-fonctions, chacune d'elles dépendant d'une seule variable.
2. les problèmes d'optimisation pour le modèle de l'AMR :
   - une boite noire dont la structure interne et le code peuvent être indisponibles, puisque les fonctions d'optimisation correspondent à un logiciel commercial ;
   - avec plusieurs objectifs (2, 3 ou > 3 objectifs) ;
   - à petite, moyenne ou grande échelle ;

- peut avoir des *Dominance Resistant Solutions* (DRS) : c'est-à-dire que certaines solutions peuvent avoir des valeurs extrêmement bonnes pour certains objectifs et des valeurs extrêmement mauvaises pour d'autres objectifs ;
- plusieurs solutions non dominées peuvent être identiques dans l'espace des objectifs et différentes dans l'espace de recherche ;
- peut être dynamique (dépend de la version du modèle qui est en cours de développement).

Au cours de la thèse présentée, les principaux résultats scientifiques et pratiques sont résumés comme suit:

1. **Un algorithme d'optimisation évolutionnaire parallèle (basé sur une archive et nommé FastEMO) a été développé**.

   FastEMO est conçu pour résoudre en parallèle des problèmes d'optimisation complexes avec plusieurs objectifs du modèle de l'AMR, mais peut également être appliqué pour résoudre d'autres problèmes d'optimisation du monde réel (qui ont un débit de calcul élevé) ayant plusieurs objectifs.

   La principale caractéristique de FastEMO est l'extensibilité (connu aussi comme la scalabilité) par rapport à la taille de la population. Il s'agit de la capacité à trouver le front de Pareto optimal en travaillant avec une grande taille de la population dans un petit nombre de générations. Cette fonctionnalité est nécessaire pour une implémentation parallèle, car cette extensibilité nous permet d'atteindre une accélération significative, en termes de temps d'exécution globale, en particulier pour résoudre les problèmes qui ont un débit de calcul élevé, lorsque les objectifs sont évaluées en parallèle. De plus, la grande taille de la population peut être efficace pour résoudre des problèmes à grande échelle, dans lesquels le modèle numérique de l'AMR peut être formulé.

   Pour atteindre cet objectif, l'algorithme présenté est dérivé de l'état de l'art *Archive-based Stochastic Ranking Evolutionary Algorithm* (ASREA) [Sharma & Collet (2010*a*)], basé sur une archive qui nous permet de réduire la complexité de calcul. FastEMO améliore la précision d'ASREA, tout en gardant la complexité de l'algorithme très petite. Une contribution clé par rapport à l'algorithme ASREA est une application d'une technique de contrôle de la zone de dominance des solutions [Sato et al. (2007)] au lieu de la dominance de Pareto conventionnelle pour améliorer l'efficacité de l'algorithme sur des problèmes si il y a plus de 3 objectifs.

   En prenant en compte que les deux modèles utilisés dans cette thèse sont en cours de développement, l'extensibilité par rapport à la dimension de l'espace de recherche et l'extensibilité par rapport à la dimension de l'espace d'objectifs sont les caractéristiques souhaitables, car ces dimensions dépendent du cas de recherche et les modifications des modèles.

   La validation expérimentale sur différentes suites de tests multi-objectifs (comme *Deb-Thiele-Laumanns-Zitzler* (DTLZ) [Deb et al. (2002)], *Walking Fish Group* (WGF) [Huband et al. (2005)] and *Black-Box Optimization Benchmarking* (BBOB) [Hansen et al. (2012)]) a confirmé que FastEMO fournit l'extensibilité requise par rapport à la population et l'extensibilité par rapport à la dimension de l'espace de recherche.

Cependant, son extensibilité par rapport au nombre d'objectifs doit encore être améliorée.

FastEMO a été appliqué avec succès pour résoudre les problèmes d'optimisation multi-objectifs du modèle de l'AMR et du modèle des MMCs (BEG-I). Par contre, afin d'utiliser les avantages proposés par FastEMO au mieux, une grande taille de la population est requise et exige des exécutions du modèle en parallèle.

2. **Un uni-objectif algorithme d'optimisation inspiré du quantique (nommé QAES) a été développé**

   QAES est un algorithme d'optimisation dérivé de la méthode quantique de diffusion Monte Carlo (DMC) [Kosztin et al. (1996)] qui permet de résoudre des problèmes uni-objectifs du modèle de l'hamiltonien de MMCs. Ces problèmes d'optimisation sont séparables (ou partiellement séparables), ont un débit de calcul élevé et une dimension d'espace de recherche à petite échelle.

   La motivation pour ce travail s'explique par la nature quantique du modèle de l'hamiltonien de MMCs où la méthode DMC parait être un bon candidat, car DMC peut approximer l'énergie de l'état fondamental pour tous les types de systèmes quantiques. De plus, la DMC se manifeste par la régulation de la taille de la population de l'algorithme en fonction du processus de diffusion. Cette caractéristique peut être utile pour les problèmes qui ont un débit de calcul élevé, car elle permet à l'algorithme de commencer par une taille de population très petite et d'augmenter ou diminuer automatiquement cette taille. Par conséquence, cela nous permet de réduire le coût du calcul global.

   Tout d'abord, une analyse de la méthode DMC en tant qu'une technique d'optimisation a été effectuée, et ses limites ont été découvertes. Afin de surmonter les problèmes détectés et d'adapter la méthode de DMC pour un processus d'optimisation, nous avons proposé d'intégrer la stratégie d'évolution (1+1) [Hansen et al. (2015)] dans la structure de DMC. À notre connaissance, la méthode de DMC n'a pas été adaptée au processus d'optimisation auparavant.

   L'avantage principal de QAES est une capacité d'ajuster un nombre actuel de solutions dans chaque génération en fonction du paysage du problème grâce à un mécanisme de contrôle de la fluctuation de la taille de la population dans la méthode de DMC. Cette fonctionnalité peut être utile pour réduire le budget d'optimisation (le nombre de fonctions d'objectif effectués), qui est nécessaire pour les problèmes intensifs en calcul. Un autre aspect positif dans ce contexte est que la taille de la population ne joue plus un rôle de paramètre de l'algorithme, car la taille de la population n'est pas fixe, fluctuant de génération en génération.

   QAES a été testé sur les suites de tests uni-objectifs BBOB [Hansen et al. (2012)] par rapport à deux algorithmes classiques (BFGS [Ros (2009)] et BIPOP-CMAES [Hansen (2009)]) et un algorithme d'optimisation par essaims particulaires quantiques (QPSO) [Sun, Feng & Xu (2004)]. Il a été démontré que QAES peut trouver le minimum global avec un coût de recherche inférieur à celui de l'algorithme QPSO.

   La validation expérimentale effectué confirme que:

   (a) la méthode de DMC est applicable pour résoudre des problèmes d'optimisation continus et uni-objectifs ;

(b) une amélioration de la capacité d'optimisation de la méthode DMC est atteinte en intégrant la stratégie d'évolution (1+1) ;

(c) QAES est efficace pour résoudre des problèmes d'optimisation séparables (et partiellement séparables) et qui ont une dimension d'espace de recherche très petite.

Cependant, les résultats expérimentaux ont révélé que QAES subit le fléau de la dimension et a des difficultés à résoudre des problèmes non séparables, où des corrélations entre les variables de décision existent. Mais malgré ces caractéristiques modestes, QAES a des perspectives prometteuses pour résoudre des problèmes d'optimisation du modèle de l'hamiltonien.

3. **Un algorithme "unifié" (nommé QIU-NSA) basé sur une technique de multiplexage a été développé.**

QIU-NSA est un algorithme "unifié" conçu pour résoudre des problèmes uni- et multi-objectifs. Le terme "unifié" est emprunté à [Seada & Deb (2014)], qui correspond à des algorithmes avec l'extensibilité par rapport à la dimension de l'espace des objectifs. Dans le cadre de cette thèse, l'extensibilité par rapport à la dimension de l'espace de recherche est aussi souhaitable.

