Multi-level approaches for optimal system design in railway applications

Alexandru Berbecea

N° d'ordre : 199

ECOLE CENTRALE DE LILLE

THESE

présentée en vue
d'obtenir le grade de

DOCTEUR

en

Spécialité : Génie Electrique

par

Alexandru BERBECEA

DOCTORAT DELIVRE PAR L'ECOLE CENTRALE DE LILLE

Titre de la thèse :

Multi-level approaches for optimal system design in railway applications

Approches multi-niveaux pour la conception systémique optimale des chaînes de traction ferroviaire

Soutenue le 10 Octobre 2012 devant le jury d'examen :

Examinateur	L. Krähenbühl	Directeur de recherche CNRS, Président du Jury	Ecole Centrale de Lyon
Rapporteur	J.-L. Coulomb	Professeur	INP Grenoble
Rapporteur	B. Sareni	Professeur	ENSEEIH Toulouse
Examinateur	S. Vivier	Maître de conférences HDR	UT de Compiègne
Examinateur	S. Brisset	Maître de conférences HDR	Ecole Centrale de Lille
Directeur de thèse	F. Gillon	Maître de conférences HDR	Ecole Centrale de Lille
Co-directeur	P. Brochet	Professeur	Ecole Centrale de Lille
Invité	E. Batista	Ingénieur	Alstom Transport, Tarbes

Thèse préparée dans le Laboratoire L2EP à l'Ecole Centrale de Lille
Ecole Doctorale SPI 072 (Lille I, Lille III, Artois, ULCO, UVHC, EC Lille)
PRES Université Lille Nord-de-France

Multi-level approaches for optimal system design in railway applications

Alexandru Berbecea

October, 10th 2012

Résumé

Dans le contexte actuel de mondialisation des marchés, le processus classique de conception par essais et erreurs, traditionnellement employé par les ingénieurs, n'est plus capable de répondre aux exigences de plus en plus accrues en termes de délais très courts, réduction des coûts de production, etc. L'outil d'optimisation propose une réponse à ces questions, en accompagnant les ingénieurs dans la tâche de conception optimale.

L'objectif de cette thèse est centré sur la conception optimale des systèmes complexes. Deux approches d'optimisation sont abordées dans ce travail: l'optimisation par modèles de substitution et la conception optimale basée sur la décomposition des systèmes complexes.

L'utilisation de la conception assistée par ordinateur (CAO) est devenue une pratique régulière dans l'industrie. La démarche d'optimisation basée sur modèles de substitution est destinée à répondre à l'optimisation des dispositifs avec ce genre de modèles coûteux de simulation, tels que les éléments finis (EF) en électromagnétisme. L'optimisation multi-objectif se présente comme un outil d'aide à la décision, en aidant le concepteur à prendre une décision éclairée. Le calcul distribué est utilisé pour réduire le temps global du processus d'optimisation.

Les systèmes d'ingénierie tels que les chaînes de traction ferroviaires sont trop complexes pour être traités comme un tout. Les stratégies d'optimisation basées sur la décomposition cherchent à répondre à la conception optimale de ces systèmes. Les approches de décomposition par modèle, discipline ou objet visent à distribuer la charge de calcul. Des stratégies de coordination multi-niveaux sont utilisées pour gérer le processus d'optimisation. Ces approches permettent à chaque équipe de spécialistes de travailler sur leur expertise de façon autonome. Les techniques d'optimisation à base de modèles de substitution peuvent être intégrées dans les stratégies d'optimisation multi-niveaux, allégeant ainsi la charge de calcul.

Les approches d'optimisation développées au sein de ce travail sont appliquées pour résoudre plusieurs problèmes d'optimisation électromagnétiques, ainsi que la conception optimale d'un système de traction ferroviaire de la Société Alstom.

Mots-clés

Conception optimale des systèmes électromagnétiques
Optimisation par modèles de substitution
Conception optimale de systèmes complexes basée sur décomposition
Optimisation multi-niveaux
Chaîne de traction ferroviaire

Abstract

Within a globalized market context, the classical trial-and-error design process traditionally employed by engineers is no longer capable of answering to the ever-growing demands in terms of short deadlines, reduced production costs, etc. The optimization tool presents itself as an answer to these issues, accompanying the engineers in the optimal design task.

The focus of this thesis is centered on the optimal design of complex systems. Two main optimization approaches are addressed in this work: the metamodel-based design optimization and the decomposition-based complex systems optimal design.

The use of computer-aided design/engineering (CAD/CAE) software has become a regular practice in the engineering design process. The metamodel-based optimization approach is intended to address the optimization of devices represented by such computational expensive simulation models, as the finite element analysis (FEA) in electromagnetics. The multi-objective optimization stands as a decision-making support tool, helping the design engineer make an informed decision. The distributed computation is employed to reduce the overall time of the optimization process.

Engineering systems such as railway traction systems are too complex to be addressed as a whole. The decomposition-based optimization strategies are intended to address the optimal design of such systems. Model, discipline or object-based decomposition approaches intend to distribute the computational burden across the system. Multi-level coordination strategies are used to manage the optimization process. Each team of specialists can work independently at the object of their expertise. The metamodel-based optimization techniques can be integrated within the multi-level decomposition-based strategies, reducing the computational burden.

The optimization approaches developed within this work are applied for solving several electromagnetic optimization problems and a railway traction system optimal design problem of the Alstom Company.

Keywords

Optimal design of electromagnetic devices
Metamodel-based optimization
Decomposition-based complex systems optimal design
Multi-level optimization
Railway traction system

À ma famille

Remerciements

Ce travail a été réalisé dans le cadre d'un projet de recherche – « OPtimisation de SIMulations » (OPSIM), au sein du « Laboratoire d'Electrotechnique et d'Electronique de Puissance de Lille » (L2EP) de l'Ecole Centrale de Lille, en collaboration avec la société Alstom Transport.

Je suis très reconnaissant à mon directeur de thèse, Dr. Frédéric GILLON, Maître de conférences HDR. Ses conseils et ses encouragements constants m'ont aidé à surmonter les périodes les plus difficiles dans la réalisation du travail de cette thèse.

Je tiens à remercier à Professeur Pascal BROCHET, mon co-directeur de thèse et directeur de l'équipe « Optimisation » du L2EP pour le très bon accueil dans son équipe avec une ambiance chaleureuse.

Ce travail n'aurait pas pu être achevé sans la contribution des autres membres de l'équipe du projet, Dr. Stéphane BRISSET, Maître de conférences HDR et Martin CANTEGREL, mon collègue dans le projet.

Je voudrais remercier à Professeur Jean-Louis COULOMB, Professeur Bruno SARENI, Dr. Laurent KRÄHENBÜHL, Directeur de recherche CNRS et Dr. Stéphane VIVIER, Maître de conférences HDR pour leur disponibilité et le temps précieux qu'ils m'ont accordé en tant que membres de mon jury de thèse.

Je tiens à exprimer ma reconnaissance à Mr. Laurent NICOD, Manageur activités R&D chez Alstom Transport. Je remercie également à Mr. Marc DEBRUYNE, Mr. Philippe CARRERE et Mr. Emmanuel BATISTA, ingénieurs Alstom Transport pour leurs conseils et remarques précieuses.

Je tiens à remercier également tous les professeurs et les membres du personnel du laboratoire, je pense en particulier à Professeur Michel HECQUET, Xavier MARGUERON, Xavier CIMETIERE et Simon THOMY pour leur support.

Un grand merci à tous mes collègues du laboratoire, en particulier à Martin CANTEGREL, Jinlin GONG, Nicolas BRACIKOWSKI, Mathieu ROSSI, Ramzi BEN-AYED, Sophie FERNANDEZ, Adrian POP, Pierre CAILLARD, Aymen AMMAR, Dan ILEA, Dmitry SAMARKANOV, Pierre RAULT, Wenhua TAN, Nicolas ALLALI, Matthias FACKAM, Sangkla KREUAWAN, Amir AHMIDI et François GRUSON. Je garderai en mémoire un très beau souvenir du temps passé avec vous à l'Ecole Centrale.

En fin, je suis fortement reconnaissant à ma grand-mère, mes parents et ma copine pour leur soutien et leurs constants encouragements tout au long de ces années de thèse. Ils ont toujours été là pour moi et sans eux, tout cela n'aurait été possible.

Contents

List of Figures

List of Tables

Symbols and notation

Abbreviations

AAO	All-At-Once
AIO	All-In-One
AL	Augmented Lagrangian
AL-AD	Augmented Lagrangian with the Alternating Direction method of multipliers
ALC	Augmented Lagrangian Coordination
ANOVA	Analysis of Variance
ATC	Analytical Target Cascading
BLISS	Bi-Level Integrated System Synthesis
CAD	Computer-Aided Design
CAE	Computer-Aided Engineering
CFD	Computational Fluid Dynamics
CO	Collaborative Optimization
CO-OSM	Collaborative Optimization with Output Space Mapping
CSSO	Collaborative Sub-Space Optimization
DOE	Design of Experiments
EGO	Efficient Global Optimization
EI	Expected Improvement
EMM	Electromagnetic model
EMM_0	No-load electromagnetic model
ESS	Energy Storage System
EV	Expected Violation
FE	Finite Element
FEA	Finite Element Analysis
FEM	Finite Element Model
FPI	Fixed Point Iteration
GEI	Generalized Expected Improvement
GA	Genetic Algorithm
IDF	Individual Disciplinary Feasible
L	linear penalty
LHS	Latin Hypercube Sampling
LIM	Linear Induction Motor
LP	Lagrangian relaxed problem

MBDO	Metamodel-Based Design Optimization
MDF	Multi-Disciplinary Feasible
MDO	Multi-Disciplinary Optimization
MEGO	Multi-objective Efficient Global Optimization
MOP	Multi-objective Optimization Problem
Multi-EGO	Multi-objective Efficient Global Optimization algorithm of Jeong and Obayashi
NSGA-II	Non-dominated Sorting Genetic Algorithm II
OSM	Output Space Mapping
ParEGO	Pareto Efficient Global Optimization
PD	Pseudo-Distance criterion
PF	Probability of Feasibility criterion
PI	Probability of Improvement criterion
PSO	Particle Swarm Optimization
Q	quadratic penalty
RBF	Radial Basis Function
RS	Response Surface
RSM	Response Surface Methodology
SM	Space Mapping
SMES	Superconducting Magnetic Energy Storage device
SOP	Single-objective Optimization Problem
SQP	Sequential Quadratic Programming
STEEM	Maximized Energy Efficiency Tramway System
TEAM22	optimization benchmark problem of the SMES device
TH	thermal discipline
THM	thermal model
UC	Ultra-Capacitor
UC-ESS	Ultra-Capacity Energy Storage System

Symbols

$\|\cdot\|_2^2$	square of the l_2 norm of a vector
$\circ$	term-by-term multiplication of vectors
α^k	scalar variable representing the step size for the update of the Lagrangian multipliers within ATC
$\mathbf{a}_{ij}$	analysis model employed with the P_{ij} sub-problem within the ATC structure
$\hat{\mathbf{a}}_{ij}$	metamodel approximation of the analysis model employed with the P_{ij} sub-problem within the ATC structure
β	constant parameter for the update of $\mathbf{w}$ coefficients within the AL relaxation of ATC
C_{ij}	the set of children of the P_{ij} sub-problem within the ATC structure
$\mathbf{c}_{ij}$	consistency constraint function vector of the sub-problem P_{ij} within the ATC structure
$\delta^{(k)}$	convergence measure at iteration k of the ATC optimization process
D_d	dominance distance
D_n	neighboring distance

$d_{i,j}$	Euclidian distance between designs *i* and *j* in the objective space
ϵ	tolerance value
ϵ_{ij}^{R}	tolerance value for the coupling variables of sub-problem P_{ij} target matching in ATC
ϵ_{ij}^{y}	tolerance value for the shared variables of sub-problem P_{ij} target matching in ATC
$\boldsymbol{\varepsilon}$	inconsistency tolerance vector representing the allowed level of inconsistency between the different disciplines within the CO formulation
F	objective functions vector
f	objective function
f_c	objective function value computed using the coarse model
f_f	objective function value computed using the fine model
$\hat{f}_{ij}$	metamodel estimation of the objective function of the P_{ij} sub-problem within the ATC structure
f_k	*k*-th objective function
$f_i^{(s_+)}$	value of the *i*-th objective function of the point from the current Pareto front, closest to the trial point
$f_i^{(s_j)}$	value of the *i*-th objective function of the *j*-th point of the current Pareto front, which is dominated by the trial point
f_{min}	minimum value of the objective function from a set of designs
g_c	inequality constraint function value computed using the coarse model
g_f	inequality constraint function value computed using the fine model
g_i	*i*-th inequality constraint function
g_{exp_i}	*i*-th expensive inequality constraint function
g_{inexp_i}	*i*-th inexpensive inequality constraint function
$\mathbf{g}$	vector of inequality constraint functions
h_c	equality constraint function value computed using the coarse model
h_f	equality constraint function value computed using the fine model
h_j	*j*-th equality constraint function
h_{exp_k}	*k*-th expensive equality constraint function
h_{inexp_k}	*k*-th inexpensive equality constraint function
$\mathbf{h}$	vector of equality constraint functions
J_i	objective function of the *i*-th disciplinary optimization
J_i^*	optimal objective function of the *i*-th disciplinary optimization
$\mathcal{L}$	list of evaluated designs
$\mathcal{L}_{ij}^{(k)}$	list of support points for the metamodel construction at the P_{ij} sub-problem at iteration *k* of the ATC optimization process
Λ	list of weighting coefficients vectors
λ	vector of weighting coefficients
Φ	normal cumulative distribution function
ϕ	normal probability density function
$\wp$	Pareto front
$\wp_{found}$	Pareto front found by the optimization algorithm
$\wp_{true}$	true Pareto front
$\pi(\mathbf{c})$	penalty function for the relaxed AiO problem within the ATC formulation

$\pi_Q(\mathbf{c})$	quadratic penalty function for the relaxed AiO problem within the ATC formulation
$\pi_L(\mathbf{c})$	Lagrangian penalty function for the relaxed AiO problem within the ATC formulation
$\pi_{AL}(\mathbf{c})$	augmented Lagrangian penalty function for the relaxed AiO problem within ATC
$\Psi(\mathbf{v})$	objective function of the Lagrangian relaxed problem (LP) within ATC
$\Psi'(\mathbf{v})$	objective function of the augmented Lagrangian relaxed problem (AL) within ATC
$\Psi^*(\mathbf{v})$	optimal objective function value of the Lagrangian relaxed problem (LP) within ATC
$\Psi'^*(\mathbf{v})$	optimal objective function value of the augmented Lagrangian relaxed problem (AL) within ATC
$\mathbf{R}_{ij}^i$	vector of the coupling variables of sub-problem P_{ij} calculated at the i-th level within the ATC structure
$\mathbf{r}_c$	output vector of the coarse model
$\mathbf{r}_f$	output vector of the fine model
$\mathbf{r}_{ij}$	response variables vector of the sub-problem P_{ij} for its parent element within ATC
$\hat{\mathbf{r}}_{ij}$	metamodel estimated response variables vector of the sub-problem P_{ij} for its parent element within ATC
$\hat{s}$	estimated standard deviation value of the metamodel prediction
$\Im$	set of infill designs
$\boldsymbol{\theta}$	vector of correcting coefficients for the OSM technique
$\theta^{(k)}$	design consistency measure at iteration k of the ATC optimization process
$\mathbf{t}_{ij}$	target variables vector of the sub-problem P_{ij} for its child elements within ATC
$\mathbf{v}_{ij}^T$	Lagrangian multipliers vector of the sub-problem P_{ij} within the ATC formulation
w_l	l-th weighting coefficient
$\mathbf{w}$	vector of weighting coefficients
$\mathbf{w}_{ij}$	vector of weighting coefficients for the coupling and shared variables of sub-problem P_{ij} within the ATC formulation
w_{ij}^R	weighting coefficients vector of the coupling variables of sub-problem P_{ij} within ATC
w_{ij}^y	weighting coefficients vector of the shared variables of sub-problem P_{ij} within ATC
x_k	k-th design variable
x_k^l	lower bound for the k-th design variable
x_k^u	upper bound for the k-th design variable
$\mathbf{x}$	vector of design variables
$\mathbf{x}^*$	optimal value of the design vector
$\bar{\mathbf{x}}$	normalized value of the design vector
$\mathbf{x}_{ij}$	local design variables vector of the sub-problem P_{ij} within the ATC structure
$\bar{\mathbf{x}}_{ij}$	local and exchanged variables vector of the sub-problem P_{ij} within the ATC structure
$\tilde{\mathbf{x}}_{ij}$	design variables vector for the analysis model of sub-problem P_{ij} within ATC structure
$\mathbf{y}_{ij}^i$	vector of the shared variables of sub-problem P_{ij} calculated at the i-th level within the ATC structure
$\hat{y}$	metamodel prediction value of the output variable y
$\mathbf{z}_i$	system-level design variable vector for the i-th sub-system within the CO framework
$\mathbf{z}_i^*$	local copies of the system-level design variable vector $\mathbf{z}_i$ within the CO framework

Introduction

The classical design process traditionally employed by engineers, in whom a trial-and-error approach is used for selecting and validating the designs to be conceived, no longer responds to the ever-growing demands in terms of short deadlines, financial budget limitations, resources reduction and "environmental-friendliness" requirements dictated by the concurrence in today's globalized market context. "Better, faster, cheaper!" is the slogan which best describes the requirements which guide today's and tomorrow's industrial design processes. Proof of these statements stand the numerous recent industrial research studies in diverse domains of engineering, among which the most notable are represented by the aeronautics, automobile, electronics and chemical industry.

To all the issues previously invoked and the continuously growing complexity of the design task performed by engineers, the **optimization** comes as a natural answer by assisting the designer in the process of **decision making**. Numerous optimization techniques have been developed over the time, especially during the last two decades, when the computational resources have known an exponential development. Nowadays, computer-aided design (CAD) and computer-aided engineering (CAE) software represent powerful analysis tools, which are employed in all domains of the engineering, such as electrical, mechanical, thermal, acoustics, vibratory etc. for simulating the behavior of the devices or systems to be conceived. These simulation codes act as virtual prototypes of the devices to be conceived. The natural tendency is to introduce such **simulation tools** into the **optimization process**, in order to benefit from the accuracy offered by these tools. However, the **integration** of such simulation tools within the optimization process raises a number of issues which need to be addressed, mainly due to the prohibitive amount of time required by the numerous model calls of classical optimization algorithms. The large number of specific feasibility constraints and conflicting objectives which require being accounted for within the design process encumber the optimization task. Also, a more **systemic optimal design approach** is desired, since strong interactions are exercised between the components of a system. The multi-disciplinary nature of the **complex systems** requires accounting for all disciplines involved into the design process. For addressing all these issues, adapted optimization techniques are required and the development of such optimization approaches makes the subject of this research work. The focus in this manuscript is thus set on the complex system design approaches, which find their application in many domains of engineering in general and in railway systems in particular.

The manuscript is organized into three chapters, as follows. The first chapter introduces the context for the research work and presents the general aspects of the complex system optimal design methodology. The single-objective and the multi-objective mathematical formulations of an optimization problem are introduced and there are underlined some important notions and

concepts which are later used within the optimization techniques and approaches developed and presented in the following chapters of the manuscript. The optimization process is only a phase, the core, of the three-step optimal design process. Two other steps, equally important, a preliminary model analysis phase and a results interpretation and analysis, finalized with the decision making must also be accounted for within the optimal design process of complex systems.

The second chapter of the manuscript addresses a specific category of optimization techniques relevant for complex engineering design problems – the **metamodel-based design optimization** approach. The **simulation codes** (FEA, CFD, etc.) are **accurate tools** for simulating the behavior of the device, allowing to account for complex physical phenomena which cannot be captured through analytical relations. The increased accuracy of these tools comes for the price of long computational time, a single evaluation of such a code taking between minutes, up to hours and even days, depending on the complexity of the model and the type of analysis required. The **integration of** such **expensive simulations** within a classical optimization process is thus prohibitive, due to the great number of successive calls of the optimization algorithm to the simulation model. For devices or systems represented by such simulation codes, a common practice consists in creating **metamodels**, which benefit of a **fast evaluation**. The main issue with the metamodels represents the accuracy of these representations. The integration of metamodels into the optimization process is discussed in this chapter and the purposes are sustained through both mathematical analytical test problems and a well-known electromagnetic benchmark problem. These optimization techniques find application in the last chapter of the manuscript, which addresses the design optimization at a larger extent, considering a system as an ensemble of models or components.

The third chapter addresses the **complex system optimization** through means of **decomposition** of the system following different perspectives and the associated specific coordination techniques of the optimizations formulated for each element of the decomposed structure. Such optimization approaches represent a current practice in the automobile and aerospace industries, where they were developed and employed for more than a decade. The motivation of analyzing such optimization techniques for **electromagnetic and railway applications** comes from the great success that they have shown in the domains where they were introduced, showing a great potential. Several decomposition perspectives are reviewed, based on different models of the same device, the disciplines involved in the representation of the system and the different physical sub-systems and components of the complex system. First, the optimization of model-based decomposed systems is introduced and addressed using a Space Mapping technique, **Output Space Mapping** (OSM). A single-phase safety isolation **transformer benchmark** is introduced as application for this optimization technique. The OSM optimization approach makes use of two models of the same device, one being a high fidelity finite element model (3D FEM) and the other one a fast evaluation coarse analytical model with a limited fidelity. The synergies of the two models are exploited by the OSM algorithm, retrieving the optimal design with a much reduced call to the expensive high fidelity FE model of the transformer. Another decomposition-based optimization approach considers the complex system by the different disciplines involved in analyzing the system. The most representative optimization technique of this category, the **Collaborative Optimization** (CO) technique is introduced and its performance is analyzed through application to the same transformer benchmark optimization problem previously mentioned. Another decomposition-based optimization technique, this time considering the complex system through its sub-systems and components is analyzed. The **Analytical Target Cascading** (ATC) is an

optimization technique which addresses the system decomposed into its sub-systems and components, which are hierarchically displaced onto several levels, starting with the most global at the top of the hierarchy and the smallest component placed at the bottom level of the hierarchy. This technique was originally introduced as a specifications formalizing method, to cascade the system-level targets down the hierarchy, to the most basic components. This optimization approach was already found to cope very well with the hierarchical organization of the **Alstom Company** and the way the system design specifications are imposed in the different departments of the company. The previous research studies of Moussouni [MOU 09a] and Kreuawan [KRE 08] have analyzed the basic configuration of this technique. The exploration of this technique is continued within this research study by going further into the functional mechanisms of the technique. A multi-level **railway test-problem** of Alstom Company, consisting in the optimal dimensioning of an ultra-capacity energy storage system onboard a tramway, is introduced and used here as application of the ATC technique.

Chapter 1 Decision support tools for complex electromagnetic systems design

In this chapter, the optimal design process is introduced and the general aspects of the complex system optimal design are addressed in general lines. The optimal design process is regarded as a three-step process, consisting of a preliminary phase of problem definition, the main phase of optimization problem solving and a result analysis and decision making phase. The basic mathematical formulations of both the single- and the multi-objective optimization are introduced and the notion of optimality in both the single- and the multi-objective context is discussed. Some classical both deterministic and stochastic optimization algorithms are briefly presented for solving single-objective optimization problems. Several techniques for transforming a multi-objective optimization problem into a single-objective problem are reviewed, which will be used later on within the advanced optimization techniques developed in the manuscript. A number of multi-dimensional data representation techniques, meant to assist the designer with the decision making, are presented towards the end of the chapter.

1.1 General aspects of an optimal design process

The optimal design approach of a device, product or system implies three main steps:

- Preliminary phase (problem formulation);
- Optimization process (algorithm run);
- Results visualization and analysis (decision making).

These sequential steps are strongly related one to another and the result of the optimal design process depends on the appropriate addressing of all three steps. The brief description of these steps is presented next.

Preliminary phase (optimization problem formulation)

The preliminary phase of the optimal design approach consists in acquiring and resembling all the necessary information about the device, system or process to be optimally designed. In this phase, the designer tries to gain as much information as possible about the object of his design, for this will help him to appropriately define the optimization problem to be solved. The result of the optimization process depends entirely on the formulation of the optimization purpose, therefore

the proper formulation of the optimization problem is crucial for the success of the optimal design process. A number of issues are addressed within this step, such as: the model (or models) of the device which are to be used within the optimization process, the parameters and design variables of the model, the type of design variables (continuous, discrete, unclassifiable), the domain of variation for the design variables, the output variables of the model, the type of optimization problem (linear, non-linear), etc.

An important action at this stage consists in determining the kind of relationships existing between the input, output and input/output variables of the model, as well as the relative influence of the input variables over the outputs of the model (sensitivity analysis). The influence of the different design variables on the outputs of the considered model are analyzed with the goal of identifying those variables which do not (or have very little) influence upon the outputs of the model, so that they will be ignored in the formulation of the optimization problem. This way, the optimization algorithm is charged with determining the optimal values of only the influent parameters, thus alleging more or less considerably the task of optimization and saving important computational time. A few tools for the analysis of the functional relationships governing a model are presented later on, in paragraph 1.2.

Optimization process (algorithm run)

The second step of the optimal design process is represented by the optimization process itself, which represents the core of the optimal design process. Complex mathematical techniques are employed at this stage for solving the optimization problem previously formulated. During the last two decades, once with the strong development of the computational power and the appearance of the personal computers, a lot of research has been dedicated to the development of optimization algorithms and mathematical optimization techniques. These optimization algorithms make successive calls to the model of the device considered, selecting then new samples to be evaluated based on the values already calculated.

Depending on the type of optimization problem previously formulated, an appropriate optimization algorithm is selected to solve this problem. The optimization problems can be classified based on several different criteria. Thus, if the objective function can be expressed as a linear combination of the design variables, we deal with a linear optimization problem or linear programming. Otherwise, the problem is said to be non-linear, solved using non-linear programming techniques (NLP). If the design variables of the problem take discrete values, the optimization problem is discrete or combinatorial. A number of combinatorial techniques with application in electrical engineering are presented in [TRA 09]. If the domain of variation of variables is continuous, then the optimization problem is continuous. It often arrives that an optimization problem presents both continuous and discrete variables. In this case, it is employed the term of mixed-integer optimization.

The optimization techniques developed and presented further on in Chapter 2 and Chapter 3 of this manuscript belong to the category of NLP techniques. Although several optimization applications addressed in this work have both discrete and continuous variables, in this work all variables of the applications are considered continuous and handled accordingly.

Results visualization and analysis (decision making)

The third and final step of the optimal design process is represented by the analysis and interpretation of the results supplied by the optimization process mentioned in the previous paragraph. When a multi-objective formulation has been chosen for the optimization problem, the result is a set of optimal trade-off designs between the considered objective functions. In this case, the final choice for the design to be considered remains at the latitude of the designer, engineers and/or managers, which will select one design to be conceived. Thus, the multi-objective optimization represents a tool for decision-making. A number of tools for the representation and visualization of the results, such as bar charts, spider diagrams, bubble plots, parallel coordinates representations, scatter plot matrix, etc. [EST 12], [NOE 12] have as goal to assist the designer with the decision-making process. Some of these multi-dimensional data representation techniques will be introduced later on in this chapter, in paragraph 1.4.2.

Now that the steps of the optimal design process were briefly described, the attention is turned to each of them at a time, starting with some means of analyzing the correlation between the variables of a model, in the preliminary phase of the optimal design process.

1.2 Variables influence and correlation

An important number of statistical analysis tools exist, assisting the designer in obtaining insights into the functional relationships between the variables of a model. Model analysis techniques and tools such as the screening technique [VIV 02], analysis of variance (ANOVA) [GOU 06], and response surface methodology are based on the realization of different experimental designs and represent notorious statistical tools. Therefore, these techniques will not be addressed here. Nevertheless, some techniques for the qualitative analysis of models, such as the Pearson correlation coefficient and the Spearman correlation coefficient represent powerful tools for gaining insights into the functional relationships of a model, being less employed currently. These coefficients will be next presented in the following paragraphs.

Pearson correlation coefficient

The Pearson correlation coefficient is a measure for the linear relationship between two input/output variables of a model [MAS 06]. The value of the coefficient is calculated based on a set of n samples (x_i, y_i), for which the model has been run. Such a set of samples can be obtained using a design of experiments (DOE) technique [GOU 06], also known as experimental design. The value taken by this coefficient is always between -1 and 1. If the graphical representation of all pairs of samples is a straight line, it means that there is a strong linear correlation between the two variables. The slope of the straight line gives the sense of the correlation: positive slope gives positive value for the Pearson coefficient, respectively negative slope corresponds to a negative value of the Pearson coefficient. A value of zero for the Pearson correlation coefficient signifies that no linear correlation exists between the corresponding pair of variables of the model.

The expression of the Pearson correlation coefficient between variables x and y is given in (1.1).

$$\rho_P(x,y) = \frac{Cov(x,y)}{\sigma_x \sigma_y} \tag{1.1}$$

where σ_x represents the standard distribution of the values of the variable x from the experimental design, σ_y represents the standard distribution of the values of y and $Cov(x,y)$represents the covariance between the two variables over the experimental design considered.

The expression of the covariance $Cov(x,y)$ between the two variables x and y is given in (1.2).

$$Cov(x,y) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) \tag{1.2}$$

where $\bar{x}$ represents the mean value of the variable x over the experimental design considered and $\bar{y}$ represents the mean value of y.

Spearman rank correlation coefficient

The Spearman rank correlation coefficient [NAK 09], also known as Spearman's rho [EST 12] is a non-parametric statistical measure of the monotonicity of a function.

The first step in calculating the Spearman correlation coefficient consists in assigning to each value of the variables from the experimental design a rank, following a descending order, i.e. $rank(x_i) = 1$ where $x_i = \max_j(x)$, $j = 1, \cdots, n$.

As in the case of the Pearson coefficient, the value taken by the Spearman correlation coefficient is always between -1 and 1. A value close to 1 for this coefficient signifies that there is a strong direct correlation between the two variables, while a value close to -1 indicates a strong inverse correlation between the two variables. If the value of this coefficient is close to zero, then it means that no correlation exists between the two variables. However, in this latter case, a non-monotonic correlation may exits.

The expression of the Spearman correlation coefficient is given in (1.3).

$$\rho_S(x,y) = \frac{\sum_{i=1}^{n}\left(\left(rank(x_i) - \overline{rank}(x)\right)\cdot\left(rank(y_i) - \overline{rank}(y)\right)\right)}{\sqrt{\sum_{i=1}^{n}\left(rank(x_i) - \overline{rank}(x)\right)^2 \cdot \sum_{i=1}^{n}\left(rank(x_i) - \overline{rank}(x)\right)^2}} \tag{1.3}$$

where $\overline{rank}(x)$ represents the mean value of the ranks of all values of the x variable, calculated according to (1.4).

$$\overline{rank}(x) = \frac{1}{n}\sum_{i=1}^{n} rank(x_i) \tag{1.4}$$

Correlation matrix

The correlation matrix or correlation chart [EST 12] represents a graphical tool for visualizing the values of different correlation coefficients. Once a correlation coefficient (Pearson, Spearman or other) is calculated for every pair of input/output variables of a model, these values can be represented under a matrix form. This way, the strong correlations between variables can be easily

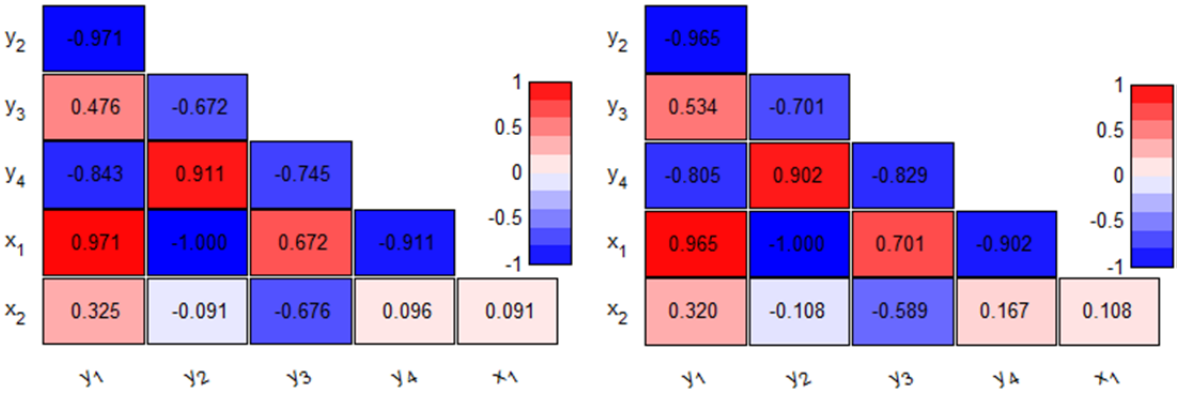
detected, as well as variables which present a negligible correlation, being therefore considered uncorrelated. A correlation matrix tool, inspired from the correlation chart of the modeFRONTIER commercial optimization software product [EST 12], has been developed under Matlab®.

To exemplify the purpose of the correlation coefficients, a simple model consisting of four mathematical analytical relations depending on two variables is considered in (1.5).

$$\begin{cases} y_1 = 4x_1 + x_2 - 4 \\ y_2 = -x_1 - 1 \\ y_3 = x_1 - x_2 - 2 \\ y_4 = (x_1 - 4)^2 + (x_2 + 1)^2 \end{cases} \quad \text{with} \quad x_1, x_2 \in [-4,4] \tag{1.5}$$

For the model expressed in (1.5), a design of experiments of 20 designs has been considered using the Latin Hypercube Sampling (LHS) methodology. The LHS technique, introduced by McKay et al. in [MCK 00] is very popular experimental design for computer experiments [KRE 08].

The correlation coefficients for the model considered in (1.5) have been calculated and represented using the correlation matrix shown in Figure 1.1.



a) Pearson correlation coefficients b) Spearman correlation coefficients

Figure 1.1 : Correlation matrix

Similar values for the two correlation coefficients, Pearson and Spearman, have been obtained, as can be seen from Figure 1.1a and Figure 1.1b. In complement to the correlation coefficients, the strong correlations are represented using dark blue or red colors (for negative, respectively positive correlations), while the low correlations have a faded color.

From these two correlation charts, it can be seen the level of correlation between the variables of the model (x_1, x_2) and the outputs (y_1, y_2, y_3, y_4). For example, a complete linear inverse correlation is immediately detected for the couple (x_1, y_2). Also, a strong linear correlation is observed between y_1 and x_1, which is logic if we take a look at the expression of y_1 in (1.5). Another remark is that although both x_1 and x_2 variables are present in the expression of y_4, due to the quadratic terms, x_1 is almost completely correlated with y_4, while no correlation is detected between y_4 and x_2. The

correlation matrix presents itself as an easy mean of empirically validating the tendencies of the functional relationships of a model.

Once reviewed the correlation matrix as a tool for the empirical model validation prior to the optimization process run, the basic mathematical formulations of both the single- and the multi-objective general optimization problems are introduced next.

1.3 Single and multi-criteria optimization

Most of the real life optimization problems are multi-objective by their nature. However, the optimization problem can be expressed using a unique objective (single-objective optimization) or several criteria can be accounted for within the expression of the optimization problem. The optimization problems may or may not present constraint functions (constrained or unconstrained optimization). There are next introduced some elementary notions about a single-objective optimization problem formulation.

1.3.1 Single-objective optimization

The general mathematical formulation [VEN 01] of a single-objective constrained optimization problem is expressed in (1.6).

$$\begin{aligned} \underset{\mathbf{x}}{\text{Minimize}} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & g_i(\mathbf{x}) \leq 0 \qquad i = 1, \cdots, n_i \\ & h_j(\mathbf{x}) = 0 \qquad j = 1, \cdots, n_e \\ \text{with} \quad & \mathbf{x} = [x_1, \cdots, x_k, \cdots, x_n] \qquad k = 1, \cdots, n \\ & x_k^l \leq x_k \leq x_k^u \end{aligned} \tag{1.6}$$

where $\mathbf{x}$ represents the vector of design variables, each variable x_k being defined between a lower and an upper bound, x_k^l respectively x_k^u, also known as "box constraints", f represents the objective function to be minimized[1], g_i represents the *i*-th inequality constraint function, h_j represents the *j*-th equality constraint function, n represents the number of design variables, n_i represents the number of inequality constraint functions and n_e represents the number of equality constraint functions of the optimization problem.

Within the formulation of an optimization problem, the equality and/or inequality constraint functions might be absent. If both equality and inequality constraint functions are lacking, the optimization problem is said to be "unconstrained". However, rare is the case in practical applications where no constraint functions are formulated within the optimization problem.

[1] The convention generally adopted by the research community expresses any optimization problem as a minimization problem. When the optimization problem implies an objective function to be maximized, this is represented within the general formulation of the optimization problem by using the minus sign ($\text{Maximize}_{\mathbf{x}} f(\mathbf{x}) \equiv \text{Minimize}_{\mathbf{x}} -f(\mathbf{x})$).

1.3.2 Local vs. global optimality

Two different types of optima exist in the resolution of a nonlinear optimization problem. These are illustrated graphically using an abstract single-dimensional continuous example function $f(\mathbf{x})$ presented in Figure 1.2.

A point $\mathbf{x}^*$ is a local minimum of the function $f(\mathbf{x})$ if the expression (1.7) is valid [MES 07].

$$f(\mathbf{x}^*) < f(\mathbf{x}) \qquad \forall \mathbf{x} \in N(\mathbf{x}^*), \mathbf{x}^* \neq \mathbf{x}, N(\mathbf{x}^*) \subset D(\mathbf{x}), D(\mathbf{x}) \subset \mathbb{R}^n \tag{1.7}$$

where $N(\mathbf{x}^*)$ is a subdomain of $D(\mathbf{x})$, the domain of definition of the $f(\mathbf{x})$ function, defining the neighborhood of the point $\mathbf{x}^*$.

The function $f(\mathbf{x})$ may have several local optima (minima).

A point $\mathbf{x}^*$ is a global minimum of the function $f(\mathbf{x})$ if the expression (1.8) is respected.

$$f(\mathbf{x}^*) \leq f(\mathbf{x}) \qquad \forall \mathbf{x} \in D(\mathbf{x}),\ \mathbf{x}^* \neq \mathbf{x}, D(\mathbf{x}) \subset \mathbb{R}^n \tag{1.8}$$

As in the case of the local optima, several global optima (minima) of the function $f(\mathbf{x})$ may exist. The uniqueness of the global solution is guaranteed if the relation (1.9) is respected.

$$f(\mathbf{x}^*) < f(\mathbf{x}) \qquad \forall \mathbf{x} \in D(\mathbf{x}),\ \mathbf{x}^* \neq \mathbf{x}, D(\mathbf{x}) \subset \mathbb{R}^n \tag{1.9}$$

Hence, the global optimum of a function is necessarily also a local optimum of the function. The reverse statement is obviously not true.

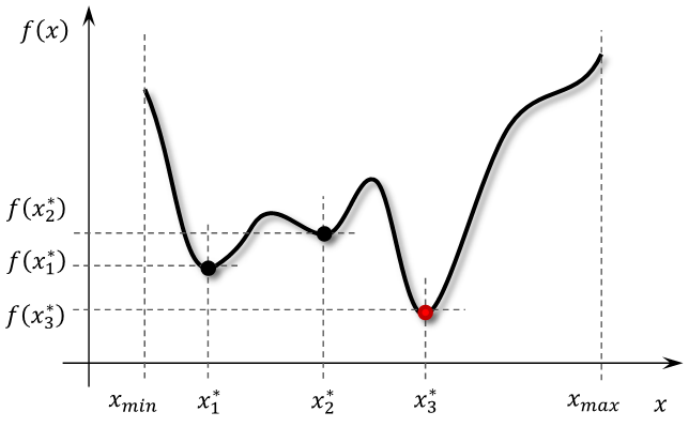


Figure 1.2 : Local vs. global optimality representation

The example function $f(x)$ presented in Figure 1.2 shows three distinct local optima (minima), x_1^*, x_2^* and x_3^*. Among these local minima, the function $f(x)$ presents only one global minimum, x_3^*, since $f(x_3^*) < f(x)$, $\forall x \in [x_{min}, x_{max}]$.

The main difficulty faced by an optimization algorithm is to avoid being caught in a local basin of attraction of the function to be optimized.

1.3.3 Deterministic optimization algorithms

A number of classical deterministic optimization algorithms have been developed over the last half century. These are gradient-based optimization algorithms, basing their behavior on the available or estimated information about the gradients of the objective and constraint functions. The gradient information is used by these algorithms to guide the search by mathematically determining the basins of attraction of the search.

These algorithms require specifying an initial design as start point and the result of the optimization process depends on the initial design considered. These algorithms produce a local search and if the model presents several basins of attraction (i.e. the problem is multi-modal), there is an important possibility of getting stuck in a local basin of attraction, therefore producing an optimal design which is not the global optimum of the problem.

Among these algorithms, the Sequential Quadratic Programming (SQP) represents one of the most notorious and most employed algorithms for solving single-objective constrained optimization problems with continuous functions. An implementation of this algorithm can be found in the greatest majority of commercial optimization software. A detailed presentation of this and other algorithms alike, with public Matlab® codes for testing different problems is given by Venkataraman in [VEN 01]. Practical application issues of the gradient-based SQP algorithm for solving electromagnetic problems with finite element models can be found in [MES 07]. A Matlab® implementation of the SQP algorithm is coded under the "*fmincon*" function, available in the Optimization Toolbox. Other deterministic approaches, such as interval methods and real algebraic geometry-based methods are equally employed for solving nonlinear optimization problems.

Variable's normalization

In practical optimization applications, it often arrives to encounter design variables with different scales. Such scale differences may impact on the result of the optimization process employing a gradient-based optimization algorithm such as SQP [GIL 09]. Hence, precautions must be taken prior to the optimization process launch by making all design variables uniform. The most common way of bringing all design variables of the problem to the same scale is to normalize the design variables. Several techniques are available for normalizing the design variables, which can be found in [MES 07]. The most often employed technique consists of transforming the vector of all design variables such that their values lie within the interval [0,1].

For this transformation, the mathematical formulation is given in (1.10).

$$\bar{\mathbf{x}} = \frac{\mathbf{x} - \mathbf{x}^l}{\mathbf{x}^u - \mathbf{x}^l} \tag{1.10}$$

where $\bar{\mathbf{x}}$ represents the normed or normalized design vector and $\mathbf{x}$ represents the vector of real or true design variable values.

The reverse transformation, from normalized to true variable values, is expressed in (1.11).

$$\mathbf{x} = \bar{\mathbf{x}} \circ (\mathbf{x}^u - \mathbf{x}^l) + \mathbf{x}^l \tag{1.11}$$

This normalization might also show useful for the constraint functions and – in a multi-objective context – for the objective functions of an optimization problem.

Multi-run process

In order to ensure the global optimality of the result, a common practice in the optimization using gradient-based algorithms consists in employing a multi-run process. As the result of the optimization using a gradient-based approach is highly dependent upon the initial design considered, the multi-run or multi-launch process consists in several (usually between 10 and 100) successive or simultaneous optimization algorithm runs, with different initial design points for each of the runs. Thus, at the end of all optimization algorithm runs, the best result is retained with the hope that it represents the global optimum of the problem. This fact is not guaranteed, but however, the larger the number of optimization runs of the multi-start, the greater the probability that the global optimal design is determined.

An important obvious drawback of the multi-run approach consists in the very important computational expense required by all the optimization runs.

1.3.4 Stochastic optimization algorithms

The stochastic optimization methods represent optimization approaches which are based on the realization of a random process. These approaches are meant to overcome the drawbacks faced by the deterministic methods, such as the possibility of finding a local optimum solution and which is strongly related to the initial configuration. Compared to the deterministic approaches, which offer the same solution for two algorithm runs starting from the same initial design configuration, the stochastic approaches are not expected to return the same result from identical conditions. This feature is related to the random process underlying the stochastic algorithm. Nevertheless, most of the commercial optimization software offers the possibility of obtaining the same results for different runs of a stochastic optimization algorithm with the same settings, through the means of a controlled pseudo-random process.

The stochastic approaches are also known as derivative-free or direct search algorithms. These algorithms do not require any information about the gradients of the objective and/or constraint functions, thus being suited for a larger category of optimization problems. The underlying mechanism of a stochastic algorithm is inspired from different processes in nature, such as biological evolutionary processes (such as reproduction, mutation, natural selection etc.), animal behavior (e.g. flocking, swarm behavior, etc.) and others. In general, these are population-based algorithms, thus requiring a large number of model evaluations.

Stochastic algorithms are global optimization methods, meaning that they seek to obtain not just a local optimum, but the global optimum of the problem, through exploration of the entire design space. In compare to the gradient-based approaches, which obtain local optima with high precision, the solutions offered by the stochastic algorithms are less accurate, but with more chances of being the global optimum of the problem. Once a solution is obtained by the stochastic optimization process, a gradient-based optimization algorithm can then be launched starting from the optimum obtained, in order to refine the values of its design variables. This technique is known as the hybrid approach, combining both features in the search for the accurate value of the global optimum.

Some of the most notorious stochastic optimization algorithms are represented by the Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO).

A Matlab® implementation of a genetic algorithm is available under the "*ga*" function of the Global Optimization Toolbox. The basic principles of the genetic algorithms are inspired from nature, simulating the natural evolution process by employing two genetic operators – crossover and mutation and a natural selection process. These are population-based optimization algorithms, usually elitist, i.e. the best obtained solutions are always kept for continuing the evolution process.

1.3.5 Multiple criteria optimization

Most of the real-world optimal design problems are multi-objective by nature. General engineering design optimization criteria such as mass reduction, efficiency improvement, environmental impact reduction are obviously antagonistic and must be all accounted for within an optimal design process. In mathematical terms, the general formulation of a multi-objective optimization problem (MOP) is expressed in (1.12).

$$\begin{aligned} \underset{\mathbf{x}}{\text{Minimize}} \quad & F(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_m(\mathbf{x})] \\ \text{subject to} \quad & g_i(\mathbf{x}) \leq 0 \qquad i = 1, \cdots, n_i \\ & h_j(\mathbf{x}) = 0 \qquad j = 1, \cdots, n_e \\ \text{with} \quad & \mathbf{x} = [x_1, \cdots, x_k, \cdots, x_n] \qquad k = 1, \cdots, n \\ & x_k^l \leq x_k \leq x_k^u \end{aligned} \tag{1.12}$$

where F represents the vector of objective functions and m is the number of objective functions of the optimization problem.

The result of an optimal design process, either single or multi-objective, is a design which best fits the formally and/or informally specified requirements. In the case when multiple optimization criteria are stated in the optimization problem, a compromise solution must be searched among a set of several optimal solutions.

In the following paragraph, some definitions concerning the optimality concept within the multi-objective context are introduced, which lie at the basis of all further multi-objective approaches presented in this manuscript.

1.3.5.1 Pareto optimality

The Pareto optimality represents a measure of efficiency in the multi-objective context [CHI 07], where several conflicting objectives must be accounted for in an optimization process. The notion of Pareto optimality or Pareto efficiency is thus similar to the global optimality in the single-objective case. The name of Pareto comes from the economics domain, taking the name of an Italian economist, Vilfredo Pareto, at the beginning of the 19th century [CHI 07]. A design is considered Pareto optimal or Pareto efficient if there does not exist any other design which improves the value of any of its objective criteria without deteriorating at least one other criterion.

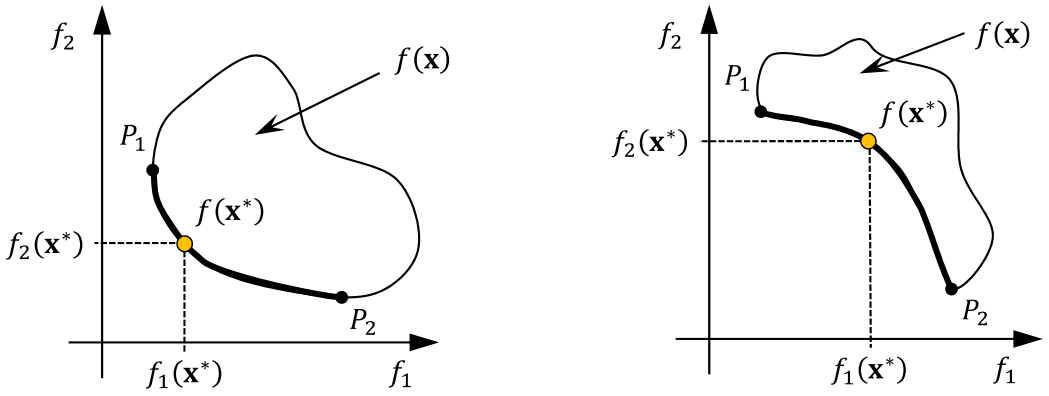
From the mathematical point of view, the definition of the Pareto optimality or Pareto efficiency [CHI 07], [PAP 08] can be expressed as in (1.13).

$$\{\mathbf{x}^* \in D \subset \mathbb{R}^n | \mathbf{f}(\mathbf{x}^*)\} \; is \; Pareto \; optimal \\ \overset{\Delta}{\Leftrightarrow} \left\{\nexists \mathbf{x} \in D | f_i(\mathbf{x}) \leq f_i(\mathbf{x}^*), i = \overline{1,n} \cap f_j(\mathbf{x}) < f_j(\mathbf{x}^*), \forall j \in \{1, \cdots, n\}\right\} \tag{1.13}$$

If a design $\mathbf{x}$, with the properties expressed in (1.13) however exists, then this solution $\mathbf{x}$ is said to dominate $\mathbf{x}^*$ in the sense of Pareto or simply dominate $\mathbf{x}^*$ [ALB 11]. Otherwise, the design $\mathbf{x}^*$ is said to be non-dominated (in the sense of Pareto).

The set of all designs respecting the definition given in (1.13) form the Pareto front or the Pareto frontier, representing thus the optimal trade-off between all objective functions.

In order to present the concept of the Pareto optimality, a generic bi-objective optimization is considered and two possible types of Pareto fronts, a convex and a non-convex front, which are represented graphically in Figure 1.3a, respectively Figure 1.3b.



a) Convex Pareto front b) Non-convex Pareto front

Figure 1.3 : Pareto front representation for a generic bi-objective optimization problem

The bold curve presents the Pareto front in both the convex and non-convex Pareto front cases. The yellow circle presents a non-dominated solution on the Pareto front.

Through analogy with the local optimality in the single-objective case, a local Pareto optimality can be defined also in the multi-objective context. The mathematical definition of the local Pareto optimality is expressed in (1.14).

$$\{\mathbf{x}^* \in D \subset \mathbb{R}^n | \mathbf{f}(\mathbf{x}^*)\} \; is \; locally \; Pareto \; optimal \\ \overset{\Delta}{\Leftrightarrow} \{\exists \delta > 0 | \mathbf{x}^* \; is \; Pareto \; optimal \; in \; D \cap N(\mathbf{x}^*, \delta)\} \tag{1.14}$$

where δ defines the radius of an n-dimensional sphere N centered in $\mathbf{x}^*$.

As in the single-objective case, a Pareto optimal design $\mathbf{x}^*$ is also locally Pareto optimal; the reverse is true only if all the objective functions are convex.

Another notion which requires being introduced is the weak Pareto optimality [CHI 07], [NAK 09]. The mathematical definition of a weak Pareto optimal solution is expressed in (1.15).

$$\{\mathbf{x}^* \in D \subset \mathbb{R}^n | \mathbf{f}(\mathbf{x}^*)\} \; is \; weakly \; Pareto \; optimal \\ \overset{\Delta}{\Leftrightarrow} \{\nexists \mathbf{x} \in D \subset \mathbb{R}^n | f_i(\mathbf{x}) < f_i(\mathbf{x}^*), i = \overline{1,n}\} \tag{1.15}$$

A weak Pareto solution is therefore a design which can be improved in one objective function while maintaining the values for the other objective functions. As stated in [NAK 09], weak Pareto solutions are not desired in the final optimal design decision making, but a number of optimization approaches can guarantee only weak Pareto optimality.

An example of bi-objective problem presenting weak Pareto solutions is given in Figure 1.4.

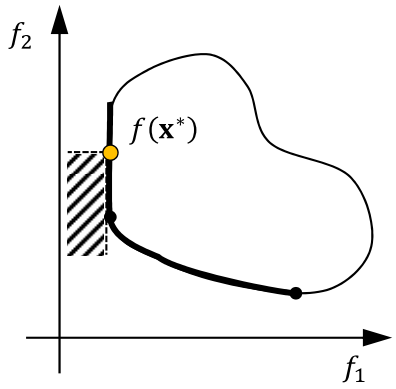


Figure 1.4 : Weak Pareto solutions of the bi-objective generic optimization problem

It can be seen that the design represented through a yellow circle, as well as all designs situated on the same vertical dotted line are weak Pareto designs, but they are not Pareto optimal.

The determination of one or more non-dominated designs of the Pareto front is the subject of all multi-objective optimization approaches. Depending on the moment of decision making in the optimization process, several categories of multi-objective optimization approaches exist, which are reviewed briefly in the following paragraph.

1.3.5.2 Decision making in the optimal design process

The compromise between the different optimization criteria can be fixed prior to the optimization process. Such optimization approaches are called "*a priori*" methods, due to the compromise between the design objectives which is imposed prior to the optimization run. All optimization criteria are expressed under the formulation of a single-objective optimization problem. The result of this optimization problem is a design which is optimal with regard to the given compromise expressed between the different optimization criteria. A number of transformation techniques, allowing the handling of multiple optimization criteria by a single-objective formulation exist in the literature and the most notorious of them are selected for presentation in paragraph 1.3.6 of this manuscript.

Nevertheless, expressing all optimization criteria under the form of a single-objective formulation might not always be possible or desired sometimes. The selection of proper values for the different weighting coefficients associated to different design objective functions involving completely different output variables such as mass, efficiency, consumption, environmental impact etc. cannot be done prior to the optimization run. In this case, the designer, engineers and the managers in a company prefer a set of different trade-off designs between the different objective functions expressed in the multi-objective optimization problem. The final choice for the optimal design remains at their latitude, who will take an informed decision among a set of possible and

equivalent (from the point of view of the optimization problem). These optimization approaches are called "*a posteriori*" methods, giving the fact that the decision for the compromise between objectives is taken after the optimization run.

A third class of optimization approaches consists in the interactive selection of designs. In this case, the designer intervenes at each step of the optimization process, expressing each time his preferences based on the current data placed at his disposal by the optimization process. The designer guides thus the optimization algorithm towards the regions of his interest. However, this type of optimization approach is less current, since it implies the regular and often intervention of the designer in the decision taking within the optimal design process.

1.3.6 Optimization problem transformation techniques

It often arrives, for different reasons, to require rephrasing the formulation of the optimization problem. The formulation of an optimization problem can be transformed in a number of different ways in order to simplify the initial problem or to allow solving it by the means of available optimization tools. Hence, a multi-objective optimization problem can be transformed into a single-objective problem, constraint functions can be transformed into objectives and vice-versa, etc. A taxonomy of transformation techniques can be found in [BRI 07]. In this section, there are addressed only those transformation techniques meaning to reduce the number of objective functions of a problem to a single objective function, also known as scalarization techniques. Some of these basic formulations will be then implemented within the more complex optimization techniques described later on, in the following chapters of the manuscript.

1.3.6.1 Weighted objectives method

The weighting objectives method [COL 02], [MIE 99] is a transformation technique consisting in the aggregation of the m objective functions from (1.12) after associating certain weighting coefficients to each of them, depending on their relative importance.

Linearly-weighted objectives method

The most known and used of the weighted aggregation techniques is the linear aggregation function [MIE 99], [BRI 07]. The formulation of the rephrased optimization problem following the linearly-weighed objectives method is given in (1.16).

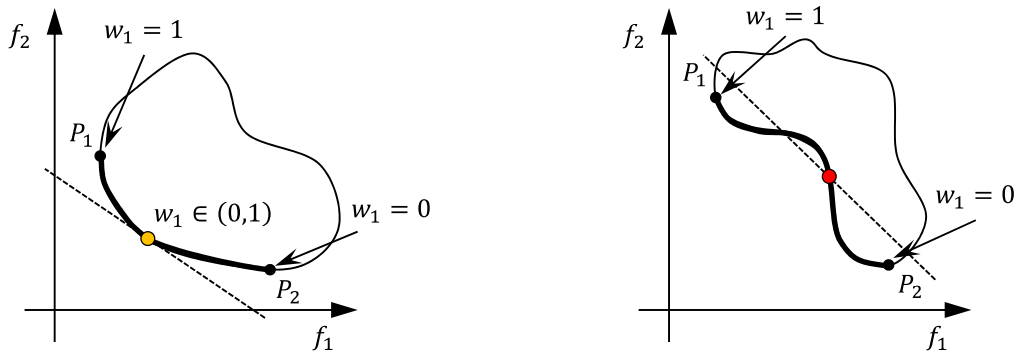
$$\begin{aligned}
&\underset{\mathbf{x}}{\text{Minimize}} && f(\mathbf{x}) = w_1 f_1(\mathbf{x}) + w_2 f_2(\mathbf{x}) + \cdots + w_m f_m(\mathbf{x}) && \sum_{l=1}^{m} w_l = 1 \\
&\text{subject to} && g_i(\mathbf{x}) \le 0 && i = 1, \cdots, n_i \\
& && h_j(\mathbf{x}) = 0 && j = 1, \cdots, n_e \\
&\text{with} && \mathbf{x} = [x_1, \cdots, x_k, \cdots, x_n] && k = 1, \cdots, n \\
& && x_k^l \le x_k \le x_k^u &&
\end{aligned} \tag{1.16}$$

where w_l represents the weighting coefficient associated to the l-th objective function.

An important remark is related to the normalization of the objective functions, which must be done prior to the aggregation [DIB 10]. In order for the aggregation to be efficient, therefore to reflect the desired relative influence of each objective, the different objective functions must be normalized using the same normalization formulations as in the case of the design variables, expressed in (1.10). The normalization must be done with respect to the minimum, respectively maximum known or estimated values of each objective function.

Launching several optimization processes using each time a different set of weighting coefficients allows for determining different solutions on the Pareto front of the initial multi-objective optimization problem.

To exemplify the mechanism of the weighted objectives method, a simple bi-objective optimization case is considered. Two figure-cases are imagined for the shape of the Pareto front of the problem, a convex and a non-convex form being represented graphically in Figure 1.5a and Figure 1.5b, respectively.



a) Convex Pareto front

b) Non-convex Pareto front

Figure 1.5 : Weighted objectives method applied to a bi-objective optimization problem

The two extreme points of the Pareto front are determined in both convex and non-convex shape Pareto front cases by setting each weighting coefficient at a time equal to *1*, hence giving exclusive importance to only one of the objectives. For intermediate values of the weighting coefficients $w_l \in (0,1)$, the corresponding design on the Pareto front is represented by the point on the line of slope $\frac{-w_1}{w_2}$ which is tangent to the Pareto front curve, situated at the exterior of the objective function's domain. Thus, while such a point can be obtained in the convex Pareto front case for any combination of weighting coefficients, the weighting objectives method fails in supplying that point in the case where the Pareto front contains non-convex zones, as can be seen from Figure 1.5b.

The advantage of this method consists in its ease of implementation. Nevertheless, the method presents some stated drawbacks which limit its application. Hence, the most important inconvenient of the method consists in its inefficiency in solving problems which have a non-convex Pareto front. In practical applications, the shape of the Pareto front is not known "*a priori*", thus applying the weighting objectives method might be misleading if the front is non-convex. Also, the selection of proper weighting coefficients for obtaining a uniform distribution of solutions along the Pareto front is difficult.

Weighted objectives method with Tchebyshev scalarization function

To overcome the drawback represented by the inability of the previously-presented weighted objectives method employing linear weights to obtain the Pareto front in the case of optimization problems presenting a non-convex Pareto front, two non-linear weighting functions have been developed [NAK 09]. A first weighting function, the Tchebyshev scalarization function is meant to guarantee that the obtained solution employing the weighted objectives formulation is weakly Pareto optimal [NAK 09].

The mathematical formulation of the Tchebyshev scalarization function is expressed in (1.17).

$$f(\mathbf{x}) = \max_{l} w_l f_l(\mathbf{x}) \qquad l = 1, \cdots, m \tag{1.17}$$

The graphical representation of the weighted objectives method employing the Tchebyshev scalarization function for determining a compromise design of a bi-objective optimization problem presenting a non-convex Pareto front is presented in Figure 1.6a.

As can be remarked from (1.17), the Tchebyshev scalarization function is not continuous, due to the presence of the *max* function in its expression. Therefore, this fact may pose certain difficulties to a gradient-based optimization algorithm. A solution to this problem might be the use of a heuristic optimization algorithm, such as genetic algorithm (GA).

Weighted objectives method with augmented Tchebyshev function

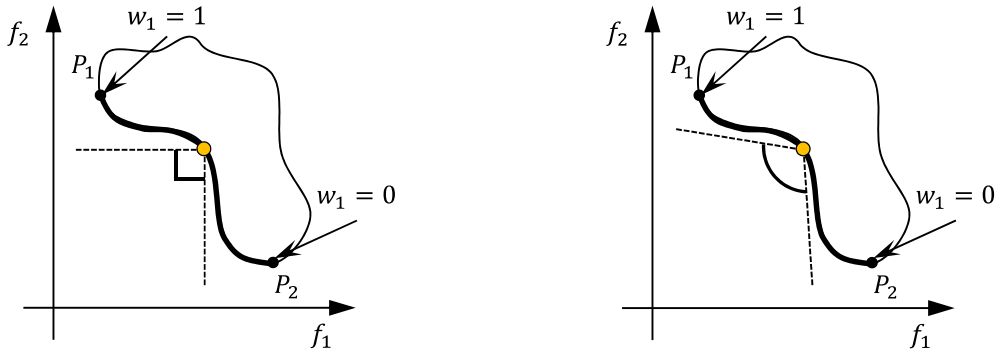
The Tchebyshev scalarization function presented in the previous paragraph only guarantees that the obtained solution is weakly Pareto optimal. For the Pareto optimality to be guaranteed, an improved weighting function has been proposed, the augmented Tchebyshev scalarization function, which represents in fact the same Tchebyshev formulation with an additional term. The expression of the augmented Tchebyshev scalarization is given in (1.18).

$$f(\mathbf{x}) = \max_{l} w_l f_l(\mathbf{x}) + \rho \sum_{l=1}^{m} w_l f_l(x) \qquad l = 1, \cdots, m \tag{1.18}$$

where ρ represents a parameter with a small positive value.

The ρ parameter allows controlling the angle determined by the two dotted lines which are presented in Figure 1.6. The non-dominated solutions situated on the convex areas of the Pareto front can be determined for any value of the ρ parameter. However, the non-dominated solutions situated on the non-convex parts of the Pareto front require using a small value for the ρ parameter, usually $\rho \in [10^{-6}, 10^{-2}]$.

The mechanism of the weighted objectives method using the augmented Tchebyshev scalarization function for finding a compromise solution on a non-convex Pareto front is presented graphically in Figure 1.6b.



a) Tchebyshev scalarization b) Augmented Tchebyshev scalarization

Figure 1.6 : Weighted objectives method with improved non-linear scalarization functions

The weighted objectives method employing the augmented Tchebyshev scalarization function will be further used by the optimization techniques proposed in the subsequent chapter of the manuscript.

Compared to the two previous scalarization functions, the augmented Tchebyshev function allows determining the non-dominated trade-off solutions of optimization problems presenting non-convex Pareto fronts, also guaranteeing the Pareto optimality of the solutions.

1.3.6.2 ϵ-constraint method

The ϵ-constraint method [BRI 07], [COL 02], [DIB 10] also known as the constraint transformation method [NAK 09], is a technique which transforms the initial multi-objective optimization problem into a single-objective constrained optimization problem so that it will be next handled using a classical single-objective optimizer.