Cette recherche est motivée par la nécessité de simplifier un processus de recherche pour des utilisateurs, qui ont besoin d'un algorithme suffisamment flexible à résoudre les différents problèmes d'optimisation (uni- et multi-objectifs) des modèles fréquemment modifiés.

Selon les résultats de notre recherche bibliographique menée dans cette thèse, il a été supposé que l'extensibilité par rapport à la dimension de l'espace des objectifs peut être obtenue en employant la méthode de multiplexage. Cette méthode sert à résoudre un plus grand nombre de problèmes, qui combine les solutions de plusieurs algorithmes aient les propriétés différentes afin d'obtenir les meilleurs résultats par rapport à chaque algorithme original séparément. Suivant cette hypothèse, nous utilisons la méthode de multiplexage pour aider l'algorithme génétique multi-objectifs (*Non-dominated Sorting Genetic Algorithm III* (NSGA-III) [Deb & Jain (2013)]) à résoudre des problèmes uni-objectifs en combinant ses solutions avec des solutions d'un algorithme d'optimisation uni-objectif - l'algorithme par essaim particulaires quantiques (QPSO) [Sun, Feng & Xu (2004)].

Dans ce travail, l'algorithme NSGA-III a été sélectionné en tant qu'une technique de base, grâce à son efficacité sur différents problèmes multi-objectifs. Mais l'algorithme NSGA-III n'est pas adapté pour résoudre des problèmes uni-objectifs [Seada & Deb (2014)] à cause les recasons suivants: (i) dans le cas d'optimisation uni-objectif, la taille de population est égale à seulement 4 individus [Seada & Deb (2014)], ce qui n'est pas suffisant pour un travail efficace des opérateurs de recominaison; (ii) une sélection de solutions au hasard [Seada & Deb (2014)].

Dans différents travaux basés sur la méthode de multiplexage, l'algorithme PSO est appliqué en raison de sa capacité d'explorer l'espace de recherche [Ibrahim, Martin, Rahnamayan & Deb (2017)]. Dans le but d'étudier l'efficacité d'un algorithme inspiré du quantique dans le cadre de la méthode de multiplexage, nous sélectionnons la version quantique de l'algorithm PSO. De plus, dans ce travail, l'algorithme QPSO

a été modifié en intégrant un paramètre du contrôle de la diversité de la population dans sa structure pour améliorer sa capacité de travailler avec une population petites (qui est défini dans NSGA-III). Techniquement, cette version de QPSO maintient la diversité dans la population, ce qui améliore les performances de NSGA-III sur les problèmes uni-objectifs sans aucune perte d'efficacité sur les problèmes multi-objectifs, car la structure NSGA-III n'est pas modifiée.

L'avantage supplémentaire de QIU-NSA est l'extensibilité par rapport à l'espace de recherche pour des problèmes séparables.

En général, la méthode de multiplexage est appliquée pour réduire le défi de choisir un algorithme approprié pour résoudre des problèmes complexes [Ibrahim, Martin, Rahnamayan & Deb (2017)] en utilisant des algorithmes d'optimisation uni- ou multi- objectifs uniquement. À notre connaissance, la méthode de multiplexage n'a pas été appliquée auparavant pour assurer l'extensibilité par rapport à la dimension de l'espace des objectifs. À cet égard, il s'agit d'une nouvelle contribution.

L'algorithme proposé est validé expérimentalement sur les fonctions uni-objectifs de BBOB [Hansen et al. (2012)], où les résultats améliorés par rapport à les algorithmes originaux (NSGA-III et QPSO) sont confirmés. De plus, QIU-NSA montre très bonne extensibilité par rapport à l'espace de recherche pour les problèmes séparables et surpasse des algorithmes de références tels que l'algorithme séparable CMA-ES [Hansen & Ostermeier (2001), Hansen et al. (2003)] et l'algorithme Limited Memory CMA-ES [Loshchilov (2014)].

Les résultats expérimentaux confirment que:

(a) la méthode de multiplexage est efficace pour assure l'extensibilité par rapport à la dimension de l'espace des objectifs ;

(b) les modifications intégrées dans la structure de l'algorithme QPSO ont un impact positif sur la capacité de l'algorithme de travailler avec une population de petite taille;

(c) les propriétés de QIU-NSA répondent à les exigences du domaine d'application.

4. **Un outil spécial pour résoudre des problèmes d'optimisation du monde réel a été développé.**

Cet outil est développé en tant qu'un logiciel flexible, convivial et open source basé sur la plateforme *EASy Specification of Evolutionary Algorithms* (EASEA), qui permet aux utilisateurs de configurer, lancer et mener facilement des expériences, visant à résoudre divers problèmes d'optimisation de nos deux modèles.

Ce travail est due au besoin des scientifiques d'avoir un instrument avec divers algorithmes d'optimisation et une interface utilisateur convivial, afin de : (i) joindre facilement n'importe quel problème d'optimisation avec un algorithme approprié ; (ii) organiser et reconstruire rapidement des expériences ; (iii) obtenir et comparer les résultats.

Pour cela, la plateforme EASEA a été étendue à une version d'optimisation multi-objectifs - EASEA 2.20, qui inclut les fonctionnalités suivantes :

(a) les *templates* de plusieurs algorithmes multi-objectifs : NSGA-II [Deb et al. (2000)], NSGA-III [Deb & Jain (2013)], ASREA [Sharma & Collet (2010*b*)], MOEA-D [Zhang & Li (2007)], IBEA [Zitzler & Künzli (2004)], CDAS [Sato et al. (2007)] ;

(b) les *templates* les algorithmes développés dans le cadre de cette thèse ;

(c) une interface de programmation d'application (API), qui permet d'effectuer les modèles des langages de programmation différentes ;

(d) les métriques de performance des algorithmes d'optimisation : *Generational Distance* (GD), *Inverted Generational Distance* (IGD) et *hypervolume*;

(e) les suites de tests : *Zitzler-Deb-Thiele* (ZDT) [Zitzler et al. (2000)]. *Deb-Thiele-Laumanns-Zitzler* (DTLZ) [Deb et al. (2002)], *Walking Fish Group* (WFG) [Huband et al. (2005)], *Unconstrained functions* [Zhang et al. (2008)], *Black-Box Optimization Benchmarking* (BBOB) [Hansen et al. (2012)] ;

(f) un support des anciennes fonctionnalités d'EASEA.

La plateforme EASEA 2.20 est universelle et peut être utilisée pour tous les type de recherche dans divers domaines.

La validation expérimentale confirme une efficacité et une commodité de l'outil proposé pour résoudre différents problèmes des modèles de MMCs et l'AMR.

5. **Une méthode, basée sur l'optimisation pour reproduire et modéliser les propriétés physiques des MMCs, a été développée.**

   Dans le cadre de la collaboration scientifique avec le laboratoire *Crismat*, nous avons développé une méthode générale pour reproduire, modéliser et étudier les propriétés physiques des matériaux magnétocaloriques (MMCs) différents. Notre motivation est expliquée par la nécessité de simplifier, d'accélérer et de généraliser le processus de simulation des propriétés des MMCs (en tant que hystérésis et l'ordre de la transition) différents pour que les scientifiques des matériaux puissent facilement les reproduire, modéliser et étudier.

   De nombreuses méthodes pour reproduire et modéliser les propriétés physiques des MMCs sont présentées dans la littérature suivante : [Sokolovskiy et al. (2009)], [**?**], [Maiorino et al. (2019)], [de Castro et al. (2020)]. Cependant, ils ont quelques sérieux inconvénients : (i) un manque de généralisation, ce qui rend les expériences très longues et coûteuses [Sokolovskiy et al. (2009)], [Sokolovskiy et al. (2010)] ; (ii) une dépendance des bases de données, qui peut limiter les recherches [Maiorino et al. (2019)], [de Castro et al. (2020)].