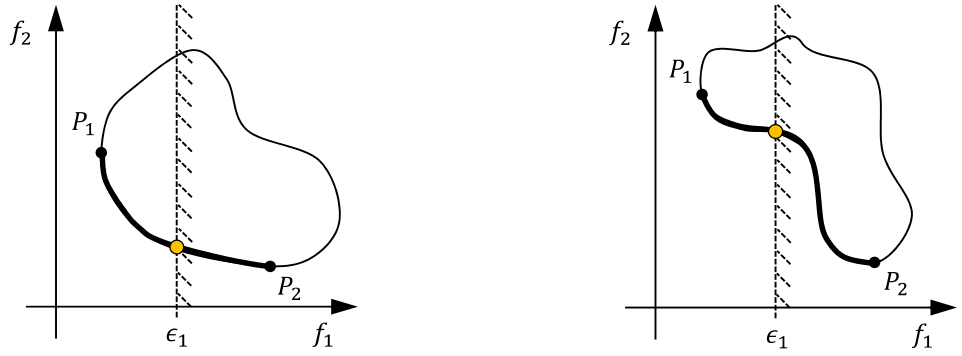
Considering the multi-objective problem formulation of (1.12), the ϵ-constraint method implies keeping one of the m objective functions (f_1 for example here) as objective, while passing all the other *m-1* objective functions in constraint. The additional constraint functions are imposed different limit values. The formulation of the obtained single-objective problem is given in (1.19).

$$
\begin{aligned}
&\underset{\mathbf{x}}{\text{Minimize}} && f_1(\mathbf{x}) && \\
&\text{subject to} && g_i(\mathbf{x}) \le 0 && i = 1, \cdots, n_i \\
& && h_j(\mathbf{x}) = 0 && j = 1, \cdots, n_e \\
& && g_{n_i+l}(\mathbf{x}) = f_l(\mathbf{x}) - \epsilon_l \le 0 && l = 2, \cdots, m \\
&\text{with} && \mathbf{x} = [x_1, \cdots, x_k, \cdots, x_n] && k = 1, \cdots, n \\
& && x_k^l \le x_k \le x_k^u &&
\end{aligned}
\tag{1.19}
$$

where ϵ_l represents the imposed limit for the l-th objective function which was transformed in constraint function.

In order to retrieve the Pareto front of the initial multi-objective problem using the ϵ-constraint method, a sequence of single-objective optimizations must be run, using several different limit values ϵ_l for the additional constraint functions.

Considering a bi-objective optimization problem, the application of the ϵ-constraint method for determining a design on the Pareto front is presented graphically in Figure 1.7. The objective function f_2 is kept as objective, while a limit ϵ_1 is imposed to f_1. The success of the ϵ-constraint method in determining a compromise design for the two figure-cases, when the Pareto front is convex and non-convex can be remarked from Figure 1.7a, respectively Figure 1.7b.



a) Convex Pareto front b) Non-convex Pareto front

Figure 1.7 : ϵ-constraint method application for determining a design on the Pareto front of a bi-objective optimization problem

As with the weighted objectives method previously presented, the advantage of this method consists in its simplicity of implementation. Nevertheless, compared to the previous weighted objectives method, the ϵ-constraint technique does not present difficulties when dealing with optimization problems presenting a non-convex Pareto front. Moreover, the total desired number of non-dominated designs belonging to the Pareto front is imposed by the designer. Different areas of the Pareto front, judged interesting by the designer, can be explored using a proper selection of the limits for the constraint functions.

An inconvenient of this method is represented by the fact that the distribution of the obtained trade-off solutions along the Pareto front might be strongly non-uniform, depending on the form of the front.

The ϵ-constraint technique was applied for solving different optimization applications which can be found later on in the following chapters of the manuscript.

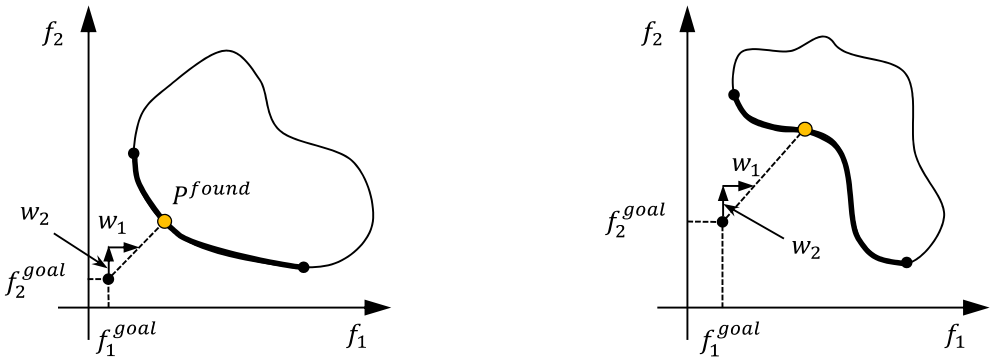
1.3.6.3 Goal-attainment method

The goal-attaining method [BRI 07], [COL 02], [DIB 10] is a technique which seeks to find the design belonging to the Pareto front which is closest to a point of given coordinates in the objective space, following a given direction. The mathematical formulation of the rephrased optimization problem using the goal-attainment method is given in (1.20).

$$
\begin{aligned}
&\underset{\mathbf{x},\xi}{\text{Minimize}} && \xi && \\
&\text{subject to} && g_i(\mathbf{x}) \le 0 && i = 1,\cdots,n_i \\
& && h_j(\mathbf{x}) = 0 && j = 1,\cdots,n_e \\
& && g_{n_i+l}(\mathbf{x}) = f_l(\mathbf{x}) - w_l\xi - f_l^{goal} \le 0 && l = 1,\cdots,m \\
&\text{with} && \mathbf{x} = [x_1,\cdots,x_k,\cdots,x_n] && k = 1,\cdots,n \\
& && x_k^l \le x_k \le x_k^u &&
\end{aligned}
\tag{1.20}
$$

where f_l^{goal} represents the goal value imposed by the designer for the l-th objective function, w_l is the weighting coefficient associated to the l-th objective function.

The goal-attainment method consists of minimizing a scalar value ξ while respecting the constraints of the initial optimization problem and m additional constraint functions depending on the goal and current objective function value, the weighting coefficients and the scalar quantity to be minimized. The mechanism of the goal-attainment method is graphically exemplified on a simple bi-objective optimization problem in Figure 1.8.



a) Convex Pareto front b) Non-convex Pareto front

Figure 1.8 : Goal-attainment method exemplification on a bi-objective case

The weighting vector $\mathbf{w} = [w_1, w_2]$ imposed by the designer gives the direction of search, relating the goal point $P^{goal}\left(f_1^{goal}, f_2^{goal}\right)$ to the Pareto front.

As can be observed from Figure 1.8, the goal-attainment method works for both convex and non-convex shapes of the Pareto front of an optimization problem.

The goal-attainment method can be employed when a goal is known or can be easily estimated, for example if the optimal design process starts from an existing design, which is sought to be improved by a given percentage.

1.3.6.4 Other transformation techniques

Several other transformation techniques allowing the resolution of a multi-objective optimization problem by a single-objective optimization technique exist in the literature. Two other techniques propose finding a design on the Pareto front by minimizing the maximum discrepancy to a goal design [BRI 07], [COL 02], also known as weighted min-max formulation [DIB 10], using the expression (1.21) or by minimizing a sum of squared differences as in (1.22).

$$\underset{\mathbf{x}}{\text{Minimize}} \quad f(\mathbf{x}) = \max_{l=1,\cdots,m} w_l \left| f_l(\mathbf{x}) - f_l^{goal} \right| \tag{1.21}$$

$$\underset{\mathbf{x}}{\text{Minimize}} \quad f(\mathbf{x}) = \sum_{l=1}^{m} w_l \left(f_l(\mathbf{x}) - f_l^{goal} \right)^2 \tag{1.22}$$

As with the goal-attainment transformation technique previously presented, the goal design can be considered the ideal point of the Pareto front [BRI 07] and the weighting coefficient vectors uniformly distributed are generated in the same manner.

1.3.6.5 Example test-problem

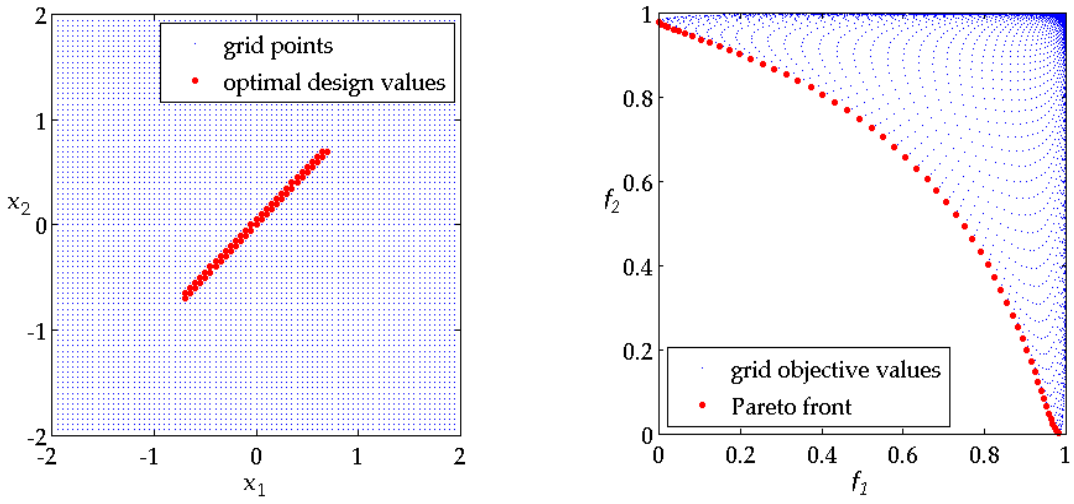
To exemplify the mechanisms of the different transformation techniques presented in the previous paragraphs, a simple analytical test-problem has been considered. The optimization problem considered is known as the VLMOP2 test-problem and was proposed by Veldhuizen et al. in [VEL 99]. The mathematical formulation of the bi-objective optimization problem VLMOP2 is expressed in (1.23).

$$\underset{\mathbf{x}}{\text{Minimize}} \begin{cases} f_1 = 1 - exp\left(-\sum_{i=1}^{n} \left(x_i - \frac{1}{\sqrt{n}} \right)^2 \right) \\ f_2 = 1 - exp\left(-\sum_{i=1}^{n} \left(x_i + \frac{1}{\sqrt{n}} \right)^2 \right) \end{cases} \tag{1.23}$$

$$\text{with} \quad \mathbf{x} = [x_1, \cdots, x_n] \qquad n = 2$$

$$x_i \in [-2,2] \qquad i \in \{1,2\}$$

The Pareto front of the problem is non-convex, fact which can be observed from the graphical representation of the front in Figure 1.9b obtained by the calculation of a grid of 80*80 designs over the design space, as in [KRE 08].

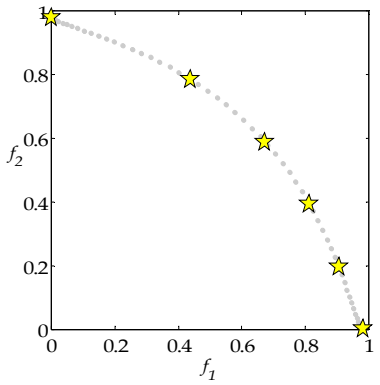


a) Design space b) Objective space

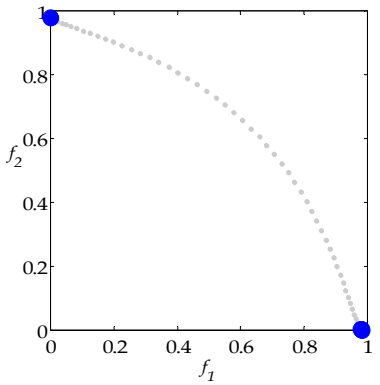
Figure 1.9 : Result of the 80*80 design grid evaluation for the analytical test-problem

The different previously presented transformation techniques have been employed for solving the bi-objective optimization problem expressed in (1.23). A number of 6 non-dominated solutions were imposed to all approaches.

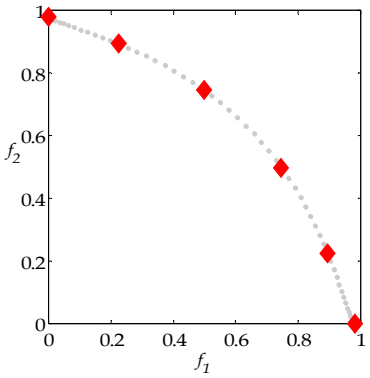
The ϵ-constraint technique was employed with success for solving this non-convex optimization problem. The Pareto front obtained is presented in Figure 1.10a. The approach for finding the Pareto front was decomposed into two steps. First, two single-objective unconstrained optimization problems were formulated in order to retrieve the two extreme points of the Pareto front, by minimizing the f_1, respectively the f_2 function. Second, for each of the 4 remaining non-dominated points, a different single-objective constrained optimization problem was formulated. The f_1 function was kept as unique objective function, while the f_2 function has been passed in constraint, as expressed in (1.19). A different limit value has been imposed to the f_2 function for each optimization, calculated based on its value for the two extreme points of the Pareto front. The Pareto front shape has been well retrieved by this transformation technique. Nevertheless, it can be also observed an irregular spacing of the solutions on the Pareto front, with a higher populated zone at the right-most part of the Pareto front in exchange for a less-denser zone at the upper-left area of the Pareto front. The ϵ-constraint technique is thus shown to be sensitive to the shape of the Pareto front. It is expected that in the extreme case of a problem presenting an "L" -shape Pareto front, the technique would fail dramatically in finding a uniformly-distributed Pareto front.



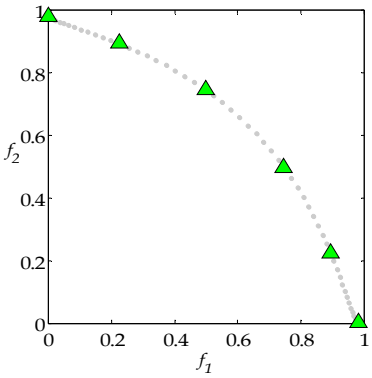
a) ϵ-constraint method



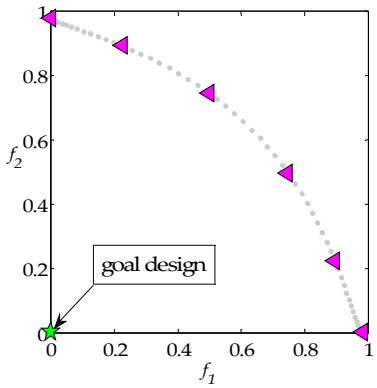
b) Linearly-weighted objectives method



c) Weighted objectives method with Tchebyshev scalarization function



d) Weighted objectives method with augmented Tchebyshev scalarization



e) Goal-attainment method

Figure 1.10 : Optimal results of the analytical test-problem optimization employing the presented transformation techniques

The weighted objectives transformation technique using the three scalarization functions presented previously in this chapter has also been employed for finding the Pareto front of the test-problem. Prior to the optimization runs, a set of 6 uniformly-spaced vectors of weighting coefficients Λ has been generated, obtaining the coefficients presented in (1.24).

$$\Lambda = \{[0,1], [0.2,0.8], [0.4,0.6], [0.6,0.4], [0.8,0.2], [1,0]\} \tag{1.24}$$

For each vector of Λ an optimization process has been run. As expected, the linearly-weighted scalarization function failed in retrieving the Pareto front of the problem, due to its inability of handling problems presenting a non-convex Pareto front. It is only the two extreme points of the Pareto front that have been obtained by this method, as can be seen from Figure 1.10b. The three optimization runs using the first three weighting vectors of Λ supplied the design corresponding to minimum f_2 value on the Pareto front, while the three other weighting vectors generated the other extreme point, corresponding to minimum f_1 value. No intermediary point on the non-convex Pareto front has been determined by this technique.

The optimal results of the weighted objectives transformation technique employing the Tchebyshev and the augmented Tchebyshev scalarization function are presented in Figure 1.10c, respectively Figure 1.10d. A uniform distribution of the points on the obtained Pareto front can be observed for the case of the two non-linear scalarization functions, Tchebyshev and augmented Tchebyshev function.

The goal-attainment transformation technique was equally employed for finding the Pareto front of the optimization test-problem. The optimal results of the goal-attainment technique are presented in Figure 1.10e. As in the case of the ϵ-constraint technique, the approach of finding the Pareto front of the problem implies two distinct steps; the first step being identical with the one from the ϵ-constraint technique. The two extreme points of the Pareto front allow defining a suitable goal design for the formulation of the subsequent optimization problems using (1.20). The goal design has been thus selected the design which presents the minimal values of each objective function of the extreme designs, $\mathbf{f}^{goal} = \left[f_1^{goal}, f_2^{goal}\right] = \left[f_1^{min}, f_2^{min}\right]$. The optimal results obtained are similar with those obtained using the weighted objectives technique employing the two non-linear scalarization functions.

Among the presented transformation techniques, the weighted objectives method employing the two non-linear scalarization functions and the goal-attainment method provided similar results for the analytical test-problem considered. The uniformity of the points distribution on the Pareto front employing the ϵ-constraint technique was shown to be strongly related to the shape of the Pareto front, with a less fortunate performance in the case of optimization problems presenting Pareto fronts showing an "L"-shape Pareto front. The weighted objectives transformation technique employing linear weighting coefficients was found to not function in the case of problems presenting a non-convex Pareto front. The choice of one technique over another depends on the context of the study, the available data concerning the optimization problem, and it is the designer's call to judge a method more appropriate than the others.

1.3.7 Complex systems specific optimization strategies

The complexity of these systems is given by:

- The expensive computation cost of heavy simulation models;
- The large number of components of the system, thus many design variables and specific constraints.

Each of these features is addressed through different methods. Hence, for the optimization problems which use costly simulation models, it is appropriate to employ metamodel-based optimization strategies for reducing the computational cost of the optimization. When the system to be designed is very large, being composed of several components and having a large number of design variables and constraint functions, it is appropriate to employ optimization strategies based on the decomposition of the system, distributing thus the computation across the system.

1.3.7.1 Metamodel-based optimization strategies

In electromagnetics and other domains of the engineering, the devices or products to be optimally designed are commonly represented through complex analysis codes (such as: finite element method (FEM), boundary element method (BEM), computational fluid dynamics (CFD), etc.), which apply the basic laws of the concerned domain on a discretized grid of small elements representing the device to be analyzed. The main problem encountered when using such complex analysis codes consists in the computationally expensiveness of such a model. Also, due to the discretization employed, these models present discontinuities over the considered domain and for some given combinations of design variables, the analysis of such codes might fail in providing valid responses. All these facts make the integration of models represented using such analysis tools within a classical optimal design process, limited or even impossible.

A common practice in the optimal design process consists in creating response surfaces or metamodels of these expensive simulation models. They are fast, continuous and offer responses for any design configuration considered. In exchange, the accuracy of such representations might be very poor, therefore offering erroneous information about the modeled device. The problematic of the integration of metamodels within the optimal design process makes the subject of Chapter 2 of this manuscript. Several means of integration of metamodels within the optimal design process are discussed and a complex multi-objective efficient optimization approach is developed.

1.3.7.2 Decomposition-based optimization strategies

A complex system is seen here as a collection of models, disciplines or components, strongly interacting and which must be considered together within a design process. The complex system is too large to be addressed as a whole, thus requiring being decomposed.

A classification of these strategies can be done based on the number of optimization algorithms employed for the optimization task. Hence, these optimization approaches fall into two categories: single-level methods and multi-level methods. The single-level methods imply a single optimization algorithm. The multi-level methods employ a separate optimization algorithm for solving each of the optimization problems of the components of the decomposed structure.

Single-level design optimization strategies

The single-level design optimization strategies refer to the classical multidisciplinary design optimization (MDO) formulations. The multi-disciplinary feasible (MDF), individual disciplinary feasible (IDF) and all-at-once (AAO) formulations, introduced by Cramer et al. [CRA 94], are intended to address multi-disciplinary optimization problems with models having a relatively reduced size. If the number of design variables or constraint functions is important, these

approaches are shown to be unreliable [ALL 08]. The single-level methods employ a single optimizer for solving the optimization problem, thus the inter-disciplinary consistency is implicitly guaranteed. These optimization approaches have been intensively studied. Recent works of Kreuawan [KRE 08] and Ben-Ayed [BEN 12b] at L2EP laboratory from Ecole Centrale de Lille address such approaches. Thus, these approaches will not be addressed here. Instead, the younger multi-level approaches make the subject of the work presented in Chapter 3 of this manuscript.

Multi-level design optimization strategies

The multi-level design optimization strategies make use of several optimizers to address the optimization of each element of the decomposed structure, thus implying the distribution of the computational burden across the decomposed structure. The different optimizers called to address the multi-level problem communicate under a specific coordinated strategy. Depending on the object of the decomposition, several categories of optimization strategies exist. Many multi-level strategies have been developed during the last two decades, such as Collaborative Optimization (CO), Analytical Target Cascading (ATC), Collaborative Sub-Space Optimization (CSSO), Bi-Level Integrated System Synthesis (BLISS), to cite only a few. The multi-level design optimization methodology makes the subject of Chapter 3 of this manuscript. Several electromagnetic applications are addressed using some of these strategies.

1.4 Pareto front quality assessment tools

Frequently, it shows the need to estimate the quality of the obtained Pareto front or to compare two or more Pareto fronts. Most of the multi-objective formulations of practical optimization problems imply a number of two or maximum three objective functions. In this case, the assessment of the Pareto front's quality or the comparison between two or more Pareto fronts can be done visually, based on the graphical 2D, respectively 3D representation of the objective function values of the optimal results. This represents a qualitative estimation or comparison of one or more Pareto fronts.

However, the visual estimation of the Pareto front quality might not suffice. In this case, a quantitative estimation of the Pareto front imposes. In comparison with the single-objective case, where the optimal result is a single value, in the multi-objective case, the optimal result is a multi-dimensional set of designs. Usually, two aspects must be addressed in the assessment of Pareto front's quality:

- Convergence of the Pareto front to the true Pareto front;
- Diversity of the solutions in the obtained Pareto front.

A number of different metrics have been proposed in the literature over the years for assessing the quality of the Pareto front, by addressing the above-mentioned aspects. Several metrics have been introduced here for addressing the above-mentioned aspects of a Pareto front.

1.4.1 Metrics definition

In order to assess the quality of a Pareto front or to objectively compare two Pareto fronts, a number of metrics have been developed and exist in the literature. Three complementary metrics for assessing the quality of a Pareto front are introduced here.

Generational distance (*GD*)

The generational distance (*GD*) measure introduced by Veldhuizen and Lamont in [VEL 00] is a measure for the closeness of the obtained front to the real Pareto front. The mathematical formulation of this metric is expressed in (1.25).

$$GD = \frac{1}{n_{found}} \sqrt{\sum_{i=1}^{n_{found}} d_i^2} \tag{1.25}$$

where n_{found} is the cardinality of the obtained Pareto front $\wp_{found}$ and d_i is the distance in the objective space between the *i*-th point of $\wp_{found}$ and the closest point on the true Pareto front $\wp_{true}$.

The closer the value of *GD* is to zero, the closer are the obtained non-dominated solutions to the true Pareto front. When the cardinality of $\wp_{found}$ is reduced, a low value for *GD* might be obtained and therefore it is suggested the use of a complementary metric, named reverse generational distance (*RGD*). The expression of this later metric is given in (1.26).

$$RGD = \frac{1}{n_{true}} \sqrt{\sum_{j=1}^{n_{true}} \tilde{d}_j^2} \tag{1.26}$$

where n_{true} is the cardinality of the true Pareto front $\wp_{true}$ and $\tilde{d}_j$ represents the distance between the *j*-th point of $\wp_{true}$ and the closest point on $\wp_{found}$.

Spacing (*S*)

The Spacing metric (*S*) [VEL 00] quantifies how uniformly the non-dominated solutions are spread out on the obtained Pareto front. The mathematical formulation of this metric is given in (1.27).

$$S = \sqrt{\frac{1}{n_{found} - 1} \sum_{i=1}^{n_{found}} \left(\bar{d} - d_i\right)^2} \tag{1.27}$$

where d_i represents a distance calculated for each *i*-th design of $\wp_{found}$ according to (1.28) and $\bar{d}$ represents the mean value of all d_i.

$$d_i = \min_j \sum_{k=1}^{m} \left| f_k^i - f_k^j \right| \tag{1.28}$$

where f_k^i and f_k^j represent the values of the k-th objective function of i-th, respectively j-th design of $\wp_{found}$.

A value of $S = 0$ signifies that the designs of $\wp_{found}$ are equally-spaced, the $\wp_{found}$ presenting a uniform distribution of points.

Error ratio (*ER*)

The error ratio (*ER*) is a measure introduced by Van Veldhuizen in [VEL 99] which accounts for the quantity of non-dominated designs of the obtained Pareto front $\wp_{found}$ which do not belong to the true Pareto front $\wp_{true}$. The mathematical formulation of the *ER* metric is expressed in (1.29).

$$ER = \frac{\sum_{i=1}^{n_{found}} e_i}{n_{found}} \tag{1.29}$$

where $e_i = 1$ if an obtained design is not on $\wp_{true}$ and $e_i = 0$ otherwise.

An obtained design is considered to belong to $\wp_{true}$ if it is within a tolerance (generally 1-5%) in the objective space from a design of $\wp_{true}$.

Obviously, a value of $ER = 1$ signifies that none of the designs of $\wp_{found}$ belongs to $\wp_{true}$, while $ER = 0$ signifies that $\wp_{found}$ is included in $\wp_{true}$.

The metrics presented here are intended to quantify the performance of a multi-objective optimization problem. Other metrics, such as the hypervolume estimation [ZIT 99] are equally available to designers. However, the graphical representation of the obtained sets of data is very helpful, compulsory even, for the designer, guiding him in the decision-making process. Some common multi-dimensional data representation techniques are next reviewed.

1.4.2 Multi-dimensional data representation

A number of graphical tools for the representation of multi-dimensional data exist in the literature and are meant to assist the designer in the decision-making process [RAH 99]. Graphical representations such as bar charts, spider diagrams, bubble plots, scatter plot matrix, parallel coordinates representation, etc. offer the designer a very good vision over the multi-dimensional results data. A few of these representation techniques, commonly used by designers, are next introduced.

Box-plot

The box-plot diagram, also known as box-and-whisker diagram or plot, is a current representation tool of series of data in statistics[2]. Its great popularity resides in the number of different information that can be represented condensed in one figure. A box-plot diagram regularly presents five different details about a set of data: the minimum value of the samples, the maximum value of the samples, the lower quartile (i.e. standard deviation at 25%), the maximum quartile (i.e. standard deviation at 75%) and the median quartile (i.e. mean value of the samples 50%).

[2] Details about the box-and-whisker diagram can be found online at: http://en.wikipedia.org/wiki/Box_plot.

Scatter plot matrix

The scatter plot matrix[3], also known as scatter chart or scatter graph[4] represents a bi-dimensional graphical representation of data. In this graph, the designs from a set of data are represented through two of their coordinates, as a cloud of points, with each coordinate of the designs along one dimension of the Cartesian space. Eventual correlations between design variables can be observed graphically. For data with a dimensionality superior to two, all combinations of two design variables scatter representation can be displayed under the form of a matrix of scatter plots, where a scatter plot of the matrix displays the two dimensions of the designs associated to the row and the column of the matrix.

Bubble plot

The bubble plot, also known as bubble chart[5] is used to represent three or more coordinates (objective function, constraint function or design variable) of a set of designs. Usually, in optimization it is used to represent the data in the associated multi-dimensional objective space. For the standard version of the bubble plot, the designs are represented as different size bubbles in a bi-dimensional Cartesian space. Each coordinate of the Cartesian space is associated one design coordinate (objective function) and the size of the bubbles is associated to a third coordinate. Supplementary information can be represented on the same graph by associating the color of the bubbles, their transparence, their shape, etc. to a different design coordinate. The feasibility of designs can be also associated to one mean of representation (color, transparency or shape of the bubbles), thus being easily recognizable.

Parallel coordinates representation

The parallel coordinates chart [WEG 90] represents a graphical tool for representing multi-dimensional data sets[6]. In this representation, parallel lines are used to represent the domain of variation along each dimension of the considered data. The parallel lines can be either displaced vertically or horizontally. This graph allows the representation of both the design variables values and those of the objectives and constraints. Each design is represented by a multi-line across the parallel lines; the intersection of the multi-line with the parallel lines giving the coordinates of the design along the different dimensions. This kind of representation is very useful for identifying the feasible sub-domain of the design space of a problem, the sub-domain containing the optimal solutions or the dependence of the objective and constraint functions on the design variables.

1.5 Commercial optimization software

A number of commercial general-purpose optimization software products are available on the market. These commercial optimization software products, dedicated to the optimal design of

[3] A Matlab® implementation of the scatter plot matrix is available at: http://www.mathworks.fr/help/techdoc/ref/plotmatrix.html.
[4] More information about the scatter plot matrix can be found at: http://en.wikipedia.org/wiki/Scatter_plot.
[5] Online information about the bubble chart is available at: http://en.wikipedia.org/wiki/Bubble_chart.
[6] More information about the parallel coordinates is available online: http://en.wikipedia.org/wiki/Parallel_coordinates.

devices, systems and processes, offer a large palette of tools for both the preliminary analysis and the post-optimization results visualization and decision making of the optimization problem, along with a number of single- and multi-objective optimization algorithms. To respond to the industrial needs, the producers of most of these software products have developed integrated bridges to the most notorious CAD and CAE software products. This way, simulation models developed using such software products can be directly integrated within the optimization process.

In order to benefit from the large selection of analysis and decision-making tools offered by such software, two commercial optimization software products have been analyzed modeFRONTIER® [EST 12], a product of Esteco Company and Optimus® [NOE 12], a product of Noesis Solutions. Both products offer the possibility of integrating externally-developed optimization algorithms within their structure. The integration of the optimization strategies developed and presented further on in this manuscript has been analyzed for the two optimization software products. An example of integration of the MEGO algorithm developed in this work, within the modeFRONTIER® product is presented in paragraph 2.5.9 of this manuscript.

1.6 Conclusion

The optimal design process of a device or system represents a three-phase process. The core element of this process is represented obviously by the optimization algorithm runs. Nevertheless, two equally important phases, a pre-processing phase and respectively a post-processing or results analysis and interpretation phase, must be correctly addressed within the process of optimal design. In this chapter, the elementary notions implied by the single- and multi-objective optimization problem formulations have been reviewed, and the notion of global optimality and Pareto optimality have been introduced for the single-objective, respectively the multi-objective optimization context. A number of different classical optimization approaches, deterministic and metaheuristic have been cited, along with some ways of addressing the optimal design of complex systems, which makes the subject of this work and which will be addressed in the following chapters. Sometimes, it might be useful to express a multi-objective optimization problem using a single-objective formulation. For this purpose, several different transformation techniques have been reviewed and their performance has been analyzed with regard to the convexity of the underlying Pareto front. For the comparison and results validation of a multi-objective optimization, different metrics have been introduced. Different tools for the preliminary model analysis and validation phase of the optimal design and the post-processing of an optimization run, having as goal to assist the designer with the decision-making task have been reviewed. Once all these elements introduced, the focus is set in the next chapter on the optimization approaches based on metamodels of the devices to be optimally conceived.

Chapter 2 Metamodel-based Design Optimization (MBDO)

In this chapter, the attention is focused on the integration of metamodels within an optimization process. The optimal design process based on the use of metamodels is called "metamodel-based design optimization" or "surrogate-assisted design optimization", depending on the different schools that worked in parallel at the development of this optimization approach. However, the two denominations address one and the same thing. The idea behind the optimization based on metamodels consists in reducing the computational burden of heavy simulations, by using fast-evaluation metamodels. This chapter starts by presenting the metamodel-based optimization process, identifying its advantages along with its drawbacks. The different types of metamodel-based optimization strategies are then presented and each of them is studied in detail, highlighting the different advances in the field, developed in this work. The application of these approaches to the optimal design of electromagnetic devices is addressed towards the end of the chapter.