   Afin d'éviter les inconvénients mentionnés, nous introduisons une nouvelle méthode généralisée, basée sur l'optimisation, pour modéliser et reproduire fidèlement les propriétés physiques des MMCs. La pertinence de cette méthode réside dans son originalité, qui permet aux utilisateurs de généraliser le processus de modélisation et de reproduction des propriétés physiques pour différents MMCs. En tant que la partie fondamentale de ces problèmes d'optimisation, nous utilisons un modèle généralisé Blume–Emery–Griffiths–Ising (BEG-I) fourni par le laboratoire *Crismat*.

Plus précisément, cette méthode est développée pour trouver une combinaison de paramètres libres de l'hamiltonien du modéle BEG-I, qui permettent au modèle de reproduire fidèlement les données expérimentales d'un MMC étudié.

Même si l'approche d'ajustement du modèle aux résultats expérimentaux n'est pas nouvelle, elle n'a jamais été impliquée pour résoudre des problèmes de modélisation des propriétés physiques de MMCs.

Les paramètres libres obtenus par la méthode proposée peuvent être utilisés dans le modèle de l'hamiltonien :

(a) pour reproduire les propriétés physiques des MMCs requis à l'intérieur du modèle de l'AMR ;

(b) pour créer une base de données de propriétés physiques pour différents MMCs.

Dans la méthode proposée, pour chaque MMC, la combinaison optimale des paramètres libres du modèle de l'hamiltonien est trouvée en tant que la meilleure solution par un algorithme d'optimisation. Pour cela, un outil a été développé avec une interface confortable pour les utilisateurs, où les différences entre les valeurs calculées et les valeurs requises des propriétés du matériau sont utilisées comme fonction(s) d'objectif(s).

La validation expérimentale confirme :

(a) une efficacité de la méthode proposée pour trouver une combinaison optimale des paramètres libres du modèle de l'hamiltonien pour différents MMCs (avec la transition de phase du premier ordre et la transition de phase du second ordre) ;

(b) une applicabilité de la méthode proposée pour mener diverses expériences pour différents matériaux ;

(c) une efficacité pour reproduire les courbes de dépendance en température et en champ magnétique de la capacité thermique et les courbes de dépendance en température et en champ magnétique de l'aimantation ;

(d) une efficacité de tous les algorithmes développés dans le cadre de cette thèse (FastEMO, QAES, QIU-NSA), en tant que noyau de l'optimisation de la méthode proposée : l'erreur absolue moyenne en pourcentage du changement d'entropie magnétique (ΔS) est environ de 0,2% ;

La méthode proposée, basée sur les algorithmes développés, devrait intéresser la communauté de la science des matériaux et de la réfrigération magnétique, car elle est capable d'assurer une reproduction fiable des propriétés de divers matériaux, ce qui est nécessaire pour obtenir une cohérence thermodynamique dans le modèle AMR.

6. **La conception du modèle du régénérateur magnétique actif (l'AMR) pour deux modes fonctionnels a été étudiée.**

   Ce travail vise à trouver une architecture innovante du modèle de l'AMR, qui peut fonctionner dans les deux modes d'application : en tant qu'un système de

réfrigération magnétique (MRS) et en tant qu'un générateur d'énergie thermomagnétique (TMG).

Nos résultats de l'étude bibliographique effectuée montre un manque de recherches sur la conception des modèles de l'AMR pour les deux modes (MRS et TMG). Ce type de recherche est nécessaire afin d'obtenir des informations détaillées sur la relation entre les valeurs des paramètres de contrôle et de conception du modèle AMR et de ses performances dans les deux modes.

Dans le cadre de cette thèse, le modèle de l'AMR a été analysé en tant qu'une partie d'un problème d'optimisation. En s'appuyant sur les résultats de cette analyse, un algorithme d'optimisation parallèle avec plusieurs objectifs, FastEMO, a été sélectionné et utilisé via l'outil basé sur EASEA.

Le cas de recherche présenté est centré sur les quatre objectifs d'optimisation suivants : l'efficacité énergétique et la densité de puissance pour chaque mode.

Les résultats de la validation expérimentale confirment que :

(a) l'outil basé sur l'EASEA est applicable avec succès pour étudier différentes configurations du modèle de l'AMR ;

(b) FastEMO est efficace pour résoudre les problèmes à quatre objectifs avec des solutions résistantes à la dominance (DRS) ;

(c) la combinaison commune des paramètres optimaux du modèle de l'AMR pour les deux modes a été déterminée.

(d) les informations détaillées sur la relation entre les paramètres et les performances du modèle de l'AMR ont été obtenues.

La recherche présentée est considérée comme un exemple. Les recherches supplémentaires sont nécessaires, avec un plus grand nombre de paramètres pris en considération.

Conclusion

Dans ce projet de thèse, nous avons proposé une approche pour optimiser les systèmes de réfrigération magnétique, basée sur les algorithmes évolutionnaires et inspirés du quantique. Nos résultats scientifiques et pratiques de cette thèse montrent que l'approche proposée résout avec succès les problèmes scientifiques suivants:

1. reproduire les propriétés physiques de divers matériaux magnétocaloriques par la méthode et l'outil proposés, qui permet de garantir une cohérence thermodynamique et une parfaite conservation de l'énergie dans le modèle de l'AMR;

2. étudier l'impact de différentes combinaisons de paramètres de contrôle et de design du modèle de l'AMR sur sa performance, en prenant en compte le fait que le modèle peut fonctionner en deux modes : en tant qu'un système de réfrigération magnétique et en tant qu'un générateur d'énergie thermomagnétique.

Ces résultats obtenus ont été exploités dans les travaux de recherche du laboratoire *Crismat* et de l'entreprise *Ubiblue* dans le cadre du projet ANR *CoolMagEvo*.

L'objectif scientifique de cette thèse est accompli: un nouvel algorithme évolutionnaire (FastEMO) et des algorithmes d'optimisation inspirés du quantique (QAES, QIU-NSA) ont été développés, étudiés et appliqués pour résoudre différents problèmes du modèle de l'hamiltonien des matériaux magnétocaloriques (MMCs) et du modèle complexe du Régénérateur Magnétique Actif (AMR). Un outil spécial a été développé sur la base de la nouvelle version de la plateforme EASEA 2.20, qui a également été fait dans le cadre de cette thèse, et testé sur différentes suites de tests et problèmes du monde réel.

Perspectives

Nous proposons plusieurs directions possibles pour les recherches futures qui sont prometteuses. Certaines de ces idées sont comme suit:

1. Puisque le modèle de l'AMR est la partie fondamentale des objectifs, le code du modèle doit être porté sur les cartes GPU, afin d'évaluer les objectifs en parallèle. Cela nous permet d'exploiter complètement l'extensibilité par rapport à la taille de la population.

2. D'après nos résultats, l'algorithme QIU-NSA est très efficace pour résoudre des problèmes séparables sur un espace de recherche à grande échelle. Nous supposons que cette propriété de l'algorithme peut être étendue pour résoudre des problèmes non séparables en convertissant des problèmes non séparables en problèmes séparables. Pour cela nous supposons d'essayer l'une des deux techniques suivantes afin de l'utiliser en tant qu'un "convertisseur de problèmes":

   (a) La méthode basée sur le *The Feed-Forward Neural Network*, proposée par [Lu & Ito (2003)];

   (b) La procédure de codage adaptatif basée sur l'adaptation de la matrice de covariance, présentée dans [Hansen (2008)].

   Les deux méthodes peuvent être considérées comme des méthodes qui peuvent être utilisées pour la transformation de problèmes d'optimisation non séparables en problèmes séparables, même lorsque la fonction d'objectif est inconnue.