2.1 Why optimizing using a metamodel?

A natural question that arises is: "Why using metamodels within an optimization process?"

In order to accurately model the numerous physical phenomena fostered by electromagnetic devices, heavy analysis codes are often used to simulate their behavior. Despite the advances in computing power, over the last decade especially, the expense of running analysis codes remains non-neglecting. Hence, single evaluations of finite element analysis codes can take for instance from a few minutes up to hours, even days, following the desired type of simulation (e.g. dynamic analysis, transient analysis etc.). The main purpose of the use of metamodels within an optimization process consists therefore in the significant overall time reduction of the optimization process, by avoiding heavy simulations with long computational time.

Numerical models are confronted to numerical noise (e.g. for the FEM, the source of numerical noise consists of the mesh adaptation and the FE discretization) [NEI 96], [MES 07], which affects or alters the convergence of the optimization algorithm, especially in the case of gradient-based algorithms. However, metamodels, mostly interpolating models, are noise-free and hence they are not confronted to such problems.

Complex numerical models are often non-robust, the analysis and even the mesh generation of such models failing for different design configurations. Usually, the development of such a complex numerical model is realized starting from an existing electromagnetic device, in order to simulate

its behavior. The simulation model can thus be adjusted and improved by aligning the simulation results to experimental measurements of the existing device. The numerical model will therefore provide accurate results for design configurations close to the initial device configuration that served at the construction of the numerical model; for important geometric and/or physical changes in the design, the model is liable for failing in providing with an accurate prediction, or, even worse, any prediction at all (failing the high fidelity simulation). When this event occurs, the design engineer has to intervene, correct the problem by adjusting diverse parameters and re-launch the simulation. The resources (e.g. time, money, experience, etc.) required by the development of a completely parameterized, fault-proof numerical model can be prohibitive. In exchange, metamodels do not need any parameter adjustment and, once again, they offer the robustness which numerical simulations are not able to provide.

In the industrial practice, it often occurs that the person which develops a model of a device or a system is not the same with the one that performs the optimal design task. Also, simulation models may be exchanged between engineers from different teams inside the same company, and which might be used for various purposes and necessitating different levels of accuracy. Thus, all the compatibility issues generated by the use of different platforms, different hardware configurations, different versions of the software, etc. might represent a real problem and result in an important time loss in the product development. Due to their reduced size and portability property, the metamodels might present themselves as a solution to this kind of problems.

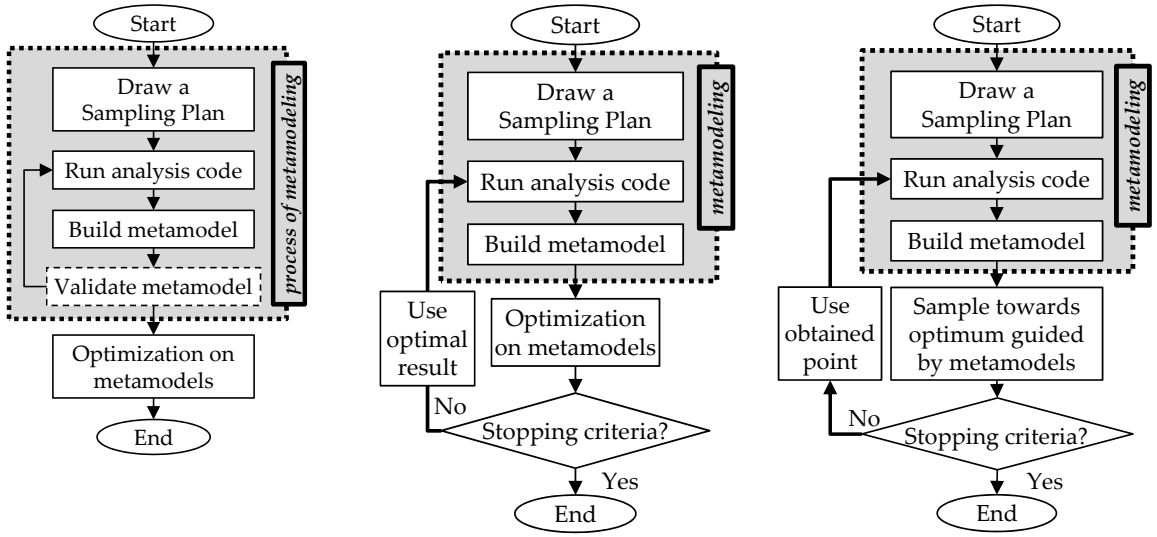
Another aspect is related to the intellectual property protection issues. For example, in the case of partnerships, models need to be exchanged between companies, or between the company and diverse service provider companies or consultancy offices. Also, the commercial or in-house CAD/CAE or other simulation software is often protected by means of static or network licenses, dongles or other means of preventing illicit use of the software. Hence, the cooperation process, ensuring the protection of the intellectual property contained by the developed models often faces a great challenge. An answer to this problem of cooperation without disclosing confidential information might come from the use of metamodels. From this point of view, the metamodel behaves as a "black-box", simulating the functional relationship between the inputs and outputs of the fine model, and without supplying any insight into the nature of the underlying relationships.

Metamodels, due to their fast evaluation characteristic, allow a thorough exploration of the design space, thus gaining "computationally cheap" insights into the functional relationship between the input and the output parameters of the high fidelity model (computationally expensive analysis code), by means of different statistical methods, such as parameter influence analysis, ANOVA, non-influential factors detection, which can be excluded from the optimization problem formulation by transforming them into fixed parameters, exploration of the design space, etc.

However, the use of metamodels within an optimization process presents also some drawbacks that must be stated. No matter how many design configurations served at constructing a metamodel, the global prediction accuracy of the metamodel will always be inferior to the accuracy of the model that served as reference; the metamodel cannot replace entirely the real model. The metamodel is less accurate than the real model, which might sometimes be deceiving from the optimization point of view and could misguide the optimization process. To overcome this drawback, several solutions have been proposed over the last decade, which will be overviewed later on in this chapter.

2.2 Metamodel-Based Design Optimization strategies

The optimization process integrating metamodels of the expensive simulation code can be found in the literature under different denominations, depending on the domain of science that is employed, the different schools, or it can be just a question of preference. Thus, to cite only a few members of the scientific community addressing the subject from different domains of the science, in Canada, Wang et al., in the mechanical engineering branch at the University of Manitoba prefer the term of "metamodel-based design optimization" [WAN 07], at the University of Southampton, Forrester, Sóbester and Keane, who deal with aerospace applications, prefer the notion of "surrogate model-based optimization" [FOR 08], Knowles et al. from the University of Manchester, UK prefer the notion of "Optimization on a given-budget of evaluations" in the computer science domain [KNO 05], Couckuyt, Crevecoeur, Gorissen et al., at the University of Gent, in Belgium use both the notion of "surrogate-based infill optimization" and "metamodel-based optimization" for their applications in the electromagnetic domain [COU 10], Villemonteix et al., at Université Paris-Sud XI, France prefer the notion of "Optimization of expensive-to-evaluate functions" for their diverse industrial applications [VIL 09], Holmström, Quttineh et al. from the Mälardalen University in Sweden use the notion of "expensive black-box optimization" within a mathematical framework [HOL 08], the notion of "Kriging-based optimization" is preferred by Ginsbourger, Le Riche, and Carraro at the Ecole Nationale Supérieure des Mines of Saint-Etienne in an applied mathematics context [GIN 10], Hemker at Technischen Universität Darmstadt in Germany speaks of "Surrogate Optimization" when addressing electrical engineering applications [HEM 08]. In the international literature, there is a great number of works addressing the integration of metamodels within the optimization process, but all these can be regrouped in the following main categories presented in Figure 2.1. In [WAN 07], an additional method is suggested, a direct sampling approach which uses metamodels only to guide an adaptive sampling, but it does not use a formal optimization process, therefore it is not considered in this study.



a) sequential approach b) adaptive approach c) criterion-based approach

Figure 2.1 : Different MBDO strategies

The first approach presented in Figure 2.1a is a sequential approach, straight-forward or one-shot approach, and does not involve any model reworking in the optimization process. A global metamodel is first built using a metamodeling strategy, in order to accurately represent the expensive simulation code. Next, a validation step might or might not be present in the metamodeling process. Once the metamodel is created, it is then used to substitute the expensive analysis code within a single- or multi-objective optimization loop using a classical optimization algorithm, either a trust-region method (gradient-based optimizer) or an evolutionary algorithm (e.g. genetic algorithm) to find the optimal design(-s). The main advantage of this approach consists in the simplicity of its implementation. The fine model is simply replaced by the objective and constraint functions metamodels within the single- or multi-objective optimization process. Also, the metamodel is independent from the optimization process, thus it can be easily exchanged. However, when the metamodels have a poor accuracy, the optimization process might fail in finding the optimal solutions of the problem. This is more problematic in the case of constrained optimization problems, where the constraint functions metamodels lack of accuracy might result in finding a suboptimal design, or worse, finding an unfeasible design.

In the second approach, synthesized in Figure 2.1b, an iterative process of metamodel adaptation is fostered within the metamodel-based optimization process. At each iteration of this metamodel-based optimization strategy, the metamodels are used within an internal single- or multi-objective optimization loop in order to search for optimal design(-s). Once a design or a set of designs found, the expensive simulation code is run and the obtained values are used for reconstructing the metamodels. The new metamodels integrate thus more information and result in an increased accuracy. In order to ensure the metamodels increase of accuracy throughout iterations, an additional global accuracy improvement phase can be present in the general optimization process. Hence, the global metamodels prediction becomes more and more accurate with the iterations, increasing the accuracy of the final optimal solution(-s). Apart from its relatively moderated implementation complexity, this approach has the advantage of naturally coping with the distributed computation, which could reduce substantially the overall optimization time. The desired number of fine model evaluations at each iteration of the MBDO process is specified by the designer. Nevertheless, a major drawback of this approach is represented by the high dependence of its success on the global accuracy of the metamodels. When many design variables are involved and/or the fine model is highly non-linear, obtaining high global accuracy metamodels is not possible. Therefore, the metamodels lack of accuracy might prevent the optimization process to converge to the optimal solution(-s) of the problem.

In many cases, the functional relationships that govern the functioning of the device to be optimized are highly non-linear, in which case globally-accurate metamodels are difficult or impossible to obtain, demanding a prohibitive amount of computation. For this type of problems, an optimal solution might be required with a reduced number of simulation model runs. Thus, the third approach considered and presented schematically in Figure 2.1c proposes to address this type of problems, by combining the search for optimal solutions with the phase of metamodels exploration, throughout the expression of an infill criterion or merit function. Compared to the previous approach, this strategy does not seek to attain a high global accuracy of the metamodels. Instead, the metamodels will present high accuracy locally, in the regions where the optimal solution(-s) lies. This approach is the most complex among the three approaches presented, but also the most efficient. Using a complex infill criterion which combines the progressive global

improvement of metamodels along the optimization process with the search of optimal solutions, the exploration of the design space is ensured while obtaining improved solutions, and with a reduced number of fine model (e.g. simulation model) evaluations. An important drawback of this approach is represented by the sequential character of its infill criterion, i.e. at each iteration of the algorithm only one infill design is selected for evaluation using the fine model. Thus, at each iteration the optimization process necessitates a valid output from the fine model. This is an important inconvenient because simulation models are never completely robust; when a design analysis fails, the automatic process is suspended and the design engineer needs to intervene to adjust the simulation models parameters and resume the optimization process. Moreover, due to its sequential character, the fine model evaluation cannot benefit from the advantages of the distributed computation. To overcome this aspect, a couple of solutions are proposed in order to adapt this MBDO approach to the distributed computation, therefore allowing for an important overall optimization time-saving. This approach is well-suited for optimization problems with a moderate number of design variables, i.e. less than 10-15 variables.

Generally, the metamodel-based optimization algorithms present two main features:

(i) Search the design space for the optimal solution(-s) using the metamodels, phase which is named "*metamodel exploitation*", or simply "*exploitation*";
(ii) Search the design space for promising areas and improve the metamodels global accuracy, phase which is called "*design space exploration*", or just "*exploration*";

These two features can be found at the basis of all metamodel-based optimization algorithms, either as two independent mechanisms, as in the case of the second MBDO approach presented above, or aggregated within a unique mechanism, as for the infill criterion of the third MBDO approach previously presented.

In the following paragraphs, the three-mentioned metamodel-based optimization approaches will be presented in detail and their advantages and drawbacks will be analyzed using both abstract analytical test-problems and physical applications addressing the optimal design of electromagnetic devices. A special attention will be dedicated to the latter one, which mainly represents the core of this chapter, containing most of the author's original contribution to the domain and to which a dedicated optimization tool was developed.

2.3 Sequential metamodel-based optimization

This approach represents the most classic of the optimization strategies employing metamodels, and also the most basic. The idea behind this strategy is to create a "cheap"-evaluation copy of the expensive simulation model, by building a metamodel for each objective and constraint function of the latter. The metamodel has the advantage of a fast evaluation, in exchange for the loss of accuracy, allowing it to be integrated into an optimization process. The sequential metamodel-based optimization technique can be described using the following steps:

- Step 1: Select an experimental design and run simulation model to compute the outputs;
- Step 2: Build a metamodel for each objective and constraint function of the optimization problem using the points of the experimental design;

- Step 3: (*optional*) Validate the metamodels created at Step 2. If metamodels are not accurate enough, then go to Step 1; otherwise continue with Step 4;
- Step 4: Use an optimization algorithm (e.g. genetic algorithm, SQP) to search for the non-dominated trade-off front using the metamodels built at Step 2;
- Step 5: Run the simulation model to calculate its outputs for the selected design.

The workflow of the sequential metamodel-based optimization technique is given in Figure 2.2.

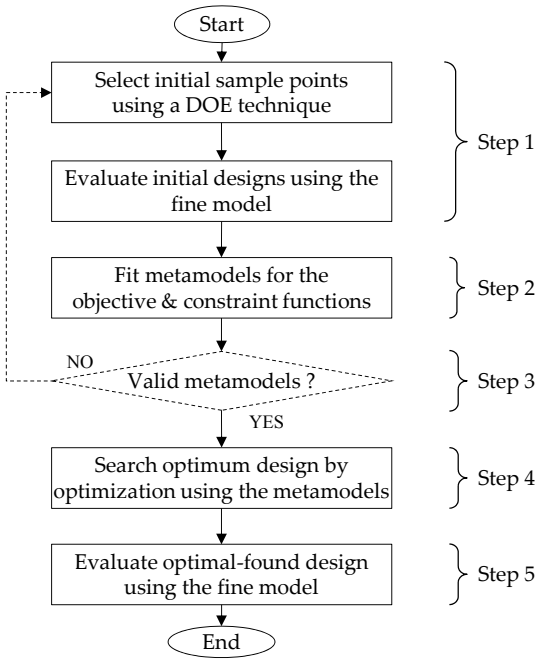


Figure 2.2 : Workflow of the sequential metamodel-based optimization

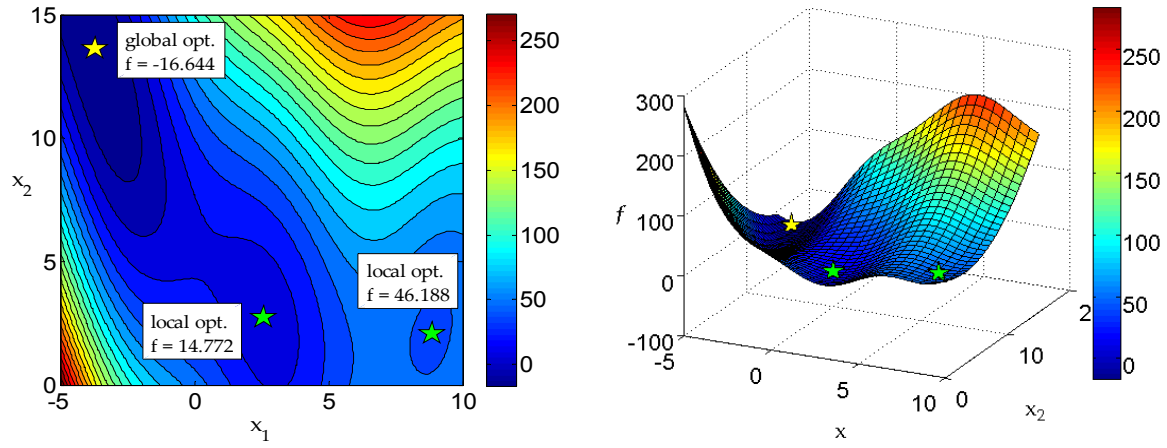
Once an experimental design is selected, the simulation model is called to compute the output parameters. A global metamodel (Response Surface (RS), Radial Basis Function (RBF), Kriging etc.) is then fit over the initial samples. An optimization loop is then launched with a classical optimization algorithm using not the "computationally expensive" simulation model, but instead, the "computationally cheap" metamodel. Hopefully, if the problem is not multi-modal and the metamodel gets to capture and accurately predict the output parameters of the analysis code that it replaces, the optimum found through this optimization will also be the global optimum of the simulation model. The mathematical expression of the optimization problem becomes thus:

$$\left.\begin{matrix}\min\limits_{\mathbf{x}} F(\mathbf{x}) \\ G(\mathbf{x}) \leq 0 \\ H(\mathbf{x}) = 0\end{matrix}\right\} \mapsto \begin{cases}\min\limits_{\mathbf{x}} \hat{F}(\mathbf{x}) \\ \hat{G}(\mathbf{x}) \leq 0 \\ \hat{H}(\mathbf{x}) = 0\end{cases} \tag{2.1}$$

To exemplify this purpose, the single-objective two-variable optimization test function, the modified Branin-Hoo function [PAR 10] is considered here.

$$\underset{x_1,x_2}{\text{Minimize}} \quad f(x_1,x_2) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left[\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 1\right] + 5x_1 \tag{2.2}$$

where $x_1 \in [-5,10]$ and $x_2 \in [0,15]$.



a) Contour representation b) 3D surface representation

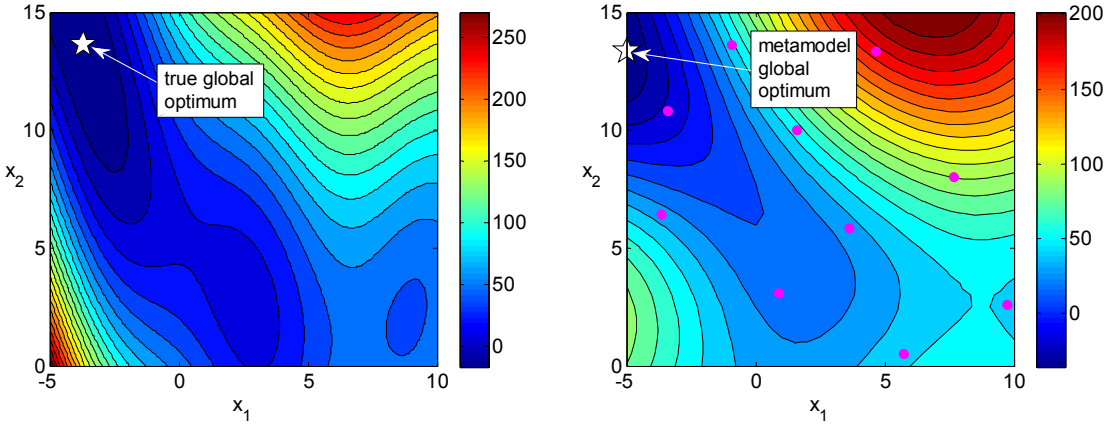
Figure 2.3 : The modified Branin-Hoo function

The modified version of the Branin-Hoo function has only one global optimum, represented by the yellow star in Figure 2.3 and two local optima, represented by the green stars in Figure 2.3 and presented with their numeric values in Table 2.1.

Table 2.1 : Global/local optima of the modified Branin-Hoo function

x_1	x_2	f	Optimum type
-3.695	13.635	-16.644	global
2.59	2.745	14.772	local
8.875	2.055	46.188	local

To launch the sequential metamodel-based optimization with the modified Branin-Hoo function, we consider a RBF metamodel as surrogate for the true function (i.e. fine model) in (2.2), and we choose a space filling Latin Hypercube sampling design (LHS) of 10 designs to sample the design space. Figure 2.4a presents the contour of the true function and the RBF metamodel is presented in Figure 2.4b. The dots in Figure 2.4b show the support points.



a) True function (2.2) b) 10 LHS support points for the RBF model

Figure 2.4 : Model comparison for the Branin-Hoo modified function

The RBF metamodel captures the global trend of the true function, as can be seen in Figure 2.4.

A gradient-based optimization algorithm (sequential quadratic programming – SQP in this case) is used to search for the global minimum of the previously created metamodel. The two optima are represented in Figure 2.4 through the white asterisks. Table 2.2 presents the optimum found using the metamodel, in comparison to the global optimum of the true function.

Table 2.2 : True function/metamodel global optima comparison

Model type	x_1	x_2	f	$\hat{f}$
true function	-3.695	13.635	-16.644	-15.371
metamodel	-5	13.364	2.342	-35.714

The results obtained for the modified Branin-Hoo function are deceiving. Although the metamodel succeeded in capturing the global trend of the modeled function, its accuracy is not good enough in order to predict the outputs of the true function with precision. The metamodel cannot substitute the real model of the function.

The main drawback of this "one-shot approach" is thus represented by the inaccuracy in the metamodels prediction of the true function. Moreover, in the case of constrained optimization problems, the lack of accuracy of the constraint functions metamodels might misguide the optimization algorithm, either by selecting a design which is suboptimal, or worse, a design which fails to respect the fine model constraint, therefore unfeasible. A solution which might overcome this drawback is represented by the use of an adaptive metamodeling strategy. The global quality of a metamodel is related to the size of its list of support points, hence adding more points to this list will improve the global accuracy of its prediction. Once we have seen that the success of the "one-shot approach" is highly dependent of the global accuracy of the objective and constraint functions metamodels, thus possibly deceiving, we will next analyze the integration of an adaptive-metamodeling within an optimization process.

2.4 Adaptive metamodel-based optimization

The adaptive metamodel-based optimization consists in performing a sequence of multiple optimization processes using a metamodel of the device to be optimized which is systematically improved from one optimization process to the following one. The optimal design point resulting from an optimization process is introduced into the list of support points that serves for constructing new metamodels for the device to be optimized. By considering the previously obtained optimal design for constructing new metamodels generally improves the prediction accuracy of the new metamodels. Thus, through this systematic process, the metamodels prediction is improved, increasing the chances of finding the global optimal design. The general process of this technique as well as single- and multi-criteria particularities and implementation issues are discussed in the following paragraphs.

2.4.1 General process

The general process of this adaptive optimization process with systematic metamodel improvement is considered. The adaptive metamodel-based optimization process consists of the following steps:

- Step 1: Select an experimental design and run simulation model to calculate the outputs;
- Step 2: Build a metamodel for the objective and each constraint function of the optimization problem using all simulated designs;
- Step 3: Use an optimization algorithm (e.g. genetic algorithm, SQP with multi-start[7]) to search for the global optimum using the metamodels built at Step 2;
- Step 4: Run the simulation model in order to validate the selected design;
- Step 5: Add the design obtained at Step 4 to the list of evaluated designs ($\mathcal{L}$);
- Step 6: Verify stopping criteria (e.g. prescribed number of simulation model evaluations, total elapsed time, prescribed metamodels accuracy, etc.). If stopping criteria are met, then stop the algorithm; otherwise go to Step 2, using the updated list $\mathcal{L}$.

The general workflow for the adaptive metamodel-based optimization process is presented graphically in Figure 2.5.

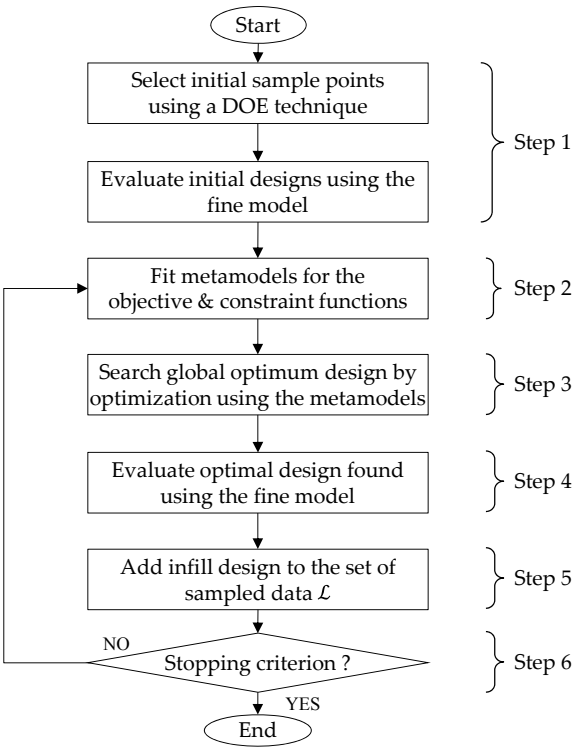


Figure 2.5 : Adaptive metamodel-based optimization process workflow

As in the previous paragraph, a metamodel is built using an initial sampling plan (generated using a space-filling strategy, such as Latin Hypercube). An optimization is launched using this metamodel and an optimum is found. This optimal point is evaluated using the fine model, is next

[7] The multi-start refers to multiple consecutive launches of the trust-region algorithm with different random initial points. This process is required when the optimization problem is expected to be multi-modal, i.e. the optimization problem has multiple local optima, or in the absence of such information. The best feasible design is retained as the global optimum.

added to the list of support points ($\mathcal{L}$) and the metamodel is rebuilt, with the hope of improving the prediction accuracy of the metamodel by the addition of supplementary information. The optimization is re-launched using the new metamodel and the process is repeated until the true location of the global optimum of the problem is found.

2.4.2 Multiple criteria

Electromagnetic optimal design problems generally have several – often conflicting – goals, and constraints. The result of such a multi-objective optimization problem is a multi-dimensional set of non-dominated trade-off designs forming a Pareto frontier. The number of dimensions of the resulting Pareto frontier equals the number of objectives of the optimization problem.

Within a multi-objective MBDO framework, a major problem that arises consists in the selection of – not just one optimal design to validate using the simulation model – but a set of non-dominated designs of a given size. Using multi-objective optimization algorithm to find the Pareto front of the metamodel at a given iteration of the MBDO algorithm, the result will consist of a large set of non-dominated trade-off solutions between the different objectives. The size of this set, in order for it to be representative for all trade-offs, increases with the number of objective functions involved.

If for the bi-objective optimization case, the selection of a representative sub-set of metamodel Pareto[8] optimal solutions to be validated using the simulation model is intuitive (both mathematically and graphically), resuming to selecting points out of a curve, for three or higher dimensions, the selection should be done among the points forming a surface, respectively a multi-dimensional surface. The difficulty consists in choosing the best-spaced designs, which, after validation with the simulation model, would provide the designer with a diversified trade-off set of designs, for him to select from.

The multi-objective adaptive MBDO technique can be described using the following steps:

- Step 1: Select an experimental design and run simulation model to calculate the outputs;
- Step 2: Build a metamodel for each objective and constraint function of the optimization problem using all simulated designs;
- Step 3: Use a multi-objective optimization algorithm (e.g. NSGA-II) to search for the non-dominated trade-off front using the metamodels built at Step 2;
- Step 4: Select a given-size well-spread sub-set of designs ($\mathfrak{Z}$) from the metamodel Pareto front obtained at Step 3;
- Step 5: Run the simulation model in order to validate the selected designs;
- Step 6: Add obtained designs to the list of evaluated designs ($\mathcal{L}$);
- Step 7: Verify stopping criteria (e.g. prescribed number of simulation model evaluations, total elapsed time, prescribed metamodels accuracy, etc.). If stopping criteria are met, then stop the algorithm; otherwise go to Step 2, using the updated list $\mathcal{L}$.

When dealing with constrained optimization problems, the constraint functions are metamodeled along with the objective functions. The constraints are then handled directly by the optimization algorithm at Step 3, through their metamodel predictions. Thus, at the beginning of

[8] The "metamodel Pareto front" refers to the non-dominated trade-off solutions found using an optimization algorithm and the metamodels of the fine model.

the MBDO process, when only a few support points are available for building the metamodels, and especially for highly non-linear constraint functions, these metamodels lack of accuracy might have an important impact on the shape of the metamodel Pareto front obtained at Step 3. This can result in an undesired reduction of the feasible design space, therefore leaving out feasible areas of the design space, or in a larger variety of non-dominated solutions, including solutions which satisfy the metamodel constraints but not the fine model calculated constraints. This is more critical when the optimal designs lie on the boundary between the feasible and unfeasible regions of the design space. To overcome this drawback, a constraint relaxation method can be employed with the MBDO process. The method consists of adding a tolerance to the metamodel prediction of the constraint, so as to reduce the chance of leaving unexplored interesting areas of the design space. Hence, the algorithm can start with a higher tolerance level (e.g. a few % of the initial constraint limit), which can be decreased progressively to zero towards the last iterations of the MBDO process, when the metamodels prediction becomes more accurate due to the infill designs ($\mathfrak{I}$) added at each MBDO iteration to the list of support points ($\mathcal{L}$).

The general workflow of the multi-objective version of the adaptive MBDO technique is presented graphically in Figure 2.6.

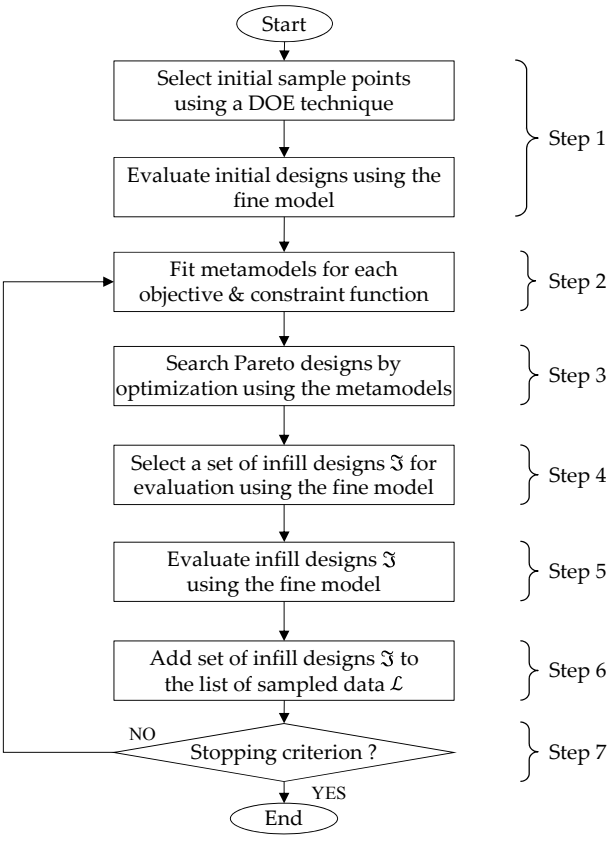


Figure 2.6 : Workflow of the multi-objective adaptive MBDO technique

When multiple optimization criteria are involved, at each iteration a set of optimal trade-off designs are to be selected instead of just one design, as for the single-objective case. This change is reflected in Step 4 of the adaptive MBDO technique, consisting of the selection of a representative sub-set of designs of given size from the large set of Pareto optimal designs obtained by the multi-objective optimization algorithm with the metamodels. The different ways of selecting a well-

spread representative sub-set of designs from the Pareto front obtained using the metamodels are presented next.

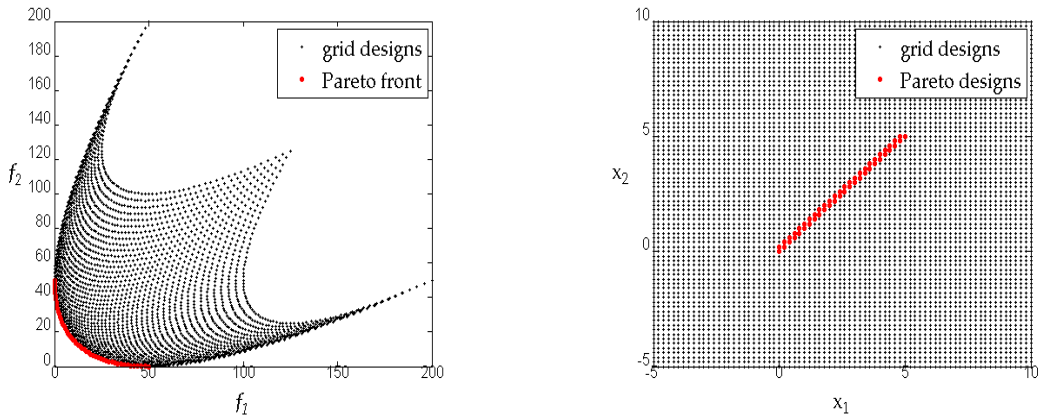
2.4.3 Well-spread sub-set selection from a metamodel-optimal n-dimension Pareto front

This stage of the MBDO process consists of selecting a reduced, but representative set of designs from the Pareto non-dominated set of points, previously obtained from the optimization using the metamodels. The selected designs are then evaluated using the fine model in order to validate their feasibility and optimality. This step represents one of the key elements of the MBDO process; the designs to be selected for validation using the fine model, at each iteration of MBDO, should be as uniformly spread across the metamodel Pareto front as possible in order to attain a wide range of optimal trade-off designs on the final Pareto front.

Let's consider the bi-objective two-dimensional unconstrained optimization test problem "Binh" [BIN 97], [GIL 09]. The mathematical formulation of the Binh optimization problem is given in (2.3).

$$\begin{aligned} \underset{x_1,x_2}{\text{Minimize}} \quad & f_1 = x_1^2 + x_2^2 \\ & f_2 = (x_1 - 5)^2 + (x_2 - 5)^2 \\ \text{with} \quad & x_1, x_2 \in [-5,10] \end{aligned} \tag{2.3}$$

The solution of the optimization problem, obtained using a grid with a step of 0.2 represented by black dots is presented in Figure 2.7. The Pareto front is represented by the red dots in Figure 2.7a and the corresponding design variables are represented by the red dots in Figure 2.7b.



a) Objective functions space b) Design space

Figure 2.7 : Optimal solution of the Binh test problem

From Figure 2.7b it can be observed that the non-dominated solutions of the Binh problem lie all on the line limited by the point of coordinates (0,0) which minimizes the 1st objective f_1, resulting in the objective couple (0,50) and the point of coordinates (5,5) which minimizes the 2nd objective f_2, resulting in the objective couple (50,0). The Pareto front is presented in Figure 2.7a.

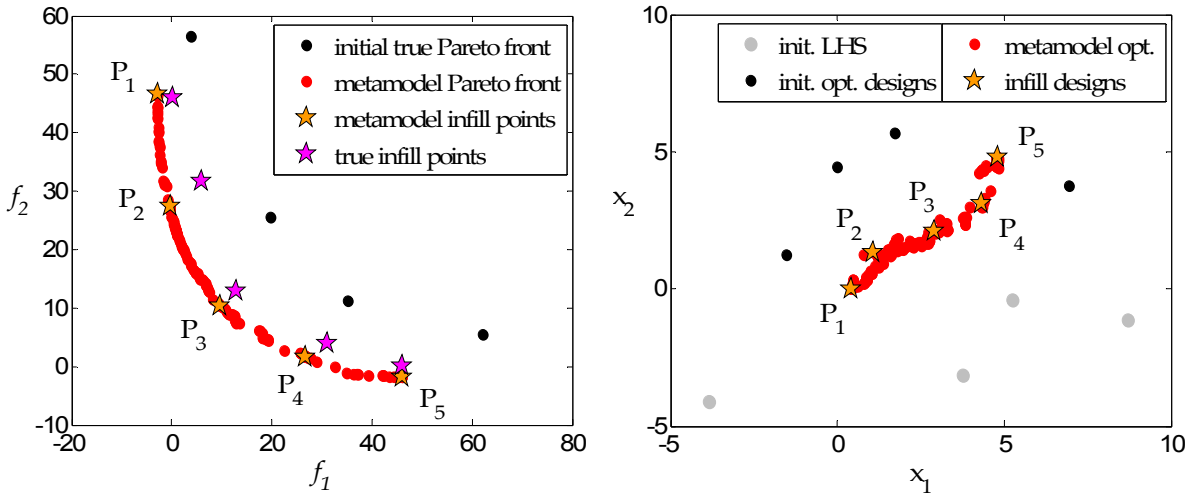
An initial experimental design of 10 points uniformly spread across the design space was considered using the Latin Hypercube Sampling (LHS) strategy. Figure 2.8 presents the resulting

initial Pareto front (black circles), which is composed of only 4 non-dominated points. Using the 10 initial LHS designs, a Kriging metamodel was created for each of the two objective functions, using the very popular Design and Analysis of Computer Experiments (DACE) toolbox implemented under Matlab®, available from [LOP 02]. The Kriging model is an interpolating metamodeling strategy, having its origin in geostatistics and was first introduced for modeling computer experiments by Sacks et al. in [SAC 89]. The mathematical expression of the Kriging predictor and the associated prediction error estimate within DACE are given in Appendix A. An optimization process is launched using the previously created metamodels of the true functions, by the means of a multi-objective genetic algorithm, NSGA-II [DEB 02], implementation of [MOU 09a]. The resulting Pareto front is presented in Figure 2.8 by the red circles.

The question that arises next is:

"Which n designs should be selected for evaluation with the fine model, in order to provide with a set of optimal compromises as diversified as possible?

The infill designs are selected with the hope that after validation using the fine model, the uniform distribution is maintained also on the true Pareto front.



a) Objective function space b) Design space

Figure 2.8 : Infill set selection for the Binh optimization test problem

In the bi-objective case, this choice is intuitive and can be made graphically. In Figure 2.8a, the 5 infill points, marked using the orange stars, were chosen among the metamodel Pareto solutions, marked by the red circles, so that the Euclidian distances between each pair of consecutive points (e.g. (P_1, P_2), (P_2, P_3), etc.) are all equal. This selection resulted in a well-spaced set of points in the design space, as can be seen from Figure 2.8b. The mathematical expression of the Euclidian distance between two points, P_i and P_j is presented in (2.4).

$$d_{P_i,P_j} = \sqrt{\left(f_1^i - f_1^j\right)^2 + \left(f_2^i - f_2^j\right)^2} \qquad i,j \in \{1,\cdots,5\},\ i \neq j \tag{2.4}$$

where d_{P_i,P_j} is the Euclidian distance between P_i and P_j, f_1^i and f_1^j are the objective function values of P_i and P_j, respectively.

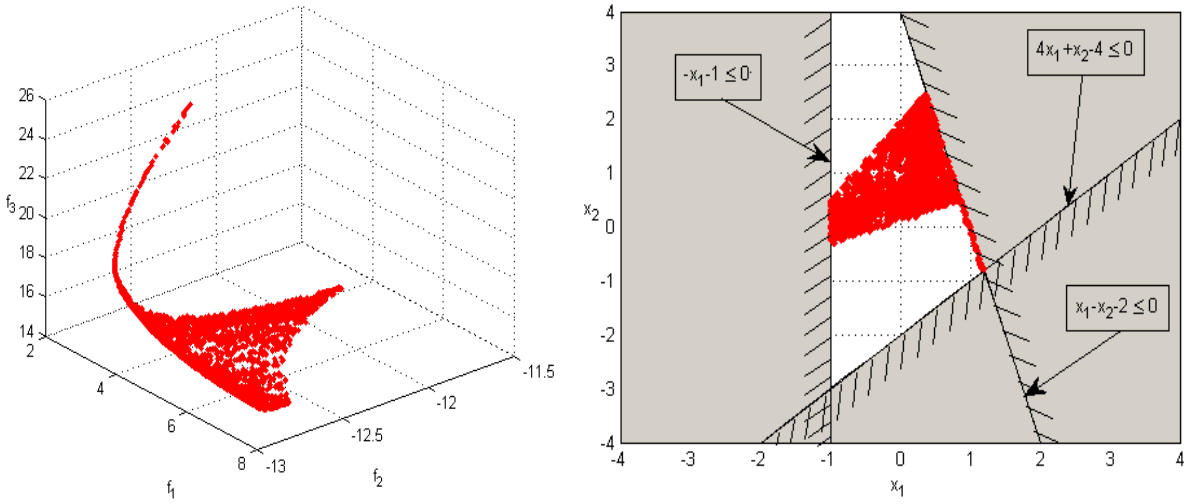
After the evaluation with the fine model (i.e. the true Binh function), the good spacing between the selected points is maintained among the points marked by the magenta stars, as can be seen from Figure 2.8a.

Things complicate if the Pareto front obtained using the metamodels is not convex or concave, it presents discontinuities, or, even worse, the Pareto front has no particular form. When the optimization problem consists of three or more objective functions, the graphical representation is less obvious and the infill point selection becomes a complex task. This is why it is necessary to clearly define the selection criteria.

To exemplify the purpose, let's consider an optimization problem consisting of three objective functions, the Viennet problem [COE 07]. The optimization problem consists of two design variables with three objective functions to minimize, and three constraint functions to respect. Therefore, the result of this optimization problem will be a 3 dimensional Pareto front. The mathematical formulation of the Viennet problem is presented in (1.5).

$$
\begin{aligned}
\underset{x_1,x_2}{\text{Minimize}} \quad & f_1 = \frac{(x_1-2)^2}{2} + \frac{(x_2+1)^2}{13} + 3 \\
& f_2 = \frac{(x_1+x_2-3)^2}{175} + \frac{(2x_2-x_1)^2}{17} - 13 \\
& f_3 = \frac{(3x_1-2x_2+4)^2}{8} + \frac{(x_1-x_2+1)^2}{27} + 15 \\
\text{with} \quad & x_1, x_2 \in [-4,4] \\
\text{subject to} \quad & g_1 = 4x_1 + x_2 - 4 \leq 0 \\
& g_2 = -x_1 - 1 \leq 0 \\
& g_3 = x_1 - x_2 - 2 \leq 0
\end{aligned}
\tag{2.5}
$$

The true Pareto front[9] of the Viennet problem is known and presented here in Figure 2.9a. The constraints are linear functions, limiting the design space, as presented in Figure 2.9b.



a) Objective space b) Design space

Figure 2.9 : Pareto front of the Viennet optimization test problem

[9] The notion "true Pareto front" refers to the Pareto frontier of the simulation model, thus composed uniquely of designs evaluated using the simulation model.

The three constraint functions will not be metamodeled; instead, they will be handled by the optimization algorithm as "inexpensive constraints"[10], thus using their true expression.

Random sub-set selection strategy

The most basic and intuitive approach for selecting a uniformly spread infill subset of the Pareto metamodel front is through random selection. A large number of subsets of randomly selected designs from the metamodel Pareto front are generated. From all these subsets it is selected the one that presents the greatest Euclidian distance between each of its elements. To summarize, the random strategy for selection of new infill points carries out the following steps:

- Step 1: Generate a large number (>1000) of subsets of randomly selected designs from the metamodel Pareto front;
- Step 2: Calculate the Euclidian distance in the objective space between the designs of each candidate subset;

$$d_{i,j}^k = d_{j,i}^k = \sqrt{\left(f_{1,i}^k - f_{1,j}^k\right)^2 + \left(f_{2,i}^k - f_{2,j}^k\right)^2 + \cdots + \left(f_{m,i}^k - f_{m,j}^k\right)^2} = \sqrt{\sum_{l=1}^{m}\left(f_{l,i}^k - f_{l,j}^k\right)^2} \tag{2.6}$$

where $d_{i,j}^k$ represents the Euclidian distance in the objective space between designs i and j of the k-th candidate subset and m is the number of objective functions of the optimization problem.

- Step 3: Retain the minimum distance for each candidate subset, d_{min}^k ;

$$d_{min}^k = \min_{i,j} d_{i,j}^k \tag{2.7}$$

- Step 4: Order descending all subsets according to the values of d_{min}^k calculated at Step 3;
- Step 5: Select the subset with the greatest value for d_{min}^k.

The example of the Binh test problem from 2.4.3 is considered. The Euclidian distance expressed in (2.6) is represented graphically in Figure 2.10 for the pair of points (P_2,P_3).

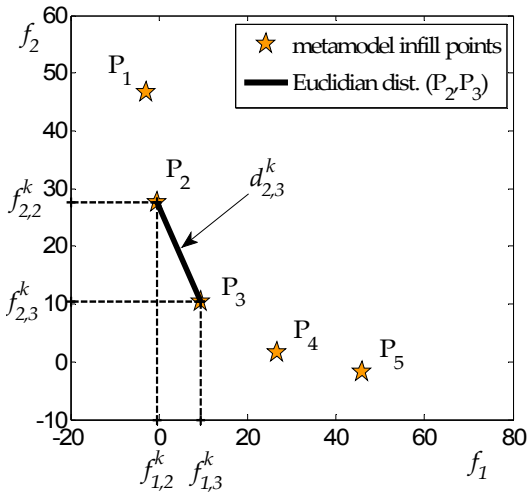


Figure 2.10 : Euclidian distance in the objective space between two points, P_2 and P_3

[10] The "inexpensive constraints" represent those constraint functions whose expression is a simple analytical combination of the design variables (e.g. geometrical constraints), thus easy to evaluate, no needing to run the expensive simulation model. In exchange, the "expensive constraints" are those constraint functions that depend on the output parameters of the simulation model, thus requiring the evaluation of the expensive model in order to compute their value.

The described selection strategy is simple to implement but its computation effort is directly related to: a) the number of objective functions, b) the size of the metamodel Pareto front, c) the number of candidate subsets, and d) the infill set size. Hence, an important drawback is that it can quickly become time consuming when, for example, the number of candidate subsets is large and/or the size of the Pareto set issued from the optimization on metamodels is important. Another drawback is given by the fact that the diversity of solutions in the infill set is directly related to the number of candidate subsets. In order to ensure a uniform distribution of solutions in the infill set, a higher number of candidate subsets must be considered, thus augmenting considerably the computation expense.

Systematic best-spread sequential selection strategy

This time, instead of generating and testing a number of candidate subsets of randomly chosen designs from the metamodel Pareto front, a systematic selection of the design with the greatest Euclidian distance (in the objective space) from the already selected points of the infill set is considered. The sequential selection process is described using the following steps:

- Step 1: Select a design from the metamodel Pareto front (random or extreme point, for instance, the point with the minimum value for one of the objective functions) and add it to the infill set;
- Step 2: Calculate the Euclidian distance between all existing points in the infill set and all the remaining designs of the metamodel Pareto front, using equation (2.6);
- Step 3: Select the design with the greatest Euclidian distance (in the objective space) to the existing designs in the infill set (Steps 3 and 4 of random sub-set selection strategy);
- Step 4: Add the design to the infill set and continue with Step 2.

This selection strategy benefits from the ease of implementation. A good spreading between points is ensured by systematically choosing the point furthest from the existing points. However, a drawback which should be mentioned consists of the fact that a considerable number of designs will be selected on the extremities of the metamodel Pareto front. If the metamodels of the objective and/or constraint functions have a low accuracy, the designs selected close to the limit of the feasible domain have high chances of not satisfying the constraints after validation using the fine model. If the metamodel Pareto front is not continuous and has many discontinuities, isolated groups of designs on the Pareto front might be missed. The computational cost of this strategy is directly correlated with the size of the metamodel Pareto front and especially, the desired size for the infill set. The greater the size of the infill set and/or the metamodel Pareto front, the greater is the computational cost of this procedure.

Discrete optimization-based selection strategy

To ensure a uniform distribution of designs inside the infill set, a strategy based on a discrete optimization process is considered. This strategy is an automation of the random sub-set selection strategy. The uniform distribution of designs inside the infill set is ensured by the convergence of the discrete optimization algorithm. The discrete optimization problem has a number of variables equal to the size of the infill set and one objective function to maximize, the distance in the objective space, between all designs of the infill set. The variables of the discrete optimization problem are

represented by the indexes of the designs from the list of the metamodel Pareto front. The mathematical formulation of the discrete optimization problem is given in (2.8).

$$\begin{aligned} \underset{\mathbf{id}}{\text{Maximize}} \quad & f = \min_{i,j}\left(d_{i,j}\right) \\ \text{with} \quad & \mathbf{id} = \left[id_1, id_2, \cdots, id_{n_{is}}\right], id_i \in \{1,2,\cdots,n_{nd}\} \\ & i,j = \overline{1 \cdots n_{is}},\ i \neq j \end{aligned} \tag{2.8}$$

where n_{is} represents the size of the infill set, n_{nd} is the size of the metamodel Pareto front, id_i is the index of the i-th of the n_{is} designs in the list of metamodel Pareto front.

The variables of this optimization problem are integers, representing position indexes in the list of metamodel Pareto front. The maximization of the distance described aims to uniformly spread the n_{is} points of the infill set. The uniform distribution of points inside the infill set is thus strongly related to the convergence of this optimization process.

Selection point strategies discussion

In the previous paragraphs there have been reviewed three methods of selecting a set of well-spread designs from the metamodel Pareto front. The goal of this step of the algorithm is to supply the algorithm with a reduced set of potentially optimal designs from a very large set ($n_{sel} \ll n_{Pareto}$) of Pareto designs obtained using the metamodels. These designs are next evaluated using the fine model and hopefully the good spreading will be maintained also on the true Pareto front. Each selection strategy has its own advantages and drawbacks. A summary of the advantages and drawbacks of all three selection strategies has been represented in Figure 2.11 using a spider diagram[11] representation. The desired criteria values are found at the extremities of beams.

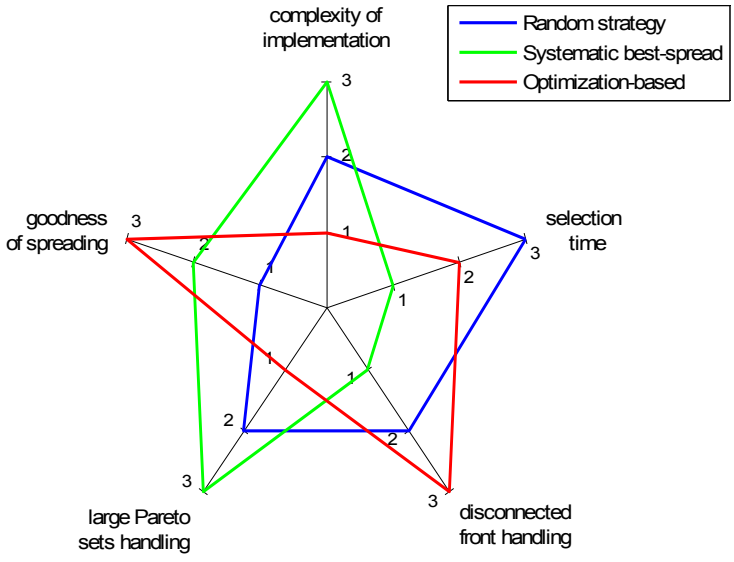


Figure 2.11 : Summary of the selection point strategies

[11] The spider diagram, also known as radar diagram or chart, is a method used in statistics for visualizing multivariate data. It allows multiple data sets comparison. More information available online: http://en.wikipedia.org/wiki/Radar_chart.

2.4.4 Application: LIM device design optimization problem

The MBDO technique was applied with success to the optimal sizing of a double-sided linear induction motor (LIM) device [GON 11]. The LIM device to be conceived represents a reduced-scale prototype of a device designated for railway system applications. The device to optimally conceive consists of two symmetrical primaries, placed face-to-face, and a secondary consisting of an aluminum plate, which is placed between the two primaries. Each of the two primaries has three concentrated windings. The static part, represented by the aluminum plate is installed on the ground, while the mobile parts, represented by the two primaries, are installed on the train. The basic structure of the double-sided LIM device is presented in Figure 2.12.

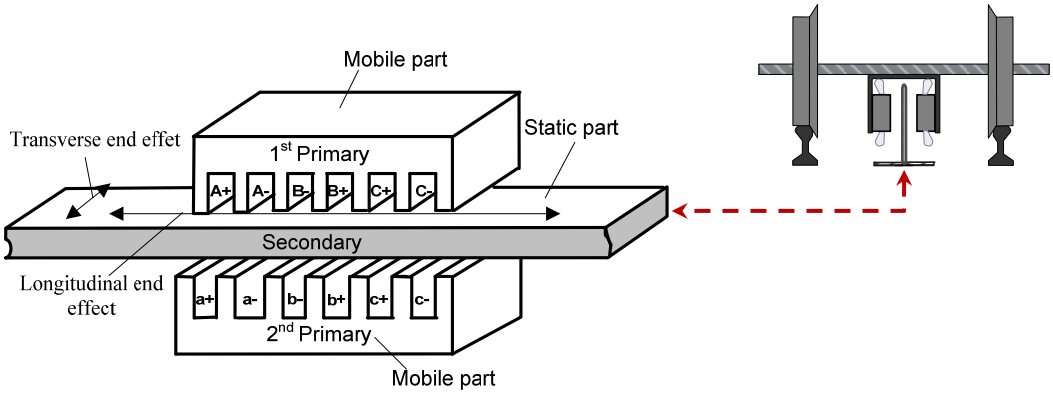


Figure 2.12 : Double-sided linear induction motor (LIM) device

The primary windings are fed from a three-phase AC symmetric voltage system, producing a travelling magnetic field, which induces eddy currents in the aluminum plate. The travelling magnetic field and the eddy currents give birth to a thrust force, which shifts the two primaries along the length of the aluminum plate. The model of the device is parameterized i.e. the dimensions of different elements of the device are easily modifiable.
This optimization study focuses on the optimal sizing of the primary of the device.

2.4.4.1 Modeling of the LIM device

A 3D finite element modeling of the LIM device has been previously developed at the L2EP laboratory, within the framework of a previous PhD thesis [GON 11a]. Due to the complexity of the physical phenomena associated with the functioning of the LIM, the 3D finite element method was considered in order to analyze the behavior of the device. To account for the two main parasite effects: the longitudinal end effect, given by the finite length of the device, and the transverse edge effect, given by the finite width of the primaries, a 3D electromagnetic modeling imposed. Furthermore, to account for the temperature influence on the operation of the LIM, a 3D thermal model has been developed. Both electromagnetic and thermal models were developed using a commercial FEM software kit, Opera 3D, a product of Cobham Company [COB 12]. The two solvers, magnetic and thermal, are managed through command lines, in batch mode, by Matlab.

In order to accurately predict the behavior of the device, the two models, electromagnetic and thermal, have been coupled. The electrical losses, consisting of the iron and Joule losses are considered as unique heat source. The magnetic model needs the conductor temperature information, which is supplied by the thermal model. To compute the conductor temperature, the thermal model needs to take into account the iron and Joule losses, supplied by the magnetic model. The coupling between the two models is graphically presented in Figure 2.13.

The interaction between several disciplines of a system makes the object of the Multidisciplinary Design Optimization (MDO) [CRA 94], [BAL 94]. The resolution of these interactions is performed using a fixed point iteration (FPI) method. This problem is called Multidisciplinary Design Analysis (MDA).

Several approaches exist for solving such optimization problems. Among them, the classic approach consists of placing the MDA with its solver directly within the optimization process. The consistency of the interactions between the disciplines is ensured by the system analyzer (the FPI). The optimizer is only concerned with the resolution of the optimization problem, in order to find the optimal results respecting the constraints. This type of problem formulation is known under the name of Multidisciplinary Feasible (MDF) formulation, and its expression is given in (2.9) [GIL 09].

$$MDF = SO[MDA] = SO\left[SA[EMM \rightarrow TM]\right] \tag{2.9}$$

where *SO* represents the system optimizer, *SA* is the system analyzer, *EMM* and *TM* represent the electromagnetic, respectively the thermal discipline, which are evaluated sequentially (denoted by the "→" sign).

The system analyzer of the coupled model, the FPI, uses in the mean from 4 to 6 iterations to converge. Depending on the configuration, one coupled model evaluation (i.e. the evaluation of a given set of design variables) takes between half hour and two hours. Hence, the evaluation time of the model is prohibitive for traditional optimization approaches, such as genetic algorithms.

The multi-objective adaptive MBDO algorithm, using the discrete optimization-based infill point selection strategy, was used for the MDF optimization of the LIM device.

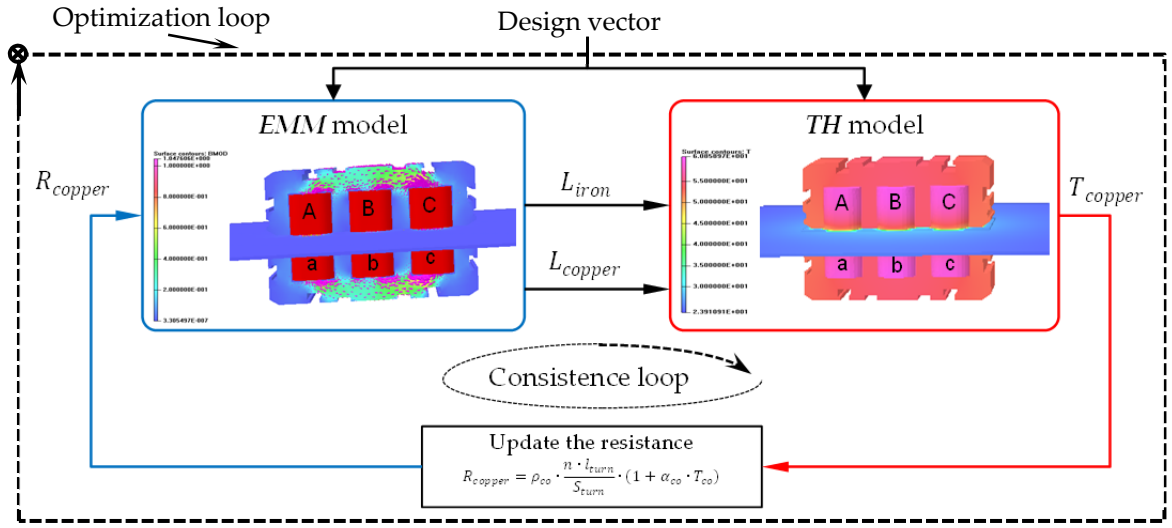


Figure 2.13 : The coupling between the electromagnetic (EMM) and thermal (TH) models

2.4.4.2 Optimization problem formulation

The optimal sizing of the LIM device, using the coupling model previously presented will be done by accounting for several optimization criteria. The goal of this optimization study is to find those designs which present a small mass, a reduced level of iron and copper losses, and a force as important as possible, while respecting the thermal feasibility constraints. However, the above mentioned criteria are antagonist (e.g. a small mass design would also present a light force, but also an important amount of losses), thus imposing a three-objective optimization problem formulation. The mathematical expression of the optimization problem is given in (2.10).

$$\begin{aligned} \underset{\mathbf{x}}{\text{Minimize}} \quad & f_1 = Mass(\mathbf{x}) \\ & f_2 = Losses(\mathbf{x}) \\ & f_3 = -Force(\mathbf{x}) \\ \text{with} \quad & \mathbf{x} = [tw_1, tw_2, tw_3, U] \\ & tw_1 \in [5,12],\ tw_2 \in [5,10],\ tw_1 \in [5,10],\ U \in [0,20] \\ \text{subject to} \quad & T_{co} - 200 \leq 0 \end{aligned} \tag{2.10}$$

where *Mass* represents the total mass of the device, *Losses* represents the sum of iron and copper losses, *Force* is the Maxwell force, tw_1 is the width of the two end teeth, tw_2 is the width of the teeth in the center of the windings, tw_3 is the width of the teeth between the windings, U represents the voltage applied to the primary, and T_{co} is the copper parts temperature. The different geometrical dimensions considered for optimization are presented graphically in Figure 2.14.

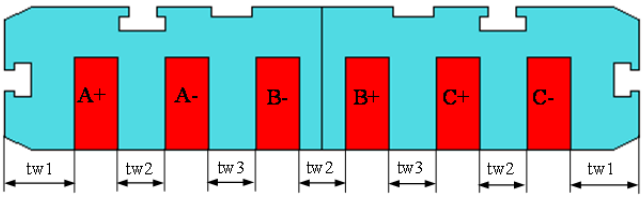


Figure 2.14 : Geometrical variables of the optimization problem

2.4.4.3 Optimization results

Due to the computationally expensive character of the 3D FE model of the LIM, a limited budget of model evaluations was imposed to the optimization process. The total budget of 150 model evaluations was divided in two quantities. Thus, the first 50 LIM device configurations were initially selected using a space-filling strategy, by the means of a Latin Hypercube Sampling (LHS). These device configurations were then evaluated using the fine model and for each of the three objective and one constraint functions, a Kriging metamodel was created. The rest of 100 device configurations have been selected by the MBDO process in packages of 10 designs, during a total of 10 MBDO iterations. Figure 2.15 presents the Pareto designs, marked with blue dots, obtained from the optimization using the metamodels of the objective and constraint functions at the first iteration

of the process. Among the total 10 000 metamodel evaluations of the genetic algorithm used to find the metamodel Pareto front, about 4 000 designs are Pareto optimal (the blue dots). The infill set selection strategy based on a discrete optimization process was used to select 10 well spread designs for evaluation with the fine model (FEM) of the LIM, represented by the large black dots in Figure 2.1. The red-filled triangles in Figure 2.15 represent the designs evaluated using the FEM of the LIM device. One can remark that there is a small difference between the metamodel predicted optimal designs and the designs evaluated using the FE model, meaning that the metamodels predict with a good accuracy the outputs of the fine model of the LIM.

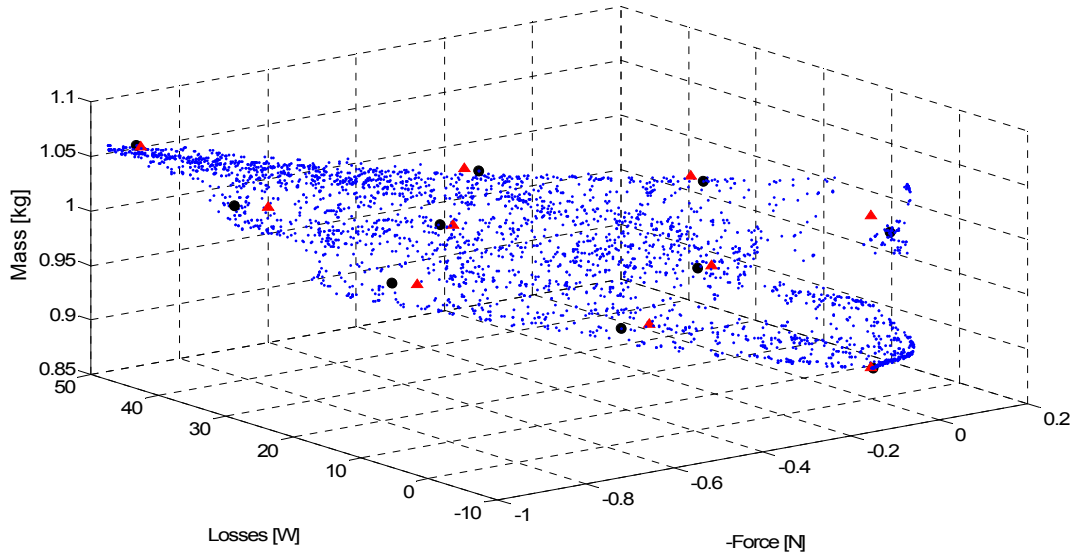


Figure 2.15 : Infill set selection of 10 well-spread designs for evaluation using the FE model at the 1st MBDO iteration of the LIM device optimization

The final Pareto front of the LIM device optimization using the presented MBDO process is presented in Figure 2.16.