3. Une perspective logique pour l'algorithme QAES, basé sur la méthode DMC, est une implémentation de la technique de renforcement, qui permet de renforcer l'algorithme QAES par un schéma d'initialisation de la première fonction d'onde. Nous supposons que si nous réussissons à utiliser l'algorithme de recuit simulé en tant qu'une technique renforcée, QAES serait boosté sur des problèmes plus complexes et démontrerait de meilleures performances de recherche.

4. Puisque le modèle BEG-I développé par le laboratoire *Crismat* est présenté comme un hamiltonien, la prochaine étape logique est de l'implémenter sur un matériel quantique, afin de trouver rapidement et précisément son état fondamental. Pour cela, l'un des algorithmes hybrides quantique-classique existants doit être utilisé.

Importance scientifique et pratique

L'importance des travaux réalisés dans le cadre de cette thèse est confirmée par les arguments suivants :

1. La théorie et la pratique de la science des matériaux magnétiques peut obtenir un progrès en appliquant la nouvelle méthode proposée pour modéliser et reproduire les propriétés physiques des matériaux magnétiques.

2. L'industrie du refroidissement magnétique a obtenu un outil d'optimisation puissant et convivial pour résoudre des problèmes du monde réel avec un débit de calcul élevé.

3. Un progrès dans le développement des algorithmes inspirés du quantique peut être atteint en appliquant les résultats de cette thèse.

4. Les solutions des problèmes de recherche présentés ont un impact direct sur le développement de la science des matériaux, de la technologie du refroidissement magnétique, et a un impact indirect sur la résolution des problèmes environnementaux.

Bibliography

Abhijith, J., Adedoyin, A., Ambrosiano, J., Anisimov, P., Bärtschi, A., Casper, W., Chennupati, G., Coffrin, C., Djidjev, H., Gunter, D. et al. (2018), 'Quantum algorithm implementations for beginners', *arXiv e-prints* pp. arXiv–1804.

Abohashima, Z., Elhosen, M., Houssein, E. H. & Mohamed, W. M. (2020), 'Classification with quantum machine learning: A survey', *arXiv preprint arXiv:2006.12270* .

Agrawal, M. K. & Matani, A. G. (2012), 'Evaluation of vapour compression refrigeration system using different refrigerants—a review', *Int J Eng Innov Technol* **2**(4), 39–43.

Aguirre, H. E. & Tanaka, K. (2007), 'Working principles, behavior, and performance of moeas on mnk-landscapes', *European Journal of Operational Research* **181**(3), 1670–1690.

Alvarez-Alvarado, M. S., Alban-Chacón, F. E., Lamilla-Rubio, E. A., Rodr´iguez-Gallegos, C. D. & Velásquez, W. (2021), 'Three novel quantum-inspired swarm optimization algorithms using different bounded potential fields', *Scientific Reports* **11**(1), 1–22.

Amaral, J., Das, S. & Amaral, V. (2011), 'The mean-field theory in the study of ferromagnets and the magnetocaloric effect', *Thermodynamics: Systems in Equilibrium and Non-Equilibrium* p. 173.

Aprea, C., Greco, A. & Maiorino, A. (2015), 'Geothermag: A geothermal magnetic refrigerator', *International Journal of refrigeration* **59**, 75–83.

Aprea, C., Greco, A. & Maiorino, A. (2017), 'An application of the artificial neural network to optimise the energy performances of a magnetic refrigerator', *International Journal of Refrigeration* **82**, 238–251.

Aprea, C., Greco, A., Maiorino, A. & Masselli, C. (2017), 'Analyzing the energetic performances of amr regenerator working with different magnetocaloric materials: Investigations and viewpoints', *Int. J. Heat Technol* **35**, S383–S390.

Audet, C., Bigeon, J., Cartier, D., Le Digabel, S. & Salomon, L. (2020), 'Performance indicators in multiobjective optimization', *European journal of operational research* .

Balli, M., Mahmed, C., Sari, O., Rahali, F. & Hadorn, J. (2011), Engineering of the magnetic cooling systems: A promising research axis for environment and energy saving, *in* 'World Engineers' Convention'.

Bellman, R. (1957), 'Dynamic programming, princeton univ', *Press Princeton, New Jersey* .

Benford, S. & Brown, G. (1981), 'T-s diagram for gadolinium near the curie temperature', *Journal of Applied Physics* **52**(3), 2110–2112.

Blázquez, J., Ipus, J., Moreno-Ram´irez, L., Álvarez-Gómez, J., Sánchez-Jiménez, D., Lozano-Pérez, S., Franco, V. & Conde, A. (2017), 'Ball milling as a way to produce magnetic and magnetocaloric materials: a review', *Journal of Materials Science* **52**(20), 11834–11850.

Blowers, P. & Lownsbury, J. M. (2010), 'Carbon dioxide emission implications if hydrofluorocarbons are regulated: a refrigeration case study'.

Bornemann, F. (1997), 'A note on the adiabatic theorem of quantum mechanics'.

Bouchekara, H. R. E.-H. & Nahas, M. (2012), Magnetic refrigeration technology at room temperature, *in* 'Trends in Electromagnetism-From Fundamentals to Applications', InTech.

Brockhoff, D., Tusar, T., Auger, A. & Hansen, N. (2016), 'Using well-understood single-objective functions in multiobjective black-box optimization test suites', *arXiv preprint arXiv:1604.00359* .

Brück, E. (2005), 'Developments in magnetocaloric refrigeration', *Journal of Physics D: Applied Physics* **38**(23), R381.

Cao, Y., Romero, J., Olson, J. P., Degroote, M., Johnson, P. D., Kieferová, M., Kivlichan, I. D., Menke, T., Peropadre, B., Sawaya, N. P. et al. (2019), 'Quantum chemistry in the age of quantum computing', *Chemical reviews* **119**(19), 10856–10915.

Chaudhary, V., Chen, X. & Ramanujan, R. V. (2019), 'Iron and manganese based magnetocaloric materials for near room temperature thermal management', *Progress in Materials Science* **100**, 64–98.

Chen, R., Dong, C., Ye, Y., Chen, Z. & Liu, Y. (2019), 'Qssa: Quantum evolutionary salp swarm algorithm for mechanical design', *IEEE Access* **7**, 145582–145595.

Choi, T. J., Togelius, J. & Cheong, Y.-G. (2020), 'Advanced cauchy mutation for differential evolution in numerical optimization', *IEEE Access* **8**, 8720–8734.

Collet, P., Krüger, F. & Maitre, O. (2013), Automatic parallelization of ec on gpgpus and clusters of gpgpu machines with easea and easea-cloud, *in* 'Massively Parallel Evolutionary Computation on GPGPUs', Springer, pp. 35–59.

Collet, P., Lutton, E., Schoenauer, M. & Louchet, J. (2000), Take it easea, *in* 'International Conference on Parallel Problem Solving from Nature', Springer, pp. 891–901.

Collet, P. & Rennard, J.-P. (2008), Stochastic optimization algorithms, *in* 'Intelligent information technologies: Concepts, methodologies, tools, and applications', IGI Global, pp. 1121–1137.

Coulomb, D., Dupont, J. & Pichard, A. (2015), 'The role of refrigeration in the global economy', *International Institute of Refrigeration: Paris, France* .

Curry, D. M. & Dagli, C. H. (2014), 'Computational complexity measures for many-objective optimization problems', *Procedia Computer Science* **36**, 185–191.

da Cruz, A. V. A., Vellasco, M. M. & Pacheco, M. A. C. (2010), Quantum-inspired evolutionary algorithms applied to numerical optimization problems, *in* 'IEEE Congress on Evolutionary Computation', IEEE, pp. 1–6.

Das, I. & Dennis, J. E. (1998), 'Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems', *SIAM journal on optimization* **8**(3), 631–657.

de Castro, P. B., Terashima, K., Yamamoto, T. D., Hou, Z., Iwasaki, S., Matsumoto, R., Adachi, S., Saito, Y., Song, P., Takeya, H. et al. (2020), 'Machine-learning-guided discovery of the gigantic magnetocaloric effect in hob 2 near the hydrogen liquefaction temperature', *NPG Asia Materials* **12**(1), 1–7.

Deb, K., Agrawal, R. B. et al. (1995), 'Simulated binary crossover for continuous search space', *Complex systems* **9**(2), 115–148.

Deb, K., Agrawal, S., Pratap, A. & Meyarivan, T. (2000), A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii, *in* 'International conference on parallel problem solving from nature', Springer, pp. 849–858.

Deb, K. & Algorithms, M.-O. O. U. E. (2001), 'Wiley, chichester, uk', *Google Scholar Google Scholar Digital Library Digital Library* .

Deb, K. & Deb, D. (2014), 'Analysing mutation schemes for real-parameter genetic algorithms', *International Journal of Artificial Intelligence and Soft Computing* **4**(1), 1–28.

Deb, K. & Jain, H. (2013), 'An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints', *IEEE transactions on evolutionary computation* **18**(4), 577–601.

Deb, K., Thiele, L., Laumanns, M. & Zitzler, E. (2002), Scalable multi-objective optimization test problems, *in* 'Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)', Vol. 1, IEEE, pp. 825–830.

Deb, K. & Tiwari, S. (2008), 'Omni-optimizer: A generic evolutionary algorithm for global optimization', *European Journal of Operations Research (EJOR)* .

Demir, İ., Ergin, F. C. & Kiraz, B. (2019), 'A new model for the multi-objective multiple allocation hub network design and routing problem', *IEEE Access* **7**, 90678–90689.

DiVincenzo, D. & Terhal, B. (1998), 'Decoherence: the obstacle to quantum computation', *Physics world* **11**(3), 53.

Doerr, B., Jansen, T., Sudholt, D., Winzen, C. & Zarges, C. (2013), 'Mutation rate matters even when optimizing monotonic functions', *Evolutionary computation* **21**(1), 1–27.

Doerr, B., Le, H. P., Makhmara, R. & Nguyen, T. D. (2017), Fast genetic algorithms, *in* 'Proceedings of the Genetic and Evolutionary Computation Conference', pp. 777–784.

Draa, A., Meshoul, S., Talbi, H. & Batouche, M. (2004), A quantum inspired differential evolution algorithm for rigid image registration, *in* 'in Proceedings of the International Conference on Computational Intelligence, Istanbul', Citeseer.

Elhara, O., Varelas, K., Nguyen, D., Tusar, T., Brockhoff, D., Hansen, N. & Auger, A. (2019), 'Coco: the large scale black-box optimization benchmarking (bbob-largescale) test suite', *arXiv preprint arXiv:1903.06396* .

Elhossini, A., Areibi, S. & Dony, R. (2010), 'Strength pareto particle swarm optimization and hybrid ea-pso for multi-objective optimization', *Evolutionary Computation* **18**(1), 127–156.

Falcón-Cardona, J. G., Gómez, R. H., Coello, C. A. C. & Tapia, M. G. C. (2021), 'Parallel multi-objective evolutionary algorithms: A comprehensive survey', *Swarm and Evolutionary Computation* p. 100960.

Farhi, E., Goldstone, J. & Gutmann, S. (2014), 'A quantum approximate optimization algorithm', *arXiv preprint arXiv:1411.4028* .

Fei, G., Zhuo-Qiu, L. & Heng-Qing, T. (2008), 'Parameters estimation online for lorenz system by a novel quantum-behaved particle swarm optimization', *Chinese Physics B* **17**(4), 1196.

Feynman, R. P. (1982), 'Simulating physics with computers', *Int. J. Theor. Phys* **21**(6/7).

Fonseca, C. M., Fleming, P. J. et al. (1993), Genetic algorithms for multiobjective optimization: Formulationdiscussion and generalization., *in* 'Icga', Vol. 93, Citeseer, pp. 416–423.

Foulkes, W., Mitas, L., Needs, R. & Rajagopal, G. (2001), 'Quantum monte carlo simulations of solids', *Reviews of Modern Physics* **73**(1), 33.

Franco, V., Blázquez, J., Ipus, J., Law, J., Moreno-Ram´irez, L. & Conde, A. (2018), 'Magnetocaloric effect: From materials research to refrigeration devices', *Progress in Materials Science* **93**, 112–232.

Ganjehsarabi, H., Dincer, I. & Gungor, A. (2016), 'Analysis and optimisation of a cascade active magnetic regenerative refrigeration system', *International Journal of Exergy* **19**(2), 143–160.

Gombi, S. & Sahu, D. (2020), 'A review on magneto-caloric materials for room temperature refrigeration', *International Journal of Automotive and Mechanical Engineering* **17**(1), 7805–7815.

Gómez, J. R., Garcia, R. F., Catoira, A. D. M. & Gómez, M. R. (2013), 'Magnetocaloric effect: A review of the thermodynamic cycles in magnetic refrigeration', *Renewable and Sustainable Energy Reviews* **17**, 74–82.

Greensite, J. (2003), 'Lecture notes on quantum mechanics'.

Grover, L. K. (1996), A fast quantum mechanical algorithm for database search, *in* 'Proceedings of the twenty-eighth annual ACM symposium on Theory of computing', pp. 212–219.

Gschneidner Jr, K. A. & Pecharsky, V. K. (2006), 'Rare earths and magnetic refrigeration', *Journal of Rare Earths* **24**(6), 641–647.

Gschneidner Jr, K. A., Pecharsky, V. & Tsokol, A. (2005), 'Recent developments in magnetocaloric materials', *Reports on progress in physics* **68**(6), 1479.

Gutfleisch, O., Gottschall, T., Fries, M., Benke, D., Radulov, I., Skokov, K. P., Wende, H., Gruner, M., Acet, M., Entel, P. et al. (2016), 'Mastering hysteresis in magnetocaloric materials', *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **374**(2074), 20150308.

Hansen, N. (2008), Adaptive encoding: How to render search coordinate system invariant, *in* 'International Conference on Parallel Problem Solving from Nature', Springer, pp. 205–214.

Hansen, N. (2009), Benchmarking a bi-population cma-es on the bbob-2009 function testbed, *in* 'Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers', pp. 2389–2396.

Hansen, N., Arnold, D. V. & Auger, A. (2015), Evolution strategies, *in* 'Springer handbook of computational intelligence', Springer, pp. 871–898.

Hansen, N., Auger, A., Finck, S. & Ros, R. (2012), 'Real-parameter black-box optimization benchmarking: Experimental setup', *Orsay, France: Université Paris Sud, Institut National de Recherche en Informatique et en Automatique (INRIA) Futurs, Équipe TAO, Tech. Rep* .

Hansen, N., Auger, A., Ros, R., Mersmann, O., Tušar, T. & Brockhoff, D. (2021), 'Coco: A platform for comparing continuous optimizers in a black-box setting', *Optimization Methods and Software* **36**(1), 114–144.

Hansen, N., Finck, S., Ros, R. & Auger, A. (2009), Real-parameter black-box optimization benchmarking 2009: Noiseless functions definitions, PhD thesis, INRIA.

Hansen, N., Gemperle, F., Auger, A. & Koumoutsakos, P. (2006), When do heavy-tail distributions help?, *in* 'Parallel Problem Solving from Nature-PPSN IX', Springer, pp. 62–71.

Hansen, N., Müller, S. D. & Koumoutsakos, P. (2003), 'Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es)', *Evolutionary computation* **11**(1), 1–18.

Hansen, N. & Ostermeier, A. (2001), 'Completely derandomized self-adaptation in evolution strategies', *Evolutionary computation* **9**(2), 159–195.

Harada, T. & Takadama, K. (2017), A study of self-adaptive semi-asynchronous evolutionary algorithm on multi-objective optimization problem, *in* 'Proceedings of the Genetic and Evolutionary Computation Conference Companion', pp. 1812–1819.