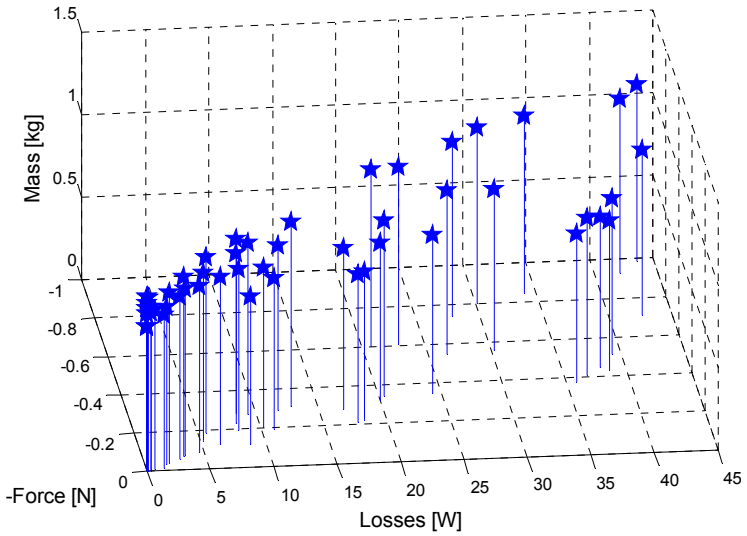


Figure 2.16 : Pareto front of the LIM device optimization using MBDO

The Pareto trade-off designs between the three objectives of the LIM device optimization problem, represented by the stars in Figure 2.16 are computed with accuracy, being issued from the fine model (FEM) of the LIM device. A good spreading of the non-dominated points along the 3D Pareto front can be remarked from Figure 2.16, thus providing the designer with a large choice of trade-off designs for him to make his final choice from.

The optimization process with an imposed budget of 150 FE model evaluations needed a week of computation. The elapsed time for the MBDO process of the LIM device was decomposed following the main steps of the algorithm. This time decomposition is presented graphically in Figure 2.17 below.

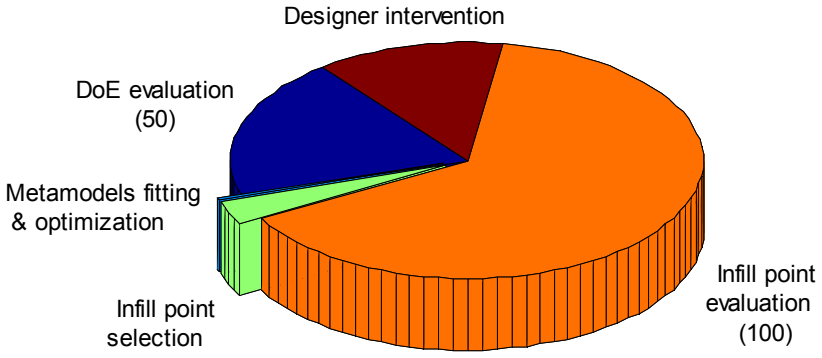


Figure 2.17 : LIM optimization time decomposition

From Figure 2.17, one can see that the total optimization time is dominated by the FEM evaluation for the infill designs considered at each iteration of the optimization process. At the opposite end, the time consumed by the metamodeling process and the optimization using metamodels is completely negligible, with less than 1% of the optimization time.

A particular time consuming step of the optimization process of the LIM device is represented by the designer's intervention within the optimization process in order to re-launch the failed FEM simulations. For certain configurations, due to the mesh discretization, the volume mesh generation process might fail, causing the optimization process to stop. A less occurring event which makes the optimization process stop is the FEM analysis failure. In these cases, the designer has to step in, manually re-mesh the sensitive regions of the device and re-launch the FE model analysis. All this wasted time and effort of the designer strongly advocates for an "as robust as possible" FE modeling of the device, prior to its integration within an optimization process. The proposed strategy allows bypassing this problem by simply ignoring the failed design evaluations. Once all the simulation model evaluations at a given iteration are completed (successfully or failed), the failed designs are left aside when the new metamodels are fitted. This represents one of the most important advantages of this method, obtaining optimal results even with failing simulations.

However, the adaptive metamodel-based optimization approach might be deceiving, resulting in the convergence to suboptimal designs, instead of the desired real optimal design. To overcome this obstacle, a strategy that manages both the global accuracy of the metamodel and the search for optimal solutions imposes. This approach is next presented in the following paragraph.

2.5 Criterion-based metamodel optimization

The metamodel that has been built and used in the previous paragraph is just an approximation of the fine model that is to be optimized. The previous MBDO strategy works well when the metamodel provides a good approximation of the fine model. In the case that the objective and constraint functions present strong non-linearities, with lots of peaks and valleys which can be very difficult to model, the previous MBDO strategy has strong chances to fail in finding the optimal solution(-s). Furthermore, in order to accurately predict the outputs of the fine model, a large set of well-spread training points is needed. The size of the design space is directly related to the number of design variables and their domain of variation. When the number of design variables is important, and their domain of variation is wide, the size of the design space becomes enormous. Hence, it is impossible to uniformly sample the design space and create a metamodel of the fine model with fair accuracy, with a limited budget of fine model evaluations (tens up to a few hundreds).

Attaining a highly accurate metamodel is however not necessary, nor desired for this MBDO strategy. Instead of wasting computational effort for sampling non-optimal, thus non-interesting, zones of the design space, the focus should be on promising areas, with a high chance of detaining the optima. Therefore, the idea of the strategy in this paragraph is to simultaneously seek the optimal solution(-s) (exploit the metamodels), while improving the global prediction accuracy of the metamodels (explore the design space). These two features can be aggregated within the expression of an update or infill criterion. The optimization process will thus be guided by a unique criterion for adding new points, called "*infill point selection criterion*" or just simply "*infill criterion*", which accounts for the optimal designs and the metamodel accuracy at the same time. An equivalent term for the infill criterion is "utility function", preferred by some other authors [HAW 07], or "figure of merit", "merit function", notion preferred by other authors [SOB 05].

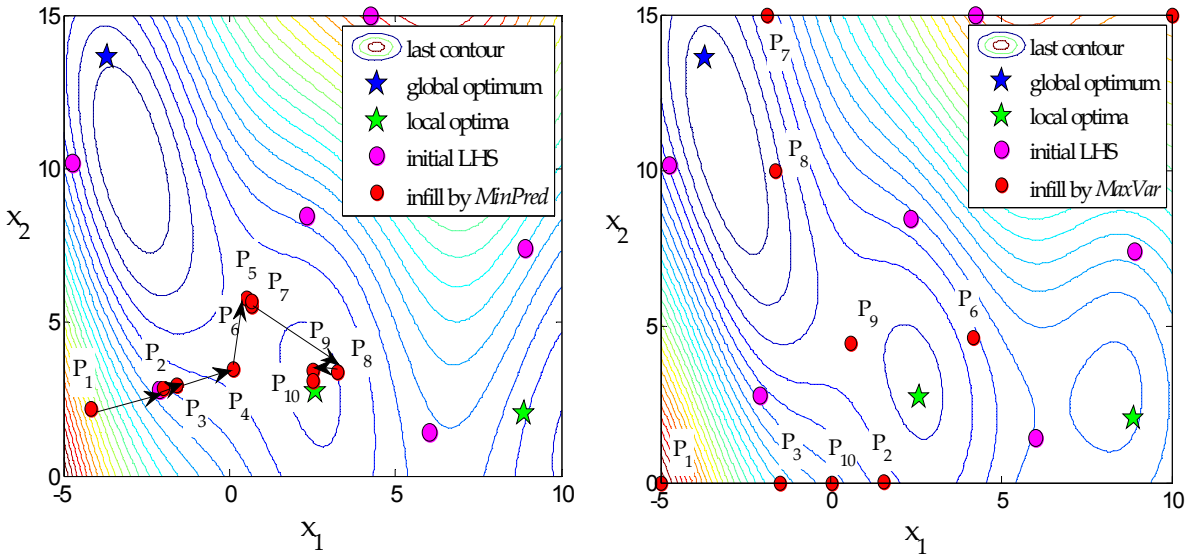
2.5.1 Single-objective infill point selection criteria

A detailed taxonomy of infill criteria for optimization with global metamodels is provided by Jones et al. [JON 01]. The simplest infill criterion is the metamodel predictor itself. This idea consists in optimizing directly using the prediction supplied by the metamodels. This "minimize prediction" ("*MinPred*") strategy is equivalent to the idea presented in paragraph 2.4. The focus in this case is totally directed on the exploitation of the metamodels, assuming a good level of global accuracy for the metamodels.

Another possible idea consists in finding the design that presents the maximum of uncertainty in the metamodels prediction (i.e. the design that gives the maximum predicted error). Sampling sequentially in those points that present the maximum of predicted error (maximum variance) will result in a progressive improvement of the global accuracy of metamodels. Note, however, that in this case, there is no mechanism of directing the search towards areas susceptible of containing the optimal solution(-s) of the problem. This criterion focuses exclusively on the exploration of the design space, while neglecting the search for optimal designs. In [SAS 02b], this uncertainty measure is named "*MaxVar*" or the "maximum variance" criterion, and represents a simplified version of the Watson and Barnes "minimize surprises" criterion [WAT 95] with application to the

geostatistics. As its name suggests, this criterion aims to reduce the metamodel prediction surprises by minimizing the maximum probability of finding an important discrepancy between the predicted value and the true value of the function at a given point. Along with this measure, two other criteria were proposed in [WAT 95] and further studied in [SAS 02b], the "locate the threshold-bounded extreme" and "locate the regional extreme" criteria. These two criteria are exactly and very similar to the probability of improvement (*PI*), respectively the expected improvement (*EI*).

In order to show the action of the two basic criteria, *MinPred* and *MaxVar*, we consider the modified Branin-Hoo function. An experimental design of 6 points is selected using the Latin Hypercube Sampling (LHS) strategy, represented by the magenta circles in Figure 2.18. A Kriging metamodel is created using the initial experimental design. A total number of 10 infill points is then selected using the *MinPred* criterion, in Figure 2.18a, respectively the *MaxVar* criterion, in Figure 2.18b. Favoring exclusively the metamodel exploitation, the *MinPred* criterion searches locally, ending up in finding one of the local optima of the function, as shown by the trajectory of points P_1 to P_{10} marked with red circles in Figure 2.18a. On the contrary, the *MaxVar* criterion performs a design space exploration, globally improving the quality of the Kriging metamodel, as can be seen from the shape of the final Kriging metamodel contour in Figure 2.18b, which is very close to the true function, previously presented in Figure 2.4a. In this case, an important number of infill points (6 points) were placed on the border of the design space, while only 4 points were selected in less accurate spots inside of the design space, hence improving metamodels global accuracy.



a) *MinPred* (exploitation only) b) *MaxVar* (exploration only)

Figure 2.18 : Basic infill criteria, *MinPred* (exploitation) and *MaxVar* (exploration) for the modified Branin-Hoo function

Both *MinPred* and *MaxVar* criteria present their advantages and their drawbacks, representing the extreme figure-cases of balance between metamodel exploitation and exploration. Based on the combination of these two criteria, a number of more complex infill criteria, with integrated mechanisms for balancing the metamodels exploration/exploitation have been proposed by Sasena [SAS 02b].

The combination of the two features previously mentioned, within the expression of an infill criterion can be done in a number of different manners. Among these criteria, the probability of improvement (*PI*) and the expected improvement (*EI*) have shown a great popularity.

Lower Confidence Bound

The simplest infill criterion consists of minimizing a lower confidence bound (*LCB*), as proposed by Cox and John in their SDO algorithm [COX 97]. The expression of this statistical lower bound is a linear combination of the predictor $\hat{y}$ and the estimated error $\hat{s}$, as in (2.11).

$$LCB = LCB(\mathbf{x}) = \hat{y}(\mathbf{x}) - \alpha\hat{s}(\mathbf{x}) \tag{2.11}$$

where α represents a constant parameter which controls the balance between metamodels exploration and exploitation.

The two extreme figure-cases of exploitation only, respectively exploration only, can be found for different values of the constant α. Thus, for $\alpha = 0$, $LCB = \hat{y}(\mathbf{x})$, yielding a criterion that exploits the metamodels, optimizing directly using their prediction. The other extreme case, when $\alpha \to \infty$, then the first term of LCB becomes negligible and minimizing LCB becomes equivalent to maximizing $\hat{s}$, therefore improving the global quality of metamodels.

From a practical point of view, the use of the lower confidence bound as an infill criterion is very marginal, due to the choice of the user-defined parameter α. In order to choose pertinent values of α for a good balance between exploration and exploitation, a certain insight into the pattern of the objective function is required, which is not always obvious, nor desired in most cases.

Probability of Improvement

One of the most popular infill criteria in the literature is the probability of improvement [JON 01]. The Probability of Improvement (*PI*) infill criterion represents the probability of the current best objective value being improved by sampling at a given point. The expression of the *PI* criterion is given in (2.12).

$$PI = P[I(\mathbf{x})] = \Phi\left(\frac{I - \hat{y}}{\hat{s}}\right) = \frac{1}{\hat{s}\sqrt{2\pi}}\int_{-\infty}^{0} e^{-[I-\hat{y}]^2/(2\hat{s}^2)}dI \tag{2.12}$$

with $I = I(\mathbf{x}) = \max\{f_{min} - \hat{y}(\mathbf{x}), 0\}$

where Φ represents the normal cumulative distribution function.

However, the *PI* criterion quantifies the probability of improving the best known value of the objective function by a certain value, but it does not give any indication on the amount of this improvement. The probability of finding a better solution is stronger in the region containing the current best known solution and having less uncertainty. Thus, this makes the *PI* criterion guide the search towards local optimal solutions.

Expected Improvement

The expected improvement (*EI*) criterion was first used by Schonlau [SCH 97] in 1997. An optimization algorithm based on the expected improvement criterion, called "Efficient Global

Optimization" (EGO), was proposed by Jones et al. in 1998 [JON 98]. Jones' EGO algorithm stands as reference in the field of metamodel-based design optimization and served as research basis for many researchers over the time.

The expected improvement criterion quantifies the amount of improvement expected to be attained by sampling at a certain point. The *EI* can be seen as an extension of the previously-presented probability of improvement criterion, proposing a quantification of the improvement given by the *PI*.

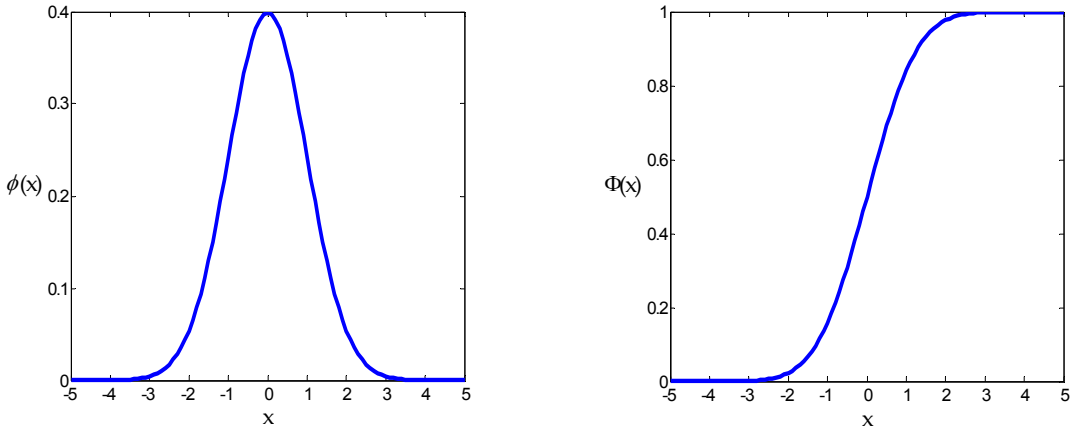
The mathematical formulation of the *EI* criterion is given in (2.13).

$$EI = E[I(\mathbf{x})] = \begin{cases} (f_{min} - \hat{y})\Phi\left(\frac{f_{min} - \hat{y}}{\hat{s}}\right) + \hat{s}\phi\left(\frac{f_{min} - \hat{y}}{\hat{s}}\right) & \text{if } \hat{s} > 0 \\ 0 & \text{if } \hat{s} = 0 \end{cases} \tag{2.13}$$

where ϕ and Φ represent the normal probability density function, respectively the normal cumulative distribution function, represented graphically in Figure 2.19.

In the expression of *EI*, one can distinguish the two terms corresponding to the exploitation of the metamodels (first term), respectively the exploration of the design space (second term).

When the value of the predicted error $\hat{s}$ is zero (i.e. point already sampled), the *EI* becomes null, meaning that for this point there is no expectation of improvement. If the predicted error $\hat{s}$ is different from zero, but small, and the predicted value of the function $\hat{y}$ is very small, in comparison with the current best known value of the function f_{min}, then the first term of the expression (2.13) becomes predominant. Thus, the search is performed locally, exploiting the good accuracy of the metamodels prediction. Otherwise, if the predicted error $\hat{s}$ is important, then the second term in (2.13) takes control, looking to explore areas of the design space with high metamodel inaccuracy.



a) Normal probability density function, ϕ b) Normal cumulative distribution function, Φ

Figure 2.19 : Normal probability density and normal cumulative distribution functions

Generalized Expected Improvement

The same team of researchers that integrated the expected improvement within the EGO algorithm, proposed in 1998 an extension of the *EI* criterion to a general case [SAS 02b]. The novel criterion, named generalized expected improvement (*GEI*) criterion proposes a control mechanism on the

balance between the metamodels exploitation and exploration. The expression of the *GEI* criterion is presented in (2.14).

$$GEI = E[I^g(\mathbf{x})] = \hat{s}^g \sum_{k=0}^{g} (-1)^k \left(\frac{g!}{k!\,(g-k)!}\right) u^{g-k} T_k$$

where $u = u(\mathbf{x}) = \dfrac{f_{min} - \hat{y}(\mathbf{x})}{\hat{s}(\mathbf{x})}$

$$T_k = -\phi(u)u^{k-1} + (k-1)T_{k-2}$$

$$T_0 = \Phi(u)$$

$$T_1 = -\phi(u)$$

$$\hat{s} = \hat{s}(\mathbf{x}) = \sqrt{\widehat{MSE}(\mathbf{x})}$$

$$g = 0,1,2,\cdots,n \tag{2.14}$$

The balance between exploration and exploitation is controlled by the means of one single parameter, g. One can see that increasing the value of g, the prediction error $\hat{s}$ becomes more important, thus the focus being more on the metamodel exploration, and the improvement of the global accuracy of metamodels.

In the expression of the *GEI*, for certain values of g we find the already overviewed *PI* and *EI* criteria. These cases are resumed in Table 2.3 below.

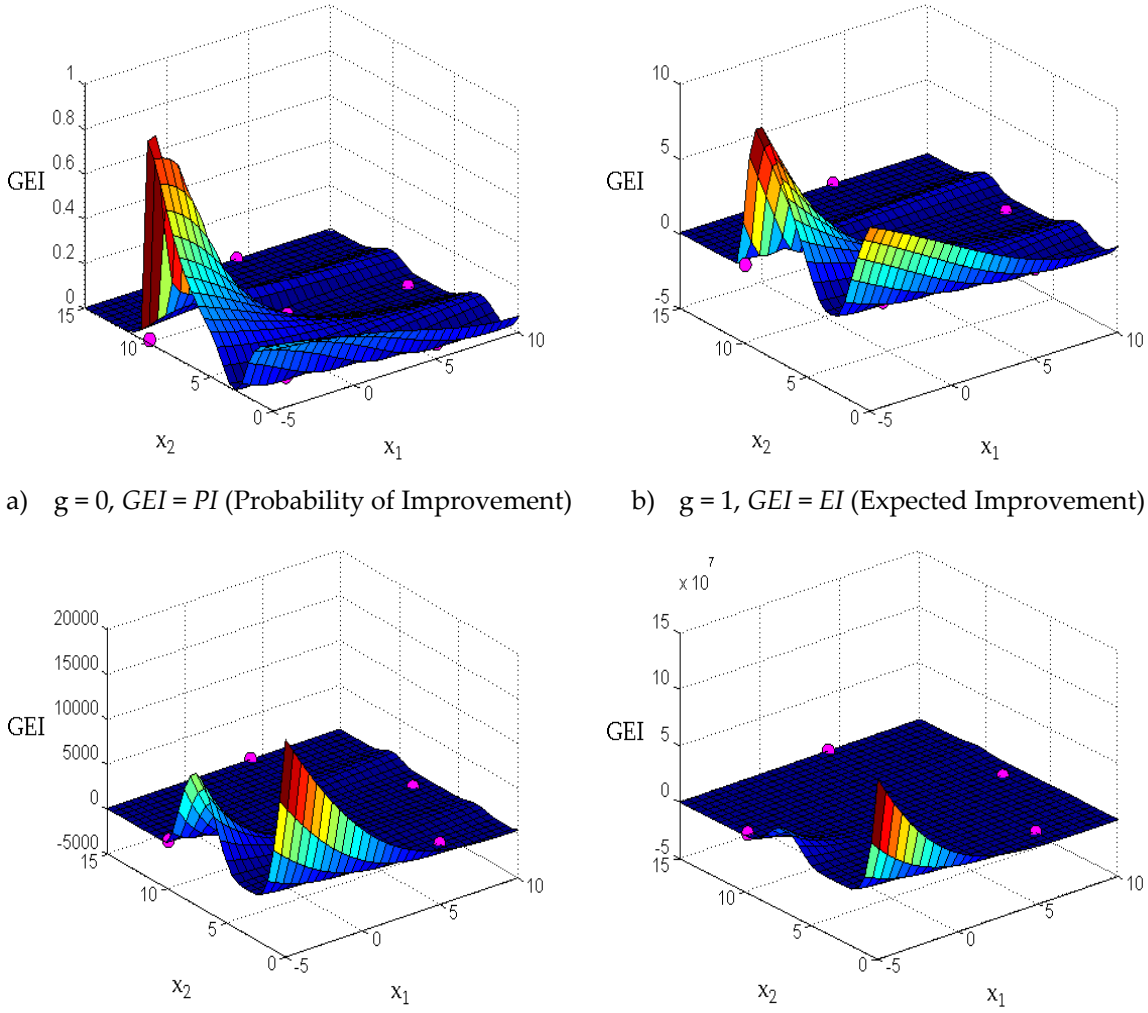
Table 2.3 : Special cases of *GEI*

Values of g	***GEI* expression**	**Infill criterion**
0	$GEI = E[I^0(\mathbf{x})] = PI(\mathbf{x})$	Probability of Improvement
1	$GEI = E[I^1(\mathbf{x})] = EI(\mathbf{x})$	Expected Improvement

To assess the influence of the g parameter on the value of *GEI* criterion, thus on the metamodels exploration/exploitation, several values of $g = \{0,1,\cdots,5\}$ were considered for the modified Branin-Hoo function. The impact of g on the value of *GEI* is presented in Figure 2.20. In the first case, when $g = 0$ a high probability of improvement is detected in the vicinity of a sampled point ($\mathbf{x} = [-5,10]$), as shown in Figure 2.20a. This point is characterized by a small predicted function value $\hat{y}$ and a high accuracy (small prediction error $\hat{s}$). When $g = 1$ the *GEI* criterion yields an expected improvement both at the point $\mathbf{x} = [-5,10]$ and at the point $\mathbf{x} = [-5,0]$, as presented in Figure 2.20b. The latter is characterized by a low metamodel accuracy (important value for the prediction error $\hat{s}$). As the value of g is increased ($g = 3$ and $g = 5$), the focus is moved from local improvement towards global metamodel improvement (extreme point $\mathbf{x} = [-5,0]$, with high prediction error $\hat{s}$), as shown in Figure 2.20c and Figure 2.20d. Hence, by tuning the g parameter, the balance between exploitation and exploration can be turned in the favor of one or the other, as desired.

From Figure 2.20, one important feature that should be remarked is the shape of the different infill criteria, consisting of a plane surface over most of the design space, especially for higher

values of the *g* parameter. This might be tricky for most classical optimization algorithms and the use of a global algorithm such as GA or a local algorithm such as SQP with a multi-start strategy is required to determine the most promising sites from the infill criterion point of view.



a) g = 0, *GEI* = *PI* (Probability of Improvement)

b) g = 1, *GEI* = *EI* (Expected Improvement)

c) g = 3

d) g = 5

Figure 2.20 : Impact of *g* on the *GEI*, for the modified Branin-Hoo function

2.5.2 *Adaptive infill strategies*

The infill criteria overviewed in the previous paragraph, except for the two extreme cases, "minimize prediction" and "maximum variance", dispose of a lever to throttle between the metamodel exploitation and exploration. Leaning this balance in the favor of one or the other can be done in a number of different ways. Before overviewing some of the possible strategies of guiding the search towards the global optimum of a function using the already presented infill criteria, some assumptions have to be made:

Hypothesis 1: Sampling at the site with the highest uncertainty level (maximum variance) of the metamodel leads to an improvement of the metamodels global accuracy.

Hypothesis 2: Sampling at the site presenting the smallest metamodel prediction value (minimize prediction) for the objective function leads to an improvement of the best known value for the objective function, while locally improving the metamodels accuracy.

Considering the above-mentioned hypothesis, one can imagine that starting the optimization process with a global search and turning to a local search during iterations will improve the metamodels global accuracy at the beginning, when a few sites were sampled and thus the metamodel has poor prediction accuracy, and will end up refining the global optimal solution.

Cooling strategy

Based on the *GEI* criterion, by analogy with the Simulated Annealing algorithm, Sasena proposed a "cooling schedule" [SAS 02b] for his "cool criterion", by starting with a large value for the g parameter, which is reduced throughout iterations up to zero at the end of the optimization process. Hence, at the beginning of the optimization process, when few points are sampled and the metamodels accuracy is poor, the focus is set on the metamodels improvement, while towards the end of the optimization, the metamodels dispose of an increased accuracy, the metamodels exploitation is favored.

Switching strategy

In the same work [SAS 02b] with the previous strategy, a "switching criterion" has been proposed. This time, the balance between metamodels exploitation and exploration is controlled by alternating between the two extreme phases.

The proposed strategy starts by globally improving the metamodels prediction using the "maximum variance" criterion for a given number of k iterations ($k = 5$), switching then to a local search using the "minimize prediction" criterion until a local optimum is sought (three consecutive points with a maximum distance of 0.1% of each other in the design space). The criterion is then replaced by the "maximum variance" and the process is repeated until the stopping criterion is met.

Weighting strategy

Based on the expression of the expected improvement criterion, Sóbester et al. introduce a new strategy for balancing between the two phases of exploration and exploitation [SOB 05]. The new strategy is based on a criterion called "weighted expected improvement" (*WEI*) which is formulated as a linear combination of the two terms of the expected improvement criterion. The mathematical expression of the *WEI* criterion is given in (2.15).

$$WEI = WEI(\mathbf{x}) = \begin{cases} w(f_{min} - \hat{y})\Phi\left(\frac{f_{min} - \hat{y}}{\hat{s}}\right) + (1-w)\hat{s}\phi\left(\frac{f_{min} - \hat{y}}{\hat{s}}\right) & \text{if } \hat{s} > 0 \\ 0 & \text{if } \hat{s} = 0 \end{cases} \tag{2.15}$$

with w being the weighting factor, $w \in [0,1]$.

Considering the two extreme values of w, one can find the two extreme figure-cases, of exploration only, respectively exploitation only. Therefore, for $w = 0$, the *WEI* criterion will locate global improving sample points thus exploring the metamodels, while using $w = 1$ for the search

with the *WEI* criterion will yield a local search, thus the pure metamodel exploitation phase. The value of $w = 0.5$ for the *WEI* yields exactly the expected improvement criterion (0.5*EI*).

The strategy proposed along with the *WEI* criterion is similar to the one proposed in [GUT 01] and consists in cycling through the values of $w = \{0.1,0.3,0.5,0.7,0.9\}$. Thus, the search starts with a global design space exploration, moving towards metamodel exploitation. This pattern is repeated along the algorithm till the stopping criterion is met.

Adaptive and metamodel based weighting strategies

Two more complex strategies based on the weighted expected improvement were proposed recently in [XIA 11]. The first one entitled "adaptive weighted expected improvement" (*AWEI*) uses a reinforcement learning inspired approach, based on a system of "rewards". Its expression is given in (2.16) below.

$$AWEI = AWEI(\mathbf{x}) = \begin{cases} w_1(f_{min} - \hat{y})\Phi\left(\frac{f_{min} - \hat{y}}{\hat{s}}\right) + w_2\hat{s}\phi\left(\frac{f_{min} - \hat{y}}{\hat{s}}\right) & \text{if } \hat{s} > 0 \\ 0 & \text{if } \hat{s} = 0 \end{cases} \tag{2.16}$$

Instead of using one weighting parameter w of given levels, as in paragraph 0, two independent weighting coefficients (w_1, w_2) are calculated automatically at each iteration using a system of numerical rewards. Two potential rewards are calculated based on the average value of the mean squared error (MSE) value of all predicted points, representing the potential amount of reward resulting from each of the two possible actions, exploration and exploitation, respectively. The determined rewards are then used to update the values of the two weighting coefficients, w_1 and w_2. Once calculated, the weighting coefficients are then used to calculate the *AWEI* criterion using (2.16).

The second approach, the "surrogate model based weighted expected improvement" (*SMWEI*) proposes to improve the action selection on a long term basis (accounting for the future iterations of the algorithm), by predicting the cumulative rewards that are likely to occur as a particular action selection. For this purpose, at each iteration a metamodel is used in parallel to the *AWEI* strategy using the calculated pair of weighting coefficients in order to search for the global optimum of the problem.

These two strategies seem however burdensome and difficult to implement. Moreover, the *AWEI* strategy is not completely adaptive, since it still requires the tuning of an internal parameter.

2.5.3 Constraint handling

An important issue with the EGO algorithm is represented by the constraint handling [SAS 02]. Initially, the EGO algorithm was conceived to address unconstrained optimization problems [SAS 01], and was later adapted to handle also constrained problems.

As we have previously seen in paragraph 2.4.3, the constraint functions come in two flavors: the ones calculated based on the outputs of the fine model ("expensive constraints"), and the constraints calculated using only the design variables, through simple analytical expressions ("inexpensive constraints" or "cheap constraints"). The expensive constraints require the run of the simulation or fine model in order to compute their value, exactly as for the case of the objective function. Thus, the expensive constraints can be handled in a similar way to the objective function.

The cheap constraints are mostly geometric or topological constraints, which are imposed in order to avoid aberrant designs (i.e. designs that are not physically realizable, with certain elements that would overlap on each other). The value for these constraints is thus known without any (or reduced) computational effort. Given their different nature, the two types of constraints can be handled differently within the optimization process. Several approaches for handling both the expensive and the cheap constraints have been proposed in the literature, each with their own advantages and drawbacks.

The inexpensive constraints, as they are easy to determine, without any (or very reduced) computational effort, are handled in most cases directly by the optimizer that searches to maximize one of the infill sampling criteria presented in paragraph 2.5.1.

However, the expensive constraints, as they are not "*a priori*" known, require special attention. The most employed means of handling this type of constraints are either through penalties or by calculating the probability of feasibility, presented herein.

The general expression of the constrained optimization problem is presented in (2.17).

$$\begin{aligned} &\underset{\mathbf{x}}{\text{Minimize}} \quad f(\mathbf{x}) \\ &\text{subject to} \quad g_{inexp_i}(\mathbf{x}) \leq 0,\ i = 1,2,\cdots,n_g^{inexp} \\ &\qquad\qquad g_{exp_j}(\mathbf{x}) \leq 0,\ j = 1,2,\cdots,n_g^{exp} \\ &\qquad\qquad h_{inexp_k}(\mathbf{x}) = 0,\ k = 1,2,\cdots,n_h^{inexp} \\ &\qquad\qquad h_{exp_l}(\mathbf{x}) = 0,\ l = 1,2,\cdots,n_h^{exp} \end{aligned} \tag{2.17}$$

where $g_{inexp_i}(\mathbf{x})$ represent the inexpensive inequality constraints, $g_{exp_j}(\mathbf{x})$ are the expensive inequality constraints, $h_{inexp_k}(\mathbf{x})$ are the inexpensive equality constraints and $h_{exp_l}(\mathbf{x})$ are the expensive equality constraints.