Harada, T. & Takadama, K. (2020), 'Analysis of semi-asynchronous multi-objective evolutionary algorithm with different asynchronies', *Soft Computing* **24**(4), 2917–2939.

Hassani, K. & Lee, W.-S. (2014), An incremental framework for classification of eeg signals using quantum particle swarm optimization, *in* '2014 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)', IEEE, pp. 40–45.

He, J., Chen, T. & Mitavskiy, B. (2011), 'Novel analysis of population scalability in evolutionary algorithms', *arXiv preprint arXiv:1108.4531* .

Herbrich, R., Graepel, T. & Obermayer, K. (1999), 'Support vector learning for ordinal regression'.

Hess, T., Maier, L., Bachmann, N., Corhan, P., Schäfer-Welsen, O., Wöllenstein, J. & Bartholomé, K. (2020), 'Thermal hysteresis and its impact on the efficiency of first-order caloric materials', *Journal of Applied Physics* **127**(7), 075103.

Hiroyasu, T., Nakayama, S. & Miki, M. (2005), Comparison study of spea2+, spea2, and nsga-ii in diesel engine emissions and fuel economy problem, *in* '2005 IEEE congress on evolutionary computation', Vol. 1, IEEE, pp. 236–242.

Houle, M. E., Kriegel, H.-P., Kröger, P., Schubert, E. & Zimek, A. (2010), Can shared-neighbor distances defeat the curse of dimensionality?, *in* 'International conference on scientific and statistical database management', Springer, pp. 482–500.

Huang, W., Zhang, Y. & Li, L. (2019), Survey on multi-objective evolutionary algorithms, *in* 'Journal of Physics: Conference Series', Vol. 1288, IOP Publishing, p. 012057.

Huband, S., Barone, L., While, L. & Hingston, P. (2005), A scalable multi-objective test problem toolkit, *in* 'International Conference on Evolutionary Multi-Criterion Optimization', Springer, pp. 280–295.

Ibrahim, A., Martin, M. V., Rahnamayan, S. & Deb, K. (2017), Fusion-based hybrid many-objective optimization algorithm, *in* '2017 IEEE Congress on Evolutionary Computation (CEC)', IEEE, pp. 2372–2381.

Ibrahim, A., Rahnamayan, S., Martin, M. V. & Deb, K. (2017), Fusion of many-objective non-dominated solutions using reference points, *in* 'International Conference on Evolutionary Multi-Criterion Optimization', Springer, pp. 314–328.

Igel, C., Hansen, N. & Roth, S. (2007), 'Covariance matrix adaptation for multi-objective optimization', *Evolutionary computation* **15**(1), 1–28.

Kennedy, J. & Eberhart, R. (1995), Particle swarm optimization, *in* 'Proceedings of ICNN'95-international conference on neural networks', Vol. 4, IEEE, pp. 1942–1948.

Kent, P. R. C. (1999), Techniques and applications of quantum Monte Carlo, PhD thesis, University of Cambridge.

Kitanovski, A. (2020), 'Energy applications of magnetocaloric materials', *Advanced Energy Materials* **10**(10), 1903741.

Kleijnen, J. P. (2009), 'Kriging metamodeling in simulation: A review', *European journal of operational research* **192**(3), 707–716.

Knowles, J. D., Thiele, L. & Zitzler, E. (2006), 'A tutorial on the performance assessment of stochastic multiobjective optimizers', *TIK-report* **214**.

Kosztin, I., Faber, B. & Schulten, K. (1996), 'Introduction to the diffusion monte carlo method', *American Journal of Physics* **64**(5), 633–644.

Kotze, J. (2008), 'Introduction to monte carlo methods for an ising model of a ferromagnet', *arXiv preprint arXiv:0803.0217* .

Kudo, F., Yoshikawa, T. & Furuhashi, T. (2011), A study on analysis of design variables in pareto solutions for conceptual design optimization problem of hybrid rocket engine, *in* '2011 IEEE Congress of Evolutionary Computation (CEC)', IEEE, pp. 2558–2562.

Kukkonen, S. & Deb, K. (2006), Improved pruning of non-dominated solutions based on crowding distance for bi-objective optimization problems, *in* '2006 IEEE International Conference on Evolutionary Computation', IEEE, pp. 1179–1186.

Larranaga, P. (2002), A review on estimation of distribution algorithms, *in* 'Estimation of distribution algorithms', Springer, pp. 57–100.

Lee, C.-Y. & Park, Y.-J. (2014), 'Effect of multivariate cauchy mutation in evolutionary programming', *IEICE TRANSACTIONS on Information and Systems* **97**(4), 821–829.

Lei, X. & Fu, A. (2008), Two-dimensional maximum entropy image segmentation method based on quantum-behaved particle swarm optimization algorithm, *in* '2008 Fourth International Conference on Natural Computation', Vol. 3, IEEE, pp. 692–696.

Li, B., Huang, S., Ye, J., Li, Y., Shen, A. & Deng, J. (2020), 'Optimal design method for llcl filters based on nsga-iii', *Journal of Power Electronics* **20**, 1250–1260.

Li, P. & Li, S. (2008), 'Quantum-inspired evolutionary algorithm for continuous space optimization based on bloch coordinates of qubits', *Neurocomputing* **72**(1-3), 581–591.

Li, S., Wang, R., Hu, W. & Sun, J. (2007), A new qpso based bp neural network for face detection, *in* 'Fuzzy information and engineering', Springer, pp. 355–363.

Lionte, S., Risser, M. & Muller, C. (2021), 'A 15kw magnetocaloric proof-of-concept unit: Initial development and first experimental results', *International Journal of Refrigeration* **122**, 256–265.

Lionte, S., Vasile, C. & Siroux, M. (2015), 'Numerical analysis of a reciprocating active magnetic regenerator', *Applied Thermal Engineering* **75**, 871–879.

Loshchilov, I. (2014), A computationally efficient limited memory cma-es for large scale optimization, *in* 'Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation', pp. 397–404.

Lu, B.-L. & Ito, K. (2003), 'Converting general nonlinear programming problems into separable programming problems with feedforward neural networks', *Neural networks* **16**(7), 1059–1074.

Luna, F. & Alba, E. (2015), Parallel multiobjective evolutionary algorithms, *in* 'Springer Handbook of Computational Intelligence', Springer, pp. 1017–1031.

Lyubina, J. (2017), 'Magnetocaloric materials for energy efficient cooling', *Journal of Physics D: Applied Physics* **50**(5), 053002.

Maier, H. R., Razavi, S., Kapelan, Z., Matott, L. S., Kasprzyk, J. & Tolson, B. A. (2019), 'Introductory overview: Optimization using evolutionary algorithms and other metaheuristics', *Environmental modelling & software* **114**, 195–213.

Maiorino, A., Del Duca, M. G., Tušek, J., Tomc, U., Kitanovski, A. & Aprea, C. (2019), 'Evaluating magnetocaloric effect in magnetocaloric materials: a novel approach based on indirect measurements using artificial neural networks', *Energies* **12**(10), 1871.

McArdle, S., Endo, S., Aspuru-Guzik, A., Benjamin, S. C. & Yuan, X. (2020), 'Quantum computational chemistry', *Reviews of Modern Physics* **92**(1), 015003.

Miettinen, K. (2000), 'Nonlinear multiobjective optimization', *JOURNAL-OPERATIONAL RESEARCH SOCIETY* **51**(2), 246–246.

Mirjalili, S. & Hashim, S. Z. M. (2010), A new hybrid psogsa algorithm for function optimization, *in* '2010 international conference on computer and information application', IEEE, pp. 374–377.

Morrish, A. H. (2001), *The physical principles of magnetism.*

Mozharivskyj, Y. (2016), 'Magnetocaloric effect and magnetocaloric materials'.

Orr, M. J. et al. (1996), 'Introduction to radial basis function networks'.

Ouskova Leonteva, A., Abdulkarimova, U., Jeannin-Girardon, A., Risser, M., Parrend, P. & Collet, P. (2020), Quantum-inspired algorithm with evolution strategy, *in* 'International Conference on the Theory and Practice of Natural Computing', Springer, pp. 95–106.

Ouskova Leonteva, A., Parrend, P., Jeannin-Girardon, A. & Collet, P. (2019), Fast evolutionary algorithm for solving large-scale multi-objective problems, *in* 'International Conference on Artificial Evolution (Evolution Artificielle)', Springer, pp. 82–95.

Pastorello, D. & Blanzieri, E. (2019), 'Learning adiabatic quantum algorithms for solving optimization problems', *arXiv preprint arXiv:1909.06870* .

Peruzzo, A., McClean, J., Shadbolt, P., Yung, M.-H., Zhou, X.-Q., Love, P. J., Aspuru-Guzik, A. & O'brien, J. L. (2014), 'A variational eigenvalue solver on a photonic quantum processor', *Nature communications* **5**(1), 1–7.

Preuss, M., Kausch, C., Bouvy, C. & Henrich, F. (2010), Decision space diversity can be essential for solving multiobjective real-world problems, *in* 'Multiple Criteria Decision Making for Sustainable Energy and Transportation Systems', Springer, pp. 367–377.

Purshouse, R. C. & Fleming, P. J. (2003), Conflict, harmony, and independence: Relationships in evolutionary multi-criterion optimisation, *in* 'International Conference on Evolutionary Multi-Criterion Optimization', Springer, pp. 16–30.

QUERRY, S., LACHICHE, N. & COLLET, P. (2017), 'Parallelizing evolutionary algorithms on gpgpu cards with the easea platform', *Programming Multicore and Many-core Computing Systems* p. 301.

Rashedi, E., Nezamabadi-Pour, H. & Saryazdi, S. (2009), 'Gsa: a gravitational search algorithm', *Information sciences* **179**(13), 2232–2248.

Razavi, S., Tolson, B. A. & Burn, D. H. (2012), 'Review of surrogate modeling in water resources', *Water Resources Research* **48**(7).

Rechenberg, I. (1973), 'Evolutionsstrategie: Optimierung technischer systeme nach prinzipien der biologischen evolution, frommann–holzboog', *Stuttgart-Bad Cannstatt* .

Rehman, O. U., Yang, J., Zhou, Q., Yang, S. & Khan, S. (2017), 'A modified qpso algorithm applied to engineering inverse problems in electromagnetics', *International Journal of Applied Electromagnetics and Mechanics* **54**(1), 107–121.

Rehman, O. U., Yang, S. & Khan, S. U. (2017), 'A modified quantum-based particle swarm optimization for engineering inverse problem', *COMPEL-The international journal for computation and mathematics in electrical and electronic engineering* .

Risser, M., Vasile, C., Keith, B., Engel, T. & Muller, C. (2012), 'Construction of consistent magnetocaloric materials data for modelling magnetic refrigerators', *international journal of refrigeration* **35**(2), 459–467.

Risser, M., Vasile, C., Muller, C. & Noume, A. (2013), 'Improvement and application of a numerical model for optimizing the design of magnetic refrigerators', *international journal of refrigeration* **36**(3), 950–957.

Roeva, O., Fidanova, S. & Paprzycki, M. (2015), Population size influence on the genetic and ant algorithms performance in case of cultivation process modeling, *in* 'Recent advances in computational optimization', Springer, pp. 107–120.

Ros, R. (2009), Benchmarking the bfgs algorithm on the bbob-2009 function testbed, *in* 'Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers', pp. 2409–2414.

Ros, R. & Hansen, N. (2008), A simple modification in cma-es achieving linear time and space complexity, *in* 'International Conference on Parallel Problem Solving from Nature', Springer, pp. 296–305.

Roy, S., Poncet, S. & Sorin, M. (2017), 'Sensitivity analysis and multiobjective optimization of a parallel-plate active magnetic regenerator using a genetic algorithm', *international journal of refrigeration* **75**, 276–285.

Rubinstein, R. Y. & Kroese, D. P. (2016), *Simulation and the Monte Carlo method*, Vol. 10, John Wiley & Sons.

Rylander, S. G. B. & Gotshall, B. (2002), 'Optimal population size and the genetic algorithm', *Population* **100**(400), 900.

Safi, H. H., Ucan, O. N. & Bayat, O. (2018), On the real world applications of many-objective evolutionary algorithms, *in* 'Proceedings of the First International Conference on Data Science, E-learning and Information Systems', pp. 1–6.

Sato, H., Aguirre, H. E. & Tanaka, K. (2007), Controlling dominance area of solutions and its impact on the performance of moeas, *in* 'International conference on evolutionary multi-criterion optimization', Springer, pp. 5–20.

Seada, H. & Deb, K. (2014), 'U-nsga-iii: A unified evolutionary algorithm for single, multiple, and many-objective optimization', *COIN report* **2014022**.

Seada, H. & Deb, K. (2015), Effect of selection operator on nsga-iii in single, multi, and many-objective optimization, *in* '2015 IEEE Congress on Evolutionary Computation (CEC)', IEEE, pp. 2915–2922.

Shan, S. & Wang, G. G. (2010), 'Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions', *Structural and multidisciplinary optimization* **41**(2), 219–241.

Sharma, D. & Collet, P. (2010*a*), An archived-based stochastic ranking evolutionary algorithm (asrea) for multi-objective optimization, *in* 'Proceedings of the 12th annual conference on Genetic and evolutionary computation', pp. 479–486.

Sharma, D. & Collet, P. (2010*b*), An archived-based stochastic ranking evolutionary algorithm (asrea) for multi-objective optimization, *in* 'Proceedings of the 12th annual conference on Genetic and evolutionary computation', pp. 479–486.

Sharma, D. & Collet, P. (2013), Implementation techniques for massively parallel multi-objective optimization, *in* 'Massively Parallel Evolutionary Computation on GPGPUs', Springer, pp. 267–286.

Shor, P. W. (1994), Algorithms for quantum computation: discrete logarithms and factoring, *in* 'Proceedings 35th annual symposium on foundations of computer science', Ieee, pp. 124–134.

Sofge, D. A. (2008), 'Prospective algorithms for quantum evolutionary computation', *arXiv preprint arXiv:0804.1133* .

Sokolovskiy, V., Buchelnikov, V. D. & Taskaev, S. (2009), Monte-carlo calculation of the magnetocaloric effect in ni-mn-ga alloys, *in* 'Solid State Phenomena', Vol. 152, Trans Tech Publ, pp. 493–496.

Sokolovskiy, V., Buchelnikov, V., Taskaev, S. & Entel, P. (2010), Theoretical model of the coupled magnetostructural phase transitions in heusler ni-mn-in alloys by monte carlo simulation, *in* 'Journal of Physics: Conference Series', Vol. 200, IOP Publishing, p. 092004.

Solomon, D. (1991), 'Design of a thermomagnetic generator', *Energy conversion and management* **31**(2), 157–173.

Stanley, K. O., D'Ambrosio, D. B. & Gauci, J. (2009), 'A hypercube-based encoding for evolving large-scale neural networks', *Artificial life* **15**(2), 185–212.

Sun, J., Fang, W., Wu, X., Palade, V. & Xu, W. (2012), 'Quantum-behaved particle swarm optimization: analysis of individual particle behavior and parameter selection', *Evolutionary computation* **20**(3), 349–393.

Sun, J., Feng, B. & Xu, W. (2004), Particle swarm optimization with particles having quantum behavior, *in* 'Proceedings of the 2004 congress on evolutionary computation (IEEE Cat. No. 04TH8753)', Vol. 1, IEEE, pp. 325–331.