The expensive equality constraints are very problematic, so their handling is difficult. To overcome this problem, these constraints are mostly transformed into two separate inequality constraints, $\left|h_{exp_k}\right| \leq \epsilon_k$, using a tolerance level ϵ_k, as in [HAW 08]. The value of ϵ_k is arbitrarily chosen as a small percentage (5%) of the range between the minimum and maximum values of the k-th expensive equality function. The same transformation can be done to handle the inexpensive equality constraint functions.

An analysis of a number of constraint handling techniques, specific to the EGO algorithm, with application to different analytical test problems can be found in [SCH 97]. In the following paragraphs, the most representative EGO constraint handling techniques existing in the literature will be presented.

Penalty function

A basic way of dealing with constraints is by penalizing the infill criterion whenever the expensive constraint functions are not respected [SAS 00]. A large negative constant is added to the expected improvement in order to restrain it from choosing points from infeasible areas of the design space. This can equally be done for the inexpensive constraint functions.

Considering for example the expected improvement criterion case, in order to account for constraints, the following modifications impose:

$$EIP = \begin{cases} E[I(\mathbf{x})] + Penalty & \text{if } g_{inexp}(\mathbf{x}) > 0 \text{ or } \hat{g}_{exp}(\mathbf{x}) > 0 \\ E[I(\mathbf{x})] & \text{if } g_{inexp}(\mathbf{x}) \leq 0 \text{ and } \hat{g}_{exp}(\mathbf{x}) \leq 0 \end{cases} \tag{2.18}$$

where $Penalty$ is a large negative[12] constant, *EIP* is the expected improvement with penalty.

However, an inherent problem with this approach consists in the selection of an appropriate value for the penalty applied to the infill criterion. To overcome this drawback, a simplified version of this procedure was presented by Sóbester in [SOB 04]. Instead of adding a penalty to the expected improvement criterion, each time the inexpensive or predicted expensive constraints are violated, the *EI* is simply set to zero. As the *EI* criterion is always positive or null, it is sufficient to set the value of *EI* to zero (i.e. there is no improvement).

$$EIP = E[I(\mathbf{x})] = \begin{cases} (f_{min}^* - \hat{y})\Phi\left(\frac{f_{min}^* - \hat{y}}{\hat{s}}\right) + \hat{s}\phi\left(\frac{f_{min}^* - \hat{y}}{\hat{s}}\right) & \text{if } \left(\hat{s} > 0 \text{ and } g_{inexp}(\mathbf{x}) \leq 0 \text{ and } \hat{g}_{exp}(\mathbf{x}) \leq 0\right) \\ 0 & \text{if } \left(\hat{s} = 0 \text{ or } g_{inexp}(\mathbf{x}) > 0 \text{ or } \hat{g}_{exp}(\mathbf{x}) > 0\right) \end{cases} \tag{2.19}$$

When the prediction error $\hat{s}$ of the objective function is null (i.e. already visited point, thus no improvement), or at least one of the expensive or inexpensive constraint functions is violated, the *EIP* criterion is set to zero. Otherwise, the *EIP* criterion is calculated using the expected improvement expression. For this modified version of the expected improvement, f_{min}^* should be the best feasible known value of the objective function. The importance of selecting the f_{min}^* instead of the f_{min} (without regard to the feasibility) was proven using a one-dimensional demonstrative example in [SOB 04].

The penalty method can be applied with all the other infill criteria earlier presented.

Probability of feasibility

The probability of feasibility method of handling constraints was proposed by Schonlau in [SCH 97]. The method consists in multiplying the expected improvement criterion by the probability of each constraint function being feasible.

Let's consider the presence of *k* expensive constraint functions within an optimization problem, $g_{exp_1}(\mathbf{x}), g_{exp_2}(\mathbf{x}), \cdots, g_{exp_k}(\mathbf{x})$, with $g_{exp_i}(\mathbf{x}) \leq 0$, for $i = 1,2,\cdots,k$.

By analogy with the probability of improvement (*PI*), the probability of feasibility (*PF*) represents the probability that the prediction will be greater or less than a constraint limit (i.e. the constraint is satisfied) when sampling in a particular point. The probability of feasibility therefore identifies feasible regions of the design space. The expression of the probability of feasibility for a constraint function is given in (2.20).

$$PF = P[F(\mathbf{x})] = \Phi\left(\frac{(G(\mathbf{x}) - g_{min}) - \hat{g}_{exp}}{\hat{s}}\right) = \frac{1}{\hat{s}\sqrt{2\pi}}\int_0^{\infty} e^{-\left[(G(\mathbf{x}) - g_{min}) - \hat{g}_{exp}\right]^2/(2\hat{s}^2)} dG \tag{2.20}$$

[12] Since the problem consists of maximizing the infill criterion (e.g. expected improvement criterion), penalizing the criterion consists of subtracting a large constant from its value, thus the negative penalty.

where $\hat{g}_{exp}$ is the metamodel prediction of the expensive constraint function, g_{min} is the limit value of the constraint, $G(\mathbf{x}) - g_{min}$ is the measure of feasibility, and $\hat{s}$ is the variance predicted by the Kriging model of the constraint function.

Then, the expected improvement with the probability of feasibility constraint handling becomes:

$$E[I(\mathbf{x}) \cap F(\mathbf{x})] = E[I(\mathbf{x})]P[F(\mathbf{x}) > g_{min}] \tag{2.21}$$

This method is referred to by Forrester et al. [FOR 08] as the "constrained expected improvement" (*CEI*). The probability of feasibility is calculated for each expensive constraint function. The *CEI* is obtained by multiplying the expected improvement by the probability of feasibility of each constraint, as in (2.22).

$$CEI = EI \cdot PF_1 \cdot PF_2 \cdot \cdots \cdot PF_n = E[I(\mathbf{x})]P[F_1(\mathbf{x})]P[F_2(\mathbf{x})] \cdots P[F_n(\mathbf{x})] \tag{2.22}$$

By contrast with the penalty method, this method gradually drives the infill criterion to zero in the zone of transition between feasible and unfeasible regions, thus smoothing the landscape of the infill criterion in this area [PAR 10].

In a similar manner, the probability of feasibility can be used with other infill criteria, such as the probability of improvement or the generalized expected improvement.

Expected violation

Another method of handling constraints within the EGO algorithm was proposed in [AUD 00]. Similar to the expected improvement criterion, Audet et al. proposed the quantification of the amount of constraint overpassing, which they term as "expected violation" (*EV*). The mathematical expression of the expected violation criterion is given in (2.23).

$$EV = E[V(\mathbf{x})] = \begin{cases} (\hat{g}_{exp} - 0)\Phi\left(\frac{\hat{g}_{exp} - 0}{\hat{s}}\right) + \hat{s}\phi\left(\frac{\hat{g}_{exp} - 0}{\hat{s}}\right) & \text{if } \hat{s} > 0 \\ 0 & \text{if } \hat{s} = 0 \end{cases} \tag{2.23}$$

The *EV* has a greater value in regions where the constraint is likely to be violated or the metamodels accuracy is poor and a lower value elsewhere.

In their implementation, the search for feasible infill points is done in two steps. First, a feasible set of solutions is sought. Secondly, among all the feasible solutions, the ones having the greatest expected improvement are selected as infill points.

Instead of maximizing or minimizing an infill criterion, they apply an enumeration method by using a very large Latin Hypercube (10000, 100000 points) to sample the design space in the search for feasible solutions. For each point of the Latin Hypercube, the *EV* is calculated for all expensive constraint functions. The points having the smallest maximum expected violation among all constraints are then selected. For these possibly feasible points the expected improvement is calculated and the point with the greatest expected improvement is selected (in the original proposition, the first 5 points are selected).

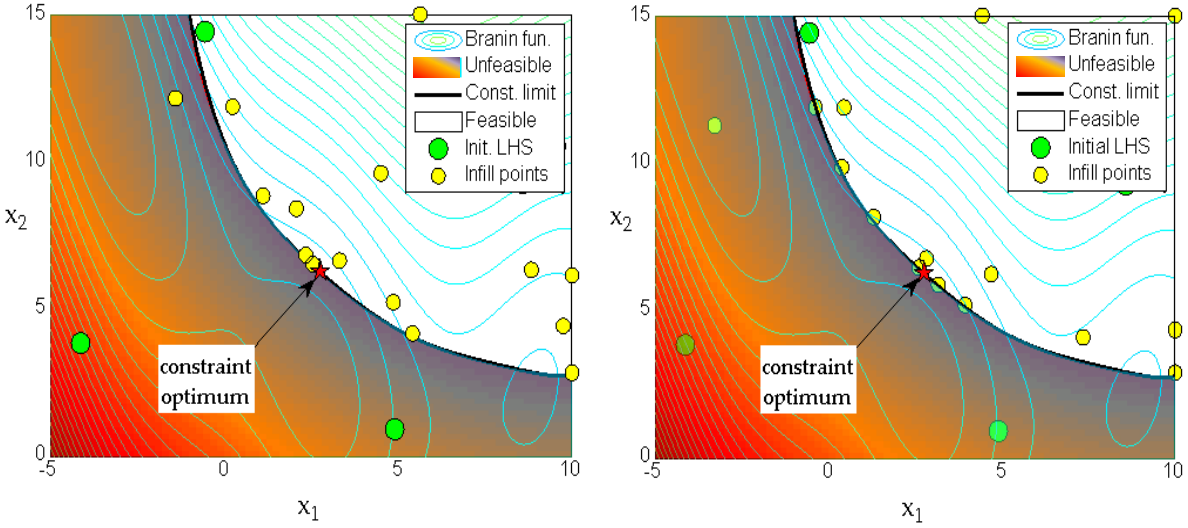
Constraint handling discussion

The constraint handling methods previously presented are analyzed here with regard to the modified Branin-Hoo function first introduced in paragraph 2.3. For this, an analytical constraint

function is considered and the expression of the constraint modified Branin-Hoo problem is given in (2.24).

$$\begin{aligned} \underset{x_1,x_2}{\text{Minimize}} \quad & f(x_1,x_2) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left[\left(1-\frac{1}{8\pi}\right)\cos x_1 + 1\right] + 5x_1 \\ \text{such as} \quad & g(x_1,x_2) = 2(18-x_1)^2 + 3(15-x_2)^2 - 40x_1 - 47x_2 - 100 \\ \text{with} \quad & x_1 \in [-5,10],\ x_2 \in [0,15] \end{aligned} \tag{2.24}$$

Two of the constraint methods previously presented, the Penalty function respectively the *PF* were considered for finding the optimum of the constrained modified Branin-Hoo function. For each optimization run, an initial experimental design of 4 points chosen using a LHS strategy was imposed. A number of 16 infill points were added by the EGO algorithm. The results are presented graphically in Figure 2.21.



a) Penalty function b) Probability of Feasibility (*PF*)

Figure 2.21 : Performance comparison of EGO algorithm with two constraint handling methods

Both optimization runs did not encounter any difficulty in finding the site of the constrained optimum of the problem, having a similar performance. From the results presented in Figure 2.21, it can be seen that most of the infill points added by the EGO algorithm were placed inside the feasible area of the design space, defined by the constraint function. However, some of the points were equally sampled into the infeasible region, as a result of the metamodel improvement feature of the *EI* infill criterion used with the EGO algorithm. Moreover, it can be remarked that most of the points sampled by the algorithm using the *PF* constraint handling method, Figure 2.21b, were displaced mainly on one sides and the other of the feasible domain boundary. In compare, the points sampled by the algorithm using the Penalty function for handling the constraint, Figure 2.21a, were placed into the feasible domain. This could be explained by the quadratic pattern of the constraint function, which was relatively simple to predict with only a few support points, and thus correctly penalize the infill criterion.

For constrained optimization problems, both methods can be applied with the EGO algorithm. The Penalty function seems more appropriate for cases where the constraint functions are relatively simple. For highly nonlinear constraint functions, where a global accurate metamodel of the constraint functions is unlikely to be obtained, the *PF* method is more appropriate. Both constraint methods were implemented in the EGO algorithm developed.

2.5.4 Application: single-objective SMES device optimization

To analyze the behavior of the EGO algorithm and to assess its performance, a classical electromagnetic test-problem from the electromagnetic community is considered. The device to be optimally sized is a superconducting magnetic energy storage system (SMES), and the optimization benchmark is known to the international electromagnetic community as the *TEAM22* problem [MAG 08]. The SMES device description along with the *TEAM22* optimization problem formulation and the optimization process details are presented towards the end of the chapter, in paragraph 2.6.

2.5.5 EGO algorithm with multiple objectives (MEGO algorithm)

So far, the focus in this work was set on single-objective optimization problems. In practice, as mentioned earlier in paragraph 2.4.2, most of the real engineering problems have several, often conflicting, optimization criteria which must be accounted for within the optimal design process. Optimization criteria such as mass, efficiency, environmental impact, production cost etc. are most often present in the formulation of electrical devices or systems optimization. Until one-two decades ago, when most of the optimization algorithms were single-objective, the designers used to group the different goals of the optimization problem within one single optimization function, by the means of some suitable weighting function which combined the goals of interest. However, expressing all the goals of a design problem under the form of a single-objective function might not always be obvious, nor desired, sometimes. Instead of a single optimal design, corresponding to "*a priori*" fixed combination of weighting coefficients, the designer may wish for an optimal set of compromise designs, for him or her to make his or her final choice from. For this, a multi-objective optimization algorithm is required. Consequently, the number of simulation model evaluations required by a multi-objective optimization algorithm to construct a decent Pareto front is much more substantial than for the single-objective optimization case. In the following paragraphs, the attention is turned to the case of optimization problems presenting multiple objectives and the way this can be accounted for by the metamodel-based design optimization algorithm EGO presented thus far, which is meant to reduce the number of calls to the expensive simulation model.

2.5.5.1 Existing multi-objective extensions of the EGO algorithm

A few attempts have been made recently to extend the principles of the single-objective EGO algorithm to the determination of not one solution (global optimum) of a problem, but instead, a set of non-dominated tradeoff solutions forming the Pareto front of a multi-objective problem. Most of these multi-objective extensions are based on the EGO algorithm with the *EI* as infill point sampling criterion. Here, there were selected for presentation the most representative multi-objective propositions.

Multi-objective *EI*

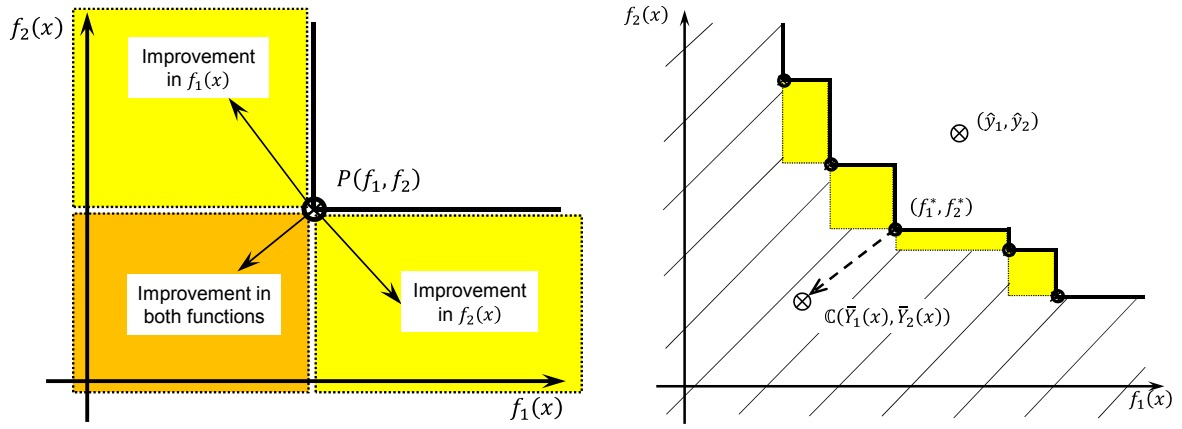
A multi-objective derivation of the expected improvement criterion has been proposed by Keane in [KEA 06]. The multi-objective expected improvement was derived for the bi-objective case. First, a bi-dimensional probability density function was calculated using the expression (2.25).

$$\phi(Y_1, Y_2) = \frac{1}{\hat{s}_1\sqrt{2\pi}} e^{-(Y_1-\hat{y}_1)^2/(2\hat{s}_1^2)} \cdot \frac{1}{\hat{s}_1\sqrt{2\pi}} e^{-(Y_2-\hat{y}_2)^2/(2\hat{s}_2^2)} \tag{2.25}$$

Instead of calculating the probability of improving each objective function individually, the probability of improving the current Pareto front is calculated.

A new point can represent an improvement point in two ways: either by improving one of the two objective functions or by improving both objective functions, as presented in Figure 2.22a.

The improvement measure is accounted through the probability of augmenting the current Pareto front. A slightly different derivation, by considering only the probability of dominating at least one non-dominated solution was developed by Forrester and Keane in [FOR 09]. The latter approach was chosen for presentation herein.



a) Possible Pareto front improvements b) Centroid of *EI* and current Pareto front

Figure 2.22 : Bi-objective expected improvement calculation

The shaded area in Figure 2.22b represents those designs that augment the Pareto front and the hatched area represents the designs that dominate one or several points of the current Pareto front.

By analogy with the single-objective case, the probability of improving the current Pareto front is obtained by integrating the volume under the joint probability density function. The probability of improving the current Pareto front is thus given by the expression (2.29).

$$P[I(\mathbf{x})] = P\left[Y_1(\mathbf{x}) < y_1^{(i)} \cap Y_2(\mathbf{x}) < y_2^{(i)}\right] = \int V_{hatch} \tag{2.26}$$

where V_{hatch} is the volume given by the hatched zone in Figure 2.22b.

In order to calculate the expected improvement, the position of its centroid is needed. Hence, the position of the centroid of *EI* is then calculated by integration with respect to the origin and division by $P[I(\mathbf{x})]$. The expression of the expected improvement is given in (2.27).

$$E[I(\mathbf{x})] = P[I(\mathbf{x})] \cdot d(\mathbb{C}, \wp) = P[I(\mathbf{x})] \sqrt{\left(\bar{Y}_1(\mathbf{x}) - y_1^*(\mathbf{x}^*)\right)^2 + \left(\bar{Y}_2(\mathbf{x}) - y_2^*(\mathbf{x}^*)\right)^2} \tag{2.27}$$

where $\mathbb{C}$ represents the centroid of *EI*, located at $\left(\bar{Y}_1(\mathbf{x}), \bar{Y}_2(\mathbf{x})\right)$ and $\wp$ is the current Pareto front, $y_1^*(\mathbf{x}^*)$ and $y_2^*(\mathbf{x}^*)$ are the coordinates of the point from the Pareto set, closest to the tested point x.

However, the decomposition of the area under the Pareto front and the calculation of its integral, as well as the determination of the centroid of *EI* is a computationally demanding task. Moreover, the generalization of the concept for cases of more than two objective functions is not at all a trivial task.

Enhanced multi-objective *PI*

An enhancement of the multi-objective probability of improvement previously presented was proposed by Hawe and Sykulsky in [HAW 07a]. Their proposition is meant to yield a more global search by acquiring larger improvements.

At a given iteration, depending on the number of points forming the current Pareto front N_{par} , the hatched area in Figure 2.22b, over which the probability of improvement is calculated, is divided into several smaller areas. Each of these areas are then associated a level of improvement, depending on the number of points from the Pareto front that are dominated by a design belonging to a certain area. This division into levels of improvement is graphically presented in Figure 2.23.

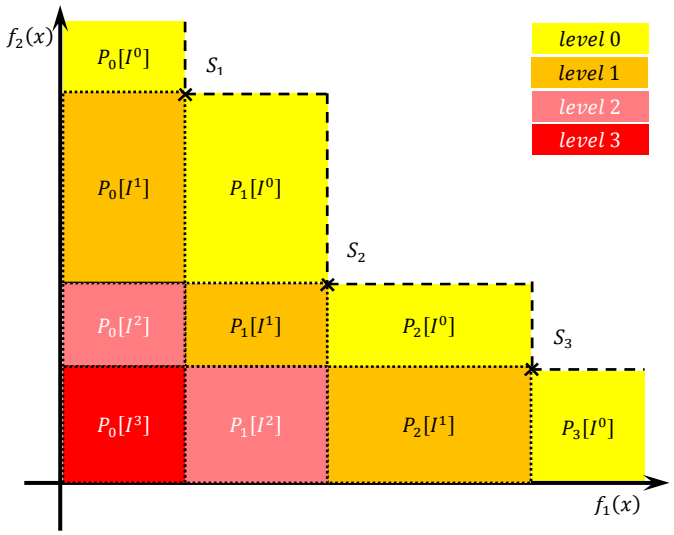


Figure 2.23 : Partitioning of the *PI* in n=5 levels of improvement

where $P[I^k(\mathbf{x})]$ represents the probability that the tested point $\mathbf{x}$ yields a level of improvement k, and $P_i[I^k(\mathbf{x})]$ represents the probability that the tested point $\mathbf{x}$ will dominate the k non-dominated solutions of the Pareto front $S_{i+1}, S_{i+2}, \cdots, S_{i+k}$.

The enhanced probability of improvement can thus be expressed as in (2.28).

$$P[I^k(\mathbf{x})] = \sum_{i=0}^{N_{par}} P_i[I^k(\mathbf{x})] \tag{2.28}$$

Even though the authors argue that the technique is extensible to optimization problems implying more than two objective functions, this exercise is less obvious and a numerical burden is required. This enhancement was introduced as a concept; no application was provided in support to its worthiness.

ParEGO algorithm

A more straightforward attempt to transform EGO in order to account for the multiple criteria of a multi-objective problem is represented by Knowles Pareto EGO algorithm (ParEGO) [KNO 05a]. In order to determine the Pareto frontier of a multi-objective optimization problem, Knowles proposes applying a transformation technique, prior to the infill search internal optimization loop of the EGO algorithm. Hence, the multi-objective optimization problem is transformed into a single-objective problem using a weighting method.

The algorithm starts by drawing an initial experimental design, which is evaluated using the fine model. A set of evenly distributed weighting vectors Λ is generated, depending on the dimension of the objective space, i.e. number of objective functions. The set of weighting vectors is generated prior to the optimization process using the expression in (2.29).

$$\Lambda = \left\{ \lambda = (\lambda_1, \lambda_2, \cdots, \lambda_k) \middle| \sum_{j=1}^{k} \lambda_j = 1 \Lambda \forall j, \lambda_j = \frac{l}{s}, l \in \{0, \cdots, s\} \right\} \tag{2.29}$$

where $|\Lambda| = \binom{s+k-1}{k-1}$, s is a constant determining the total number of weighting vectors of Λ, λ_j is a weighting vector whose elements are evenly distributed so that its sum is equal to the unit.

A vector of weighting coefficients λ is then selected at random among the initially generated set of evenly distributed weighting vectors Λ. Using the selected λ weighting coefficients vector, the normalized objective function values of the initial experimental design are then aggregated within a single-objective measure, by applying the augmented Tchebyshev scalarization formulation [NAK 09] presented in the first chapter of the manuscript. The mathematical expression of the augmented Tchebyshev function is given in (2.30).

$$f_\lambda(\mathbf{x}) = \max_{j=1:k} \left(\lambda_j \cdot f_j(\mathbf{x}) \right) + \rho \sum_{j=1}^{k} \lambda_j \cdot f_j(\mathbf{x})$$
$$\text{with} \quad \bar{f}_j(\mathbf{x}) = \frac{f_j(\mathbf{x}) - f_j^{min}}{f_j^{max} - f_j^{min}} \tag{2.30}$$

where $f_\lambda(\mathbf{x})$ is the single-objective measure obtained using the augmented Tchebyshev formulation for the weighting vector λ, $f_j(\mathbf{x})$ is the value of the j-th objective function, f_j^{min} and f_j^{max} are the minimum, respectively the maximum known or supposed values of the j-th objective function, $\bar{f}_j(\mathbf{x})$ is the normalized value of the j-th objective function to the range [0,1], and ρ is a constant with a small positive value ($\rho = 0.05$ for their implementation).

The infill search problem is reduced then to the maximization of the expected improvement of expression (2.13). Other single-objective infill criteria might be also used. At each iteration of the algorithm, the infill criterion (*EI* for their implementation) is calculated for the single-objective measure obtained by aggregation of the objective functions of the problem. An infill point is

obtained and evaluated using the fine model. The aggregation measure is then recalculated using the same vector of weighting coefficients λ. This weighting vector is maintained for a number of iterations (5 for their implementation). After that, the next weighting coefficient λ in the set of vectors Λ is selected, so that the weighting between the objective functions is modified regularly. The rotation of the different weighting vectors along iterations allows for progressively determining the Pareto front of the initial multi-objective optimization problem.

This weighting procedure is the simplest multi-objective extension of EGO. A comparison of the ParEGO algorithm to an evolutionary algorithm on a test suite of a number of analytical test functions with a limited budget of 250 function evaluations is presented in [KNO 05]. A more extensive comparison of ParEGO to similar and other algorithms, this time accounting for the noise in the objective functions is presented in [KNO 09]. To assess the impact of noise on the performance of the algorithms, different levels of numerical noise were artificially added to the expressions of the objective functions.

Multi-EGO algorithm

Another approach for extending the applicability of EGO to account for multiple criteria is represented by Jeong and Obayashi's Multi-EGO algorithm [JEO 05]. Instead of aggregating the infill criteria (expected improvement) of each objective function within one criterion, they propose using the *EI*'s of the objective functions directly in a multi-objective optimization. The formulation of the infill sampling optimization problem is presented in (2.31).

$$\begin{aligned} &\underset{\mathbf{x}}{\text{Minimize}} \quad \{f_1^*(\mathbf{x}), f_2^*(\mathbf{x}), \cdots, f_k^*(\mathbf{x})\} \\ &\text{with} \quad f_i^*(\mathbf{x}) = EI\left(\mathbf{x}, f_i, f_i^{best}\right), \qquad i \in \{1, k\} \end{aligned} \tag{2.31}$$

where k is the number of objective functions of the multi-objective problem, $EI(\mathbf{x}, f_i)$ is the expected improvement value of i-th objective function for the design vector x, with respect to the current best known value of the i-th objective function, f_i^{best}.

A multi-objective evolutionary algorithm (MOEA) is used to determine the Pareto front of the expected improvements of all objective functions. MOEA's are population-based algorithms, therefore the subsequent Pareto front is composed of a large set of points. From the *EI* Pareto front, a reduced set of k solutions having the best *EI* value for each objective function is selected. Additionally, for a better convergence of the Pareto front, the solution located closest to the center of the n-dimensional *EI* Pareto front is also selected. The selected $k+1$ infill points are then evaluated using the fine model. Their algorithm was applied with success to an aerodynamic shape optimization problem [JEO 05]. A slightly modified version of this algorithm, including a k-means clustering of the *EI* Pareto designs in order to limit the number of infill designs, was successfully applied to the multi-objective (4 objective functions) optimization of the combustion chamber of a Diesel engine [JEO 06].

Scalarizing one-stage algorithm

Another algorithm, which uses a transformation technique to turn the multi-objective problem formulation into a single-objective problem, was presented by Hawe and Sykulski in [HAW 08].

The objective functions of the multi-objective problem are normalized with respect to their known minimum and maximum values, so that they lie in the interval [0,1], according to the expression given in (2.32).

$$\bar{f}_i(\mathbf{x}) = \frac{f_i(\mathbf{x}) - f_i^{min}}{f_i^{max} - f_i^{min}} \tag{2.32}$$

where $\bar{f}_i(\mathbf{x})$ is the i-th normalized objective function, f_i^{min} and f_i^{max} represent the current minimum, respectively the maximum values of the i-th objective function.

Then, the augmented weighted Tchebyshev function presented in (2.30) is used to aggregate all objectives within a unique variable. The same way as for the ParEGO algorithm, a normalized weight vector is used, and at each iteration, a different weight vector is chosen at random from the list of initially generated weight vectors.

Once the multi-objective problem is transformed into a single-objective problem, a conditional likelihood based single-objective infill criterion [JON 01] is maximized with respect to the expensive constraint functions.

2.5.5.2 Pseudo-distance infill point selection criterion

The pseudo-distance (PD) infill criterion, initially proposed by Kreuawan in [KRE 08] was integrated within the MEGO (Multi-objective EGO) algorithm developed by Berbecea et al. [BER 10]. This infill point selection criterion is based on the definition of the non-domination concept which is found at the basis of the NSGA-II algorithm [DEB 02].

In order for a Pareto front to represent a good trade-off between the objective functions that define it, two conditions have to be fulfilled, which can be expressed as:

(i) The non-dominated points that compose the Pareto front should be situated as close as possible to the utopia point (point created using the minimum value of each objective function among all points of the Pareto front);

(ii) The members of the Pareto front should be as uniformly spaced as possible.

The two previously mentioned features of a proper Pareto front are found at the basis of the mechanisms for the advancement of the Pareto front of the pseudo-distance criterion.

The mathematical expression of the pseudo-distance infill criterion is given in (2.33).

$$PD = PD(\mathbf{x}) = D_d(\mathbf{x}) + D_n(\mathbf{x}) \tag{2.33}$$

$$D_d(\mathbf{x}) = \sum_{i=1}^{m} D_d^{f_i}(\mathbf{x}) = \sum_{i=1}^{m} \sum_{j=1}^{n_{dom}} \left(\left(\frac{f_i^{(s_j)} - \hat{f}_i(\mathbf{x})}{f_{i\_max} - f_{i\_min}} \right) \cdot \frac{1}{\hat{s}_i(\mathbf{x})} \right) \tag{2.34}$$

$$D_n(\mathbf{x}) = \sum_{i=1}^{m} D_n^{f_i}(\mathbf{x}) = \sum_{i=1}^{m} \left(\left(\frac{f_i^{(s_+)} - \hat{f}_i(\mathbf{x})}{f_{i\_max} - f_{i\_min}} \right) \cdot \hat{s}_i(\mathbf{x}) \right) \tag{2.35}$$

where m represents the number of design objectives, n_{dom} is the number of points on the current Pareto front, which are dominated by the trial point x, f_{i_max} and f_{i_min} are the known maximum, respectively minimum of the i-th objective function on the current Pareto front, $\hat{f}_i$ and $\hat{s}_i$ are the

Kriging metamodels prediction of the *i*-th objective function, respectively the estimated error associated to this prediction, $f_i^{(s_j)}$ is the value of the *i*-th objective function of the *j*-th point of the current Pareto front, which is dominated by the trial point $\mathbf{x}$, $f_i^{(s_+)}$ is the value of the *i*-th objective function of the point from the current Pareto front, closest to the trial point.

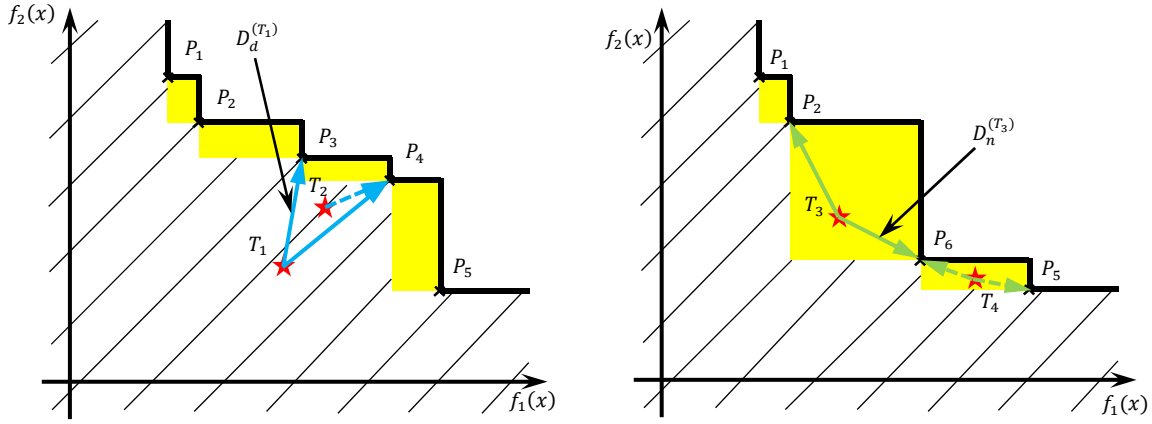
The infill sampling problem consists thus of maximizing the pseudo-distance infill criterion expressed in (2.33)-(2.35). The pseudo-distance infill criterion is composed of two terms. The first term, $D_d(\mathbf{x})$, expressed in (2.34) is called the dominance distance and it focuses on points that dominate one or more points of the current Pareto front. The second term, $D_n(\mathbf{x})$, expressed in (2.35), is named neighboring distance, addressing those points that do not dominate any of the existing points from the current Pareto front, but which would augment the size of the Pareto front.