Sun, J., Liu, J. & Xu, W. (2007), 'Using quantum-behaved particle swarm optimization algorithm to solve non-linear programming problems', *International Journal of Computer Mathematics* **84**(2), 261–272.

Sun, J., Xu, W. & Feng, B. (2004), A global search strategy of quantum-behaved particle swarm optimization, *in* 'IEEE Conference on Cybernetics and Intelligent Systems, 2004.', Vol. 1, IEEE, pp. 111–116.

Szu, H. & Hartley, R. (1987), 'Fast simulated annealing', *Physics letters A* **122**(3-4), 157–162.

Talbi, E.-G. (2019), 'A unified view of parallel multi-objective evolutionary algorithms', *Journal of Parallel and Distributed Computing* **133**, 349–358.

Tang, L. & Wang, X. (2012), 'A hybrid multiobjective evolutionary algorithm for multiobjective optimization problems', *IEEE Transactions on Evolutionary Computation* **17**(1), 20–45.

Teyber, R., Trevizoli, P., Christiaanse, T., Govindappa, P., Niknia, I. & Rowe, A. (2018), 'Semi-analytic amr element model', *Applied Thermal Engineering* **128**, 1022–1029.

Thangaraj, R., Pant, M., Abraham, A. & Bouvry, P. (2011), 'Particle swarm optimization: hybridization perspectives and experimental illustrations', *Applied Mathematics and Computation* **217**(12), 5208–5226.

Tian, N., Lai, C.-H., Pericleous, K., Sun, J. & Xu, W. (2011), Contraction-expansion coefficient learning in quantum-behaved particle swarm optimization, *in* '2011 10th International Symposium on Distributed Computing and Applications to Business, Engineering and Science', IEEE, pp. 303–308.

Timani, B. & Peralta, R. (2017), 'Aggregated surrogate simulator for groundwater-surface water management via simulation-optimization modeling: Theory, development and tests', *Environmental modelling & software* **96**, 210–231.

Toulouse, J., Assaraf, R. & Umrigar, C. J. (2016), Introduction to the variational and diffusion monte carlo methods, *in* 'Advances in Quantum Chemistry', Vol. 73, Elsevier, pp. 285–314.

Tsutsui, S. & Collet, P. (2013), *Massively parallel evolutionary computation on GPGPUs*, Vol. 453, Springer.

Tu, S., Rehman, O. U., Rehman, S. U., Ullah, S., Waqas, M. & Zhu, R. (2020), 'A novel quantum inspired particle swarm optimization algorithm for electromagnetic applications', *IEEE Access* **8**, 21909–21916.

Tušek, J., Kitanovski, A., Prebil, I. & Poredoš, A. (2011), 'Dynamic operation of an active magnetic regenerator (amr): Numerical optimization of a packed-bed amr', *International Journal of Refrigeration* **34**(6), 1507–1517.

Varelas, K. (2019), Benchmarking large scale variants of cma-es and l-bfgs-b on the bbob-largescale testbed, *in* 'Proceedings of the Genetic and Evolutionary Computation Conference Companion', pp. 1937–1945.

Vasile, C., Engel, T., Risser, M. & Muller, C. (2008), Energy efficient and environmental safe magnetic cooling system, *in* 'The 7th International Conference', pp. 22–23.

Verdon, G., Arrazola, J. M., Brádler, K. & Killoran, N. (2019), 'A quantum approximate optimization algorithm for continuous problems', *arXiv preprint arXiv:1902.00409* .

Vives, E., Castán, T. & Lindgård, P.-A. (1996), 'Degenerate blume-emery-griffiths model for the martensitic transformation', *Physical Review B* **53**(14), 8915.

Wagner, T., Beume, N. & Naujoks, B. (2007), Pareto-, aggregation-, and indicator-based methods in many-objective optimization, *in* 'International conference on evolutionary multi-criterion optimization', Springer, pp. 742–756.

Wang, P., Cheng, K., Huang, Y., Li, B., Ye, X. & Chen, X. (2018), 'Multiscale quantum harmonic oscillator algorithm for multimodal optimization', *Computational intelligence and neuroscience* **2018**.

Wolpert, D. H. & Macready, W. G. (1997), 'No free lunch theorems for optimization', *IEEE transactions on evolutionary computation* **1**(1), 67–82.

Yao, X. & Liu, Y. (1997), Fast evolution strategies, *in* 'International Conference on Evolutionary Programming', Springer, pp. 149–161.

Zen, A., Sorella, S., Gillan, M. J., Michaelides, A. & Alfe, D. (2016), 'Boosting the accuracy and speed of quantum monte carlo: Size consistency and time step', *Physical Review B* **93**(24), 241118.

Zhang, Q. & Li, H. (2007), 'Moea/d: A multiobjective evolutionary algorithm based on decomposition', *IEEE Transactions on evolutionary computation* **11**(6), 712–731.

Zhang, Q., Zhou, A., Zhao, S., Suganthan, P. N., Liu, W. & Tiwari, S. (2008), 'Multiobjective optimization test instances for the cec 2009 special session and competition'.

Zhisheng, Z. (2010), 'Quantum-behaved particle swarm optimization algorithm for economic load dispatch of power system', *Expert Systems with Applications* **37**(2), 1800–1803.

Zitzler, E., Deb, K. & Thiele, L. (2000), 'Comparison of multiobjective evolutionary algorithms: Empirical results', *Evolutionary computation* **8**(2), 173–195.

Zitzler, E. & Künzli, S. (2004), Indicator-based selection in multiobjective search, *in* 'International conference on parallel problem solving from nature', Springer, pp. 832–842.

Zitzler, E., Laumanns, M. & Thiele, L. (2001), 'Spea2: Improving the strength pareto evolutionary algorithm', *TIK-report* **103**.

Université de Strasbourg

Anna OUSKOVA LEONTEVA

École doctorale Mathématiques, sciences de l'information et de l'ingénieur | ED 269 Université de Strasbourg

Algorithmes évolutionnaires et inspirés du quantique pour l'optimisation de systèmes de réfrigération magnétique

Résumé

Cette thèse est réalisée dans le cadre du projet ANR \textit{CoolMagEvo}, qui se donne comme objectif le développement de systèmes de réfrigération magnétique. Nous disposons de deux modèles de simulation dont on a assigné une fonction de problèmes d'optimisation: un modèle de Matériaux Magnetocaloriques (MMCs) et un modèle d'un Régénérateur Magnétique Actif (AMR). Selon les exigences des modèles, nous avons développé trois algorithmes d'optimisation. Par la suite, nous avons développé un outil basé sur les algorithmes proposés pour résoudre facilement et efficacement différents problèmes de ces modèles de simulation. Ensuite, à l'aide de cet outil, nous avons eu la possibilité d'ériger la nouvelle méthode basée sur l'optimisation, qui permet de reproduire les propriétés physiques de différents MMCs. Enfin, nous avons étudié l'impact des paramètres de contrôle et de dégin du modèle de l'AMR sur ses performances pour deux modes d'application.

Mots-clés : Algorithmes évolutionnaires, algorithmes d'optimisation inspirés du quantique, optimisation des modèles de simulation, matériaux magnétocaloriques, réfrigération magnétique.

Résumé en anglais

This thesis is carried out within the frame of the \textit{CoolMagEvo} ANR project, which aims at developing of magneto cooling systems. In this context, we consider the two following simulation models as time-consuming optimization problems: a model of Magneto Caloric Materials (MCMs) and a model of an Active Magnetic Regenerator (AMR). According to the defined requirements of the problems, we develop three special optimization algorithms. Then, we develop the software tool based on the proposed algorithms for easily solving different problems of these simulation models. Next, using this tool, we present the new method based on optimization, for reproducing physical properties of different MCMs. Finally, we study the impact of the control and design parameters of the AMR model on its performance in two application modes.

Keywords: Evolutionary algorithms, quantum-inspired optimization algorithms, optimization of simulation models, magneto caloric materials, magnetic refrigeration.