In the expression of the pseudo-distance criterion, the uncertainty of the metamodels prediction is equally accounted for within the two terms of the criterion. When $\hat{s} \to 0$, the metamodels prediction present a high accuracy. In this case, the second term of the pseudo-distance, the neighboring distance, tends to zero and the first term, the dominance distance, becomes important. Hence, the maximization of the pseudo-distance infill criterion becomes equivalent to maximizing the dominance distance, $D_d(\mathbf{x})$. On the contrary, when $\hat{s} \to \infty$, the metamodels prediction for these points is highly inaccurate. In this case, the first term of the pseudo-distance criterion, the dominance distance, becomes negligible, while the second one, the neighboring distance, becomes dominant. For this figure-case, maximizing the pseudo-distance reduces to maximizing the neighboring distance, $D_n(\mathbf{x})$.

The dominance distance $D_d(\mathbf{x})$ searches among the points that dominate one or more points of the Pareto front, and which are predicted by the metamodel with a high accuracy. According to the dominance distance, the point that dominates one or more existing Pareto points with the greatest distance will be selected. This term is meant to address the first feature of a good Pareto front, expressed in (i) and is thus responsible of advancing the Pareto front towards the utopia point.

The neighboring distance, $D_n(\mathbf{x})$ addresses those points which have a less accurate prediction and are equivalent to the existing Pareto points (do not dominate any of the existing Pareto points). This distance searches thus for less accurate points which fill at best the gaps between the existing points of the Pareto front. This term responds hence to the second characteristic of a good Pareto front, (ii) by attempting to improve the spacing of the Pareto front.

To exemplify the purpose, let's consider a simple example of a bi-objective optimization problem. Suppose that at the *i*-th iteration of the optimization process using the pseudo-distance infill criterion, the Pareto front is composed of five points, marked P_1, P_2, P_3, P_4, P_5, as in Figure 2.24a. The two trial points $T_1$and T_2, belonging to the hatched area in Figure 2.24a, dominate one or more points of the current Pareto front. Between the two trail points, the one having the longest dominance distance to the current Pareto front will be selected, thus T_1, as its dominance distance is clearly larger than that of T_2. This point will next be evaluated using the fine model, yielding a new point on the Pareto front, P_6, and eliminating two existing non-dominated points, P_3 and P_4. At the next iteration of the optimization process, the new Pareto front is the one given in Figure 2.24b, consisting of points P_1, P_2, P_6, P_5. Considering two other trial points, T_3 and T_4, which in this case do not dominate any of the existing points of the Pareto front, as they belong to the shaded area in Figure 2.24b, the selection will be done with regard to their neighboring distance values. Hence, the point T_3 will be selected over T_4, due to its larger neighboring distance, $D_n^{(T_3)}$.



a) Iteration *i* b) Iteration *i*+1

Figure 2.24 : Example of optimization using the pseudo-distance infill criterion

2.5.6 *Multi-objective EGO algorithm (MEGO)*

A Multi-objective Efficient Global Optimization (MEGO) algorithm integrating the above-presented pseudo-distance multi-objective infill criterion has been developed [BER 10]. Figure 2.25 presents the flowchart of the MEGO algorithm. The MEGO algorithm follows the following main lines:

- Step 1: An initial sampling plan is selected using a design of experiments technique, such as the Latin Hypercube Sampling (LHS) technique and the designs are evaluated using the fine model (e.g. simulation model);
- Step 2: Kriging metamodels are fitted for each objective and constraint function of the optimization problem over the list of sampled data ($\mathcal{L}$);
- Step 3: An infill point is searched by maximizing the pseudo-distance multi-objective infill criterion (paragraph 2.5.5.2);
- Step 4: Evaluate the previously-determined infill point using the fine model;
- Step 5: Test the infill point for improvement by recalculating the current Pareto front and add the point to the set of Pareto solutions ($\wp$) if improvement is found; to accelerate the calculus, an external C code developed by Yi Cao[13] for extracting the Pareto front from a list of evaluated designs is used here;
- Step 6: Add the evaluated infill point to the list of sampled data ($\mathcal{L}$), augmenting the information used for fitting the Kriging metamodels;
- Step 7: Verify the stopping criterion (e.g. a pre-imposed number of iterations is attained); if the stopping criterion is met, then stop the process, otherwise continue with Step 2.

The workflow of the multi-objective EGO algorithm developed (MEGO) is given in Figure 2.25.

[13] The well-written C code wrapped into a Matlab function for extracting the Pareto front out of a list of evaluated designs developed by Yi Cao is available online on the Mathworks MatlabCentral site at the following address: http://www.mathworks.com/matlabcentral/fileexchange/17251-pareto-front

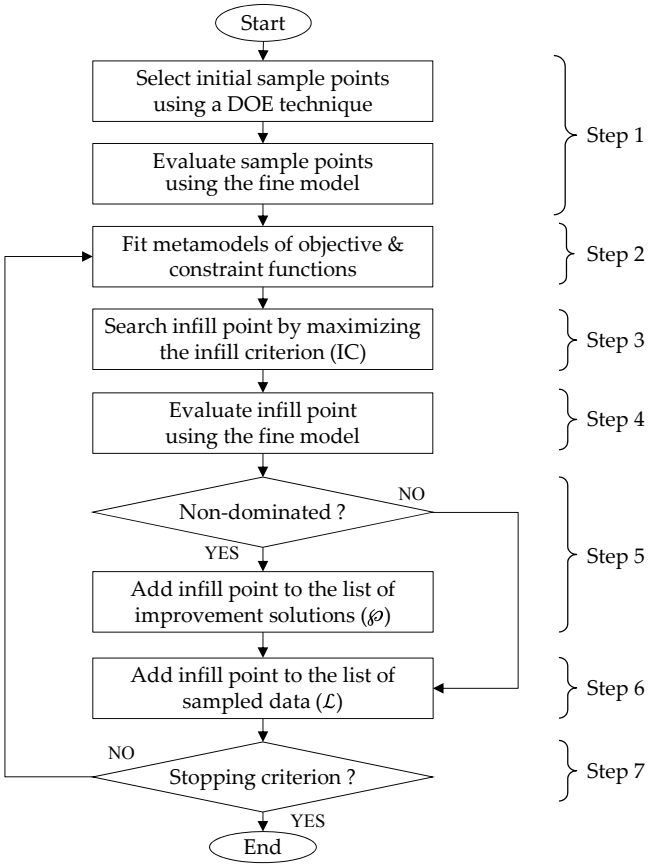


Figure 2.25 : Multi-objective Efficient Global Optimization (MEGO) algorithm flowchart

2.5.7 Distributed computation-suited MEGO

In its initial form, the MEGO algorithm, as all other EGO-based approaches have a sequential mechanism of adding new infill points. At a given iteration, one single infill point is selected for evaluation with the fine model. However, as the fine model is in most of the cases a simulation model (e.g. FE analysis), requiring an important computational burden, the idea of distributing the fine model computation on several different machines, or on different cores of the same machine brings out an important potential gain in computational time. For this to be possible, the algorithm should be able to generate at each iteration not just one design, but instead, a set of N_{core} designs. Therefore, a sum of structural modifications to the algorithm is required. Once generated, these designs are then evaluated simultaneously using the fine model, each on a different available core of the machine. The total fine model evaluation time of an iteration would be reduced to the slowest of the model evaluations. Ideally, the time gain would be directly proportional to the number of cores that perform the model evaluations for different configurations (e.g., the time needed for the concurrent evaluation of 4 model configurations on 4 different cores of a machine would be 4 times less than the time needed to perform the model evaluations sequentially, on a single core). In order to test the purpose, two different strategies for generating a set of N_{core} designs to allow

distributing the model evaluations are implemented and tested on a high-end server[14] which disposes of 2 CPUs with 4 cores each. The two strategies were successfully tested with the optimization of the SMES device of the *TEAM22* benchmark problem on different core configurations (from 1 up to 8 cores) of the server. The results of the SMES optimization can be found in [BER 10].

2.5.7.1 Hybrid model approach

The first method that has been implemented and tested was initially proposed by Schonlau [SCH 97] within a single-objective optimization context based on the expected improvement. The idea consists of using the objective function metamodel prediction in order to generate a set of N_{core} infill designs at each iteration k of the algorithm. This approach can be described using the following steps:

- Step 1: Initiate the list $\mathcal{L}'$ by $\mathcal{L}' = \mathcal{L}$ and set $i = 1$;
- Step 2: Fit Kriging metamodels over the list $\mathcal{L}'$;
- Step 3: Search for an infill point by maximizing the pseudo-distance infill criterion;
- Step 4: Evaluate the infill design using the Kriging metamodels;
- Step 5: Augment the list $\mathcal{L}'$ by adding the Kriging metamodels prediction of the previously determined infill point to $\mathcal{L}'$;
- Step 6: Verify if the number of N_{core} infill points has been reached; if yes, then evaluate the N_{core} infill designs by the fine model using all available cores of the machine; otherwise, continue with Step 2.

The above detailed steps can be graphically represented using the workflow in Figure 2.26.

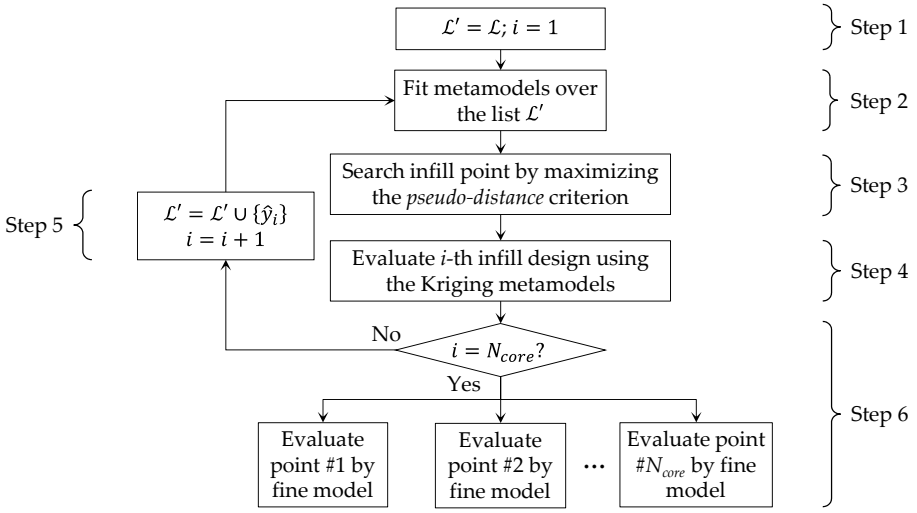


Figure 2.26 : Workflow for generating N_{core} infill points using the "hybrid model" approach

[14] The server's configuration consists of an Intel® Xeon® X5470 at 3.33GHz, dual CPU, on 64-bit, 4cores/CPU, with 32GB of RAM, turning on a Windows Server 2007 64-bit platform with SP2.

This approach was named here "hybrid model approach", with regard to the hybrid evaluation of designs (i.e. the list of sampled points ($\mathcal{L}$) contains a mix of both fine model evaluated designs and Kriging metamodel predictions).

The infill point generation in this case remains sequential. However, this does not have a strong impact on the computational time since the infill point search phase consumes a negligible amount of time compared to the infill point evaluation with the fine model.

2.5.7.2 Weighted pseudo-distance approach

The second implemented and tested approach for generating the N_{core} designs to be evaluated concurrently using the fine model is inspired in part by the technique used by Sobester with the weighted expected improvement (see paragraph 2.5.2), and in part by Knowles' approach for extending EGO to handle multiple objectives (see paragraph 2.5.5.1). In Sobester's approach, the weighting coefficients which control the balance between metamodel exploration and exploitation of the expected improvement are used in order to generate the desired number of infill points. Here, it is proposed to introduce a controlled weighting in the expression of the pseudo-distance multi-objective infill criterion, which would balance between the different objective functions of the multi-objective optimization problem. The variable weighting between the objective functions of the multi-objective problem is meant to progressively determine the entire Pareto front. The expression of the weighted pseudo-distance (WPD) criterion is given in (2.36)-(2.38).

$$WPD(\mathbf{x}, w) = D_d(\mathbf{x}, w) + D_n(\mathbf{x}, w) = \sum_{i=1}^{m} w_i D_d^{f_i}(\mathbf{x}) + \sum_{i=1}^{m} w_i D_n^{f_i}(\mathbf{x}) \tag{2.36}$$

$$D_d(\mathbf{x}, w) = \sum_{i=1}^{m} w_i D_d^{f_i}(\mathbf{x}) = w_1 D_d^{f_1}(\mathbf{x}) + w_2 D_d^{f_2}(\mathbf{x}) + \cdots + w_m D_d^{f_m}(\mathbf{x}) \tag{2.37}$$

$$D_n(\mathbf{x}, w) = \sum_{i=1}^{m} w_i D_n^{f_i}(\mathbf{x}) = w_1 D_n^{f_1}(\mathbf{x}) + w_2 D_n^{f_2}(\mathbf{x}) + \cdots + w_m D_n^{f_m}(\mathbf{x}) \tag{2.38}$$

Depending on the number of cores available for distributing the calculi, a list of uniformly spaced weight vectors W is generated at the beginning of the optimization process, using the procedure proposed by Knowles for his ParEGO algorithm (paragraph 2.5.5.1). The procedure for generating N_{core} infill points using the weighted pseudo-distance approach can be described using the following steps:

- Step 1: Select a weight vector w_i at random among the list of all possible combinations of weights, W, making sure that no vector is selected twice;
- Step 2: Determine a new candidate location by maximizing the weighted pseudo-distance criterion for the previously selected combination of weights w_i, $\mathrm{Max}_{\mathbf{x}}\, WPD(\mathbf{x}, w_i)$;
- Step 3: Verify if the desired number of infill designs is attained; if this number is reached, then launch the fine model evaluations of all infill points, otherwise increment i and continue with Step 1.

The workflow sequence that generates N_{core} infill designs within the MEGO algorithm is presented in Figure 2.27.

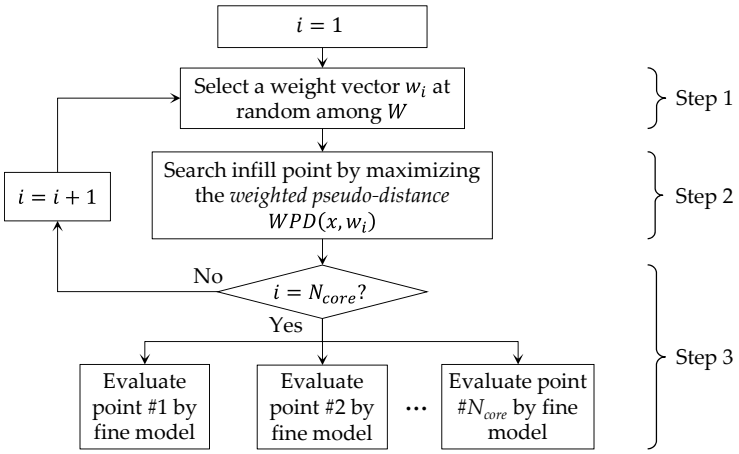


Figure 2.27 : Workflow for generating N_{core} points using the "weighted pseudo-distance" approach

The infill search phase using different weight vectors can be equally distributed, but the computational time gain of this phase would however rest negligible compared to the fine model evaluation phase of the algorithm.

The two approaches which generate a set of infill points for distributing the calculi over the number of available cores of the machine have been implemented with the developed MEGO algorithm. The workflow of the MEGO algorithm integrating these two strategies for generating N_{core} infill points is presented in Figure 2.28. Different time trackers ($t_0 \div t_5$) have been placed in Figure 2.28 in order to track down the elapsed time of all major steps of the algorithm and will be addressed later, in paragraph 2.6.4.3, which addresses the speedup for the optimization of an electromagnetic device.

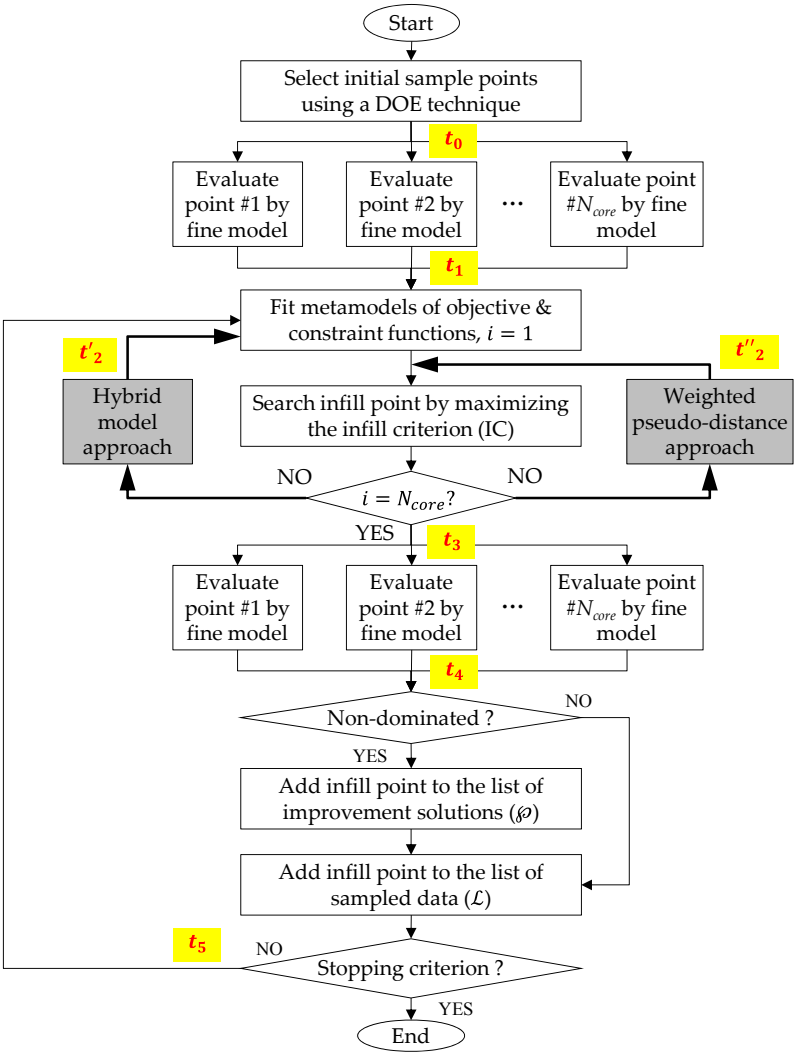


Figure 2.28 : MEGO algorithm workflow implementing the two strategies for distributing the calculi over several cores of one or more machines

Another phase of the algorithm that is equally subjected to distributed calculation of the fine model is represented by the initial sampling plan evaluation, as can be seen from Figure 2.28. At this step of the algorithm, the distribution of the fine model calculation can be done without any additional complexity.

To reduce the total optimization time and to fully benefit from the computational power at hand represented by the eight cores of the high-end server, the two strategies for generating a set of infill designs presented here have been implemented within the developed MEGO algorithm. The total optimization time gain is directly related to the number of available cores of the server. The previously-mentioned electromagnetic optimization test-problem *TEAM22* has been used as benchmark for evaluating the potential optimization time-gain resulting from the use of different number of cores for the server. The results of this benchmark are presented at the end of this chapter, in paragraph 2.6.4.3.

2.5.8 *Development of the MEGO tool – graphical user interface*

In order to facilitate the user the optimization problem formulation, as well as the parameter setting for the different modules of the MEGO algorithm, a graphical user interface (GUI) has been developed equally under Matlab®. The main window of the interface is presented in Figure 2.29.

The GUI is organized into several modules. The first module, entitled "Problem setup" gathers the general information relative to the optimization problem definition (number of design variables, objective and constraint functions, as well as the upper and lower bounds of the design variables (i.e. box constraints)).

The second module, named "Project info", assembles all information relative to the model used in the optimization process (project name and path, name of the function that launches the fine model evaluation). Remark that the GUI accepts only one function for launching both objective and expensive constraint functions. This decision was made with respect to the vision of the model as a black-box. In most electromagnetic engineering cases, the models of electromagnetic devices represent simulation models (e.g. FE model), hence both objective and constraint functions are obtained through just one launching of the model. However, the inexpensive constraints are basically geometric constraints, evaluated independently of the simulation model and with no (or much reduced) computational expense, thus specified here in a distinct function.

The third module of the interface, "Sampling plan", is dedicated to the experimental design (or sampling plan) selection and parameter setting. The type of desired initial sampling plan can be specified here by the user among a list of available experimental designs (e.g. random, random feasible, Latin Hypercube, Central Composite Design). An already evaluated sampling plan can be reused here, by specifying the corresponding data file.

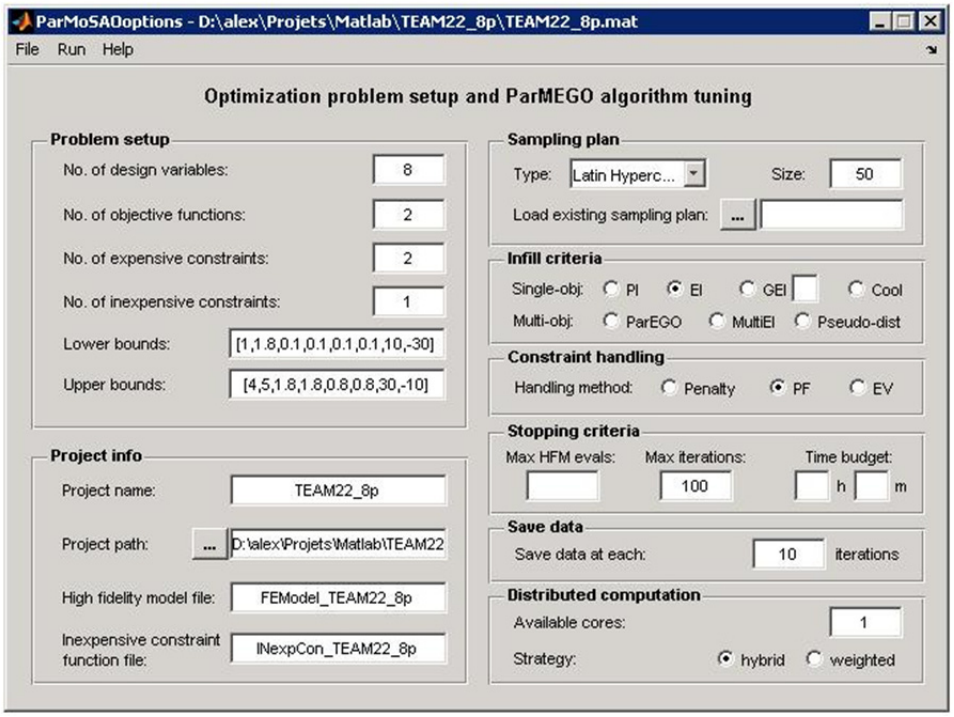


Figure 2.29 : Main window of the MEGO graphical user interface (GUI)

Several both single- and multi-objective infill criteria among those discussed in paragraphs 2.5.1 and 2.5.5.1, along with different constraint handling methods presented in paragraph 2.5.3 have been implemented within the algorithm and their selection is made possible in the graphical interface.

Other parameters, such as stopping criteria (e.g. given number of fine model evaluations, number of iterations, given time budget), results saving parameters and parameters required for distributing the fine model evaluation (number of available cores, parallelization strategy, server configuration) can also be specified in the interface.

2.5.9 MEGO coupling with ModeFrontier® optimization software

First of all, it is necessary to state that the author does not intend to promote here any commercial software in any way. The decision of choosing one optimization software over another for coupling with the developed algorithm has been made here purely on license availability considerations and ease of coupling implementation. Both tested optimization software, ModeFrontier® [EST 12] and Optimus® [NOE 12], offer the possibility of adding an external scheduler to their optimization process. For convenience, only one of the two software has been chosen for coupling with the MEGO algorithm developed in this work.

The advantages of coupling the developed MEGO algorithm with a commercial optimization software, such as Esteco's ModeFRONTIER® are manifold. Apart from the different optimization algorithms available under ModeFrontier®, the software fosters a large palette of statistical analysis tools (e.g. ANOVA, correlation matrices, scatter plot matrices, box-whiskers etc.) which are meant to supply the designer with insights into the model of the device to be optimally designed, thus helping him to analyze and formulate his optimization problem. The response surface methodology (RSM) is equally present in the software; metamodels such as polynomial, RBF or Kriging can be created and surfed with ease. A sum of response surface tools is equally available in the software. Moreover, the software beneficiates of an important number of visualization and decision-support tools (e.g. 3D and 4D bubble plots, history charts, parallel coordinates representations, clustering charts etc.), which are meant to assist the designer in the decision-making process. In order for the user to beneficiate from all these analysis and visualization tools, along with the developed MEGO algorithm, all within a homogenous design environment, the MEGO algorithm implemented in Matlab® has been coupled with the ModeFrontier® optimization software. This coupling was possible due to the opening of the commercial software offered by Esteco starting with version 4.3.0 of the ModeFrontier® software, by the means of an external scheduler bridge integrated in the software's workspace.

Figure 2.30 presents the coupling and the information flux between ModeFrontier® optimization environment and the MEGO algorithm developed under Matlab®, for the optimal design of a SMES device modeled by finite elements in Opera® 2D. The roles of the different modules presented in Figure 2.30, are presented in Table 2.4.

Table 2.4 : Role of the different modules in Figure 2.30 within the SMES optimization process

Module	Role
ModeFrontier® environment	start optimization, manage and visualize results
MEGO optimizer	perform optimization on a low-budget of FEM evaluations
Analyzer	launch FEM evaluation, export results to ModeFrontier®
Opera® 2D	perform FEM evaluation (pre-, post-processing and analysis)

The SMES device optimization is launched by ModeFrontier® in a master-slave process. ModeFrontier® sends a start signal to Matlab®, passing the control to the optimizer, the MEGO algorithm. The optimization process is then launched in Matlab® and the MEGO algorithm makes regular calls to the finite element model of the SMES device, throughout a Matlab®-developed bridge. The SMES model evaluation is done using the commercial finite element optimization software Opera® 2D of Cobham. The FE model evaluation implies three distinct phases: pre-processing (the mesh is generated for the given geometrical configuration), analysis (a static analysis is performed for the SMES model, generating a results file) and post-processing (analysis results are extracted and made available to the Matlab® model launching function by the Matlab®-Opera® bridge). Depending on the geometrical configuration, one FE model evaluation (pre-, post-processing and analysis) takes between 2-3 and 10-15 seconds. The Matlab® model launching function then exports the results to the ModeFrontier® optimization environment. Hence, the optimization process can be followed interactively in the ModeFrontier® environment. Once the optimization process has finished, the control is returned to ModeFrontier® and the optimization results are managed and analyzed in its environment.

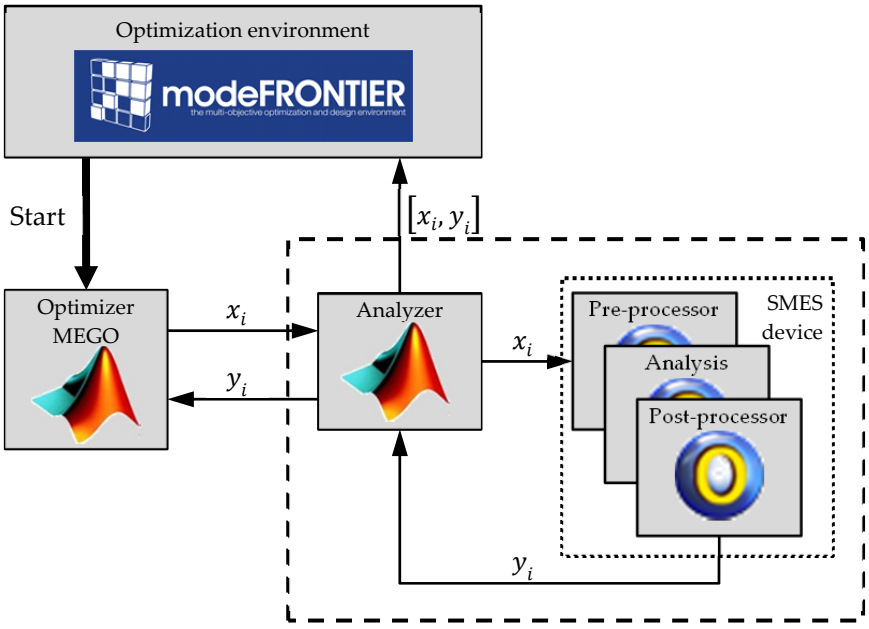


Figure 2.30 : MEGO optimizer integration within ModeFrontier® environment

2.6 Application: SMES device optimization problem

In order to assess the performance of the different single- and multi-objective MBDO approaches presented in this chapter, a classic electromagnetic test-problem of the electromagnetic community is considered. The device to be optimally sized is a superconducting magnetic energy storage system (SMES) [ALO 08], [MAG 08]. The SMES device consists of two concentric coils made of superconducting wires, disposed as in Figure 2.31. The coils are fed with energy from the network in opposite directions. Then, the coils are short-circuited, the energy being stored in the coil's magnetic fields. The energy is released when needed by reconnecting the coils to the network.

The goal of the optimization benchmark is to find those configurations (geometric and electric parameters of the coils) that give a storage capability of 50 kWh (180 MJ) of the device, and a minimum value for the stray field at a given distance from the device.

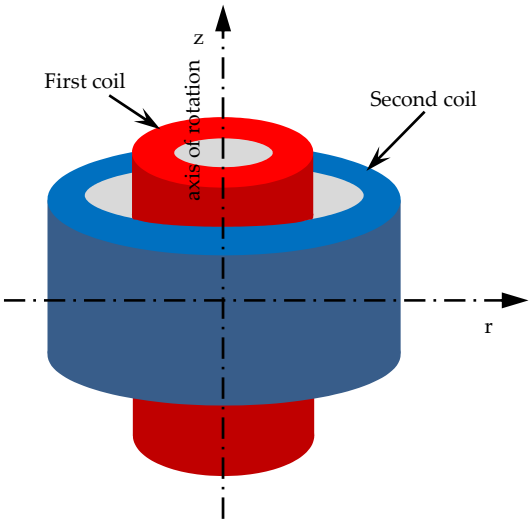


Figure 2.31 : 3D representation of the SMES device

2.6.1 *TEAM22 benchmark description*

The SMES device optimization problem has been first proposed as single-objective constrained optimization benchmark to the international electromagnetic community in [ALO 96] and it is known under the name *TEAM22* benchmark or *TEAM22* test-case problem.

A detailed description of the *TEAM22* benchmark, along with some optimal results and Fortran and Python implementations for the SMES model evaluation can be found on the IGTE web site[15].

To present the different design parameters of both coils of the SMES device, a transverse cut view of the SMES device is given in Figure 2.32.

The goal of the optimization problem is to find those design configurations which offer a value for the energy stored by the SMES device as close as possible to a reference value ($E_{ref} = 180MJ$) and a value for the stray field, measured in 22 points along a line at 10m from the device (represented in Figure 2.32), as small as possible compared to a reference value.

[15] *TEAM22* problem on the IGTE web site at: http://www.igte.tugraz.at/team22/index.php.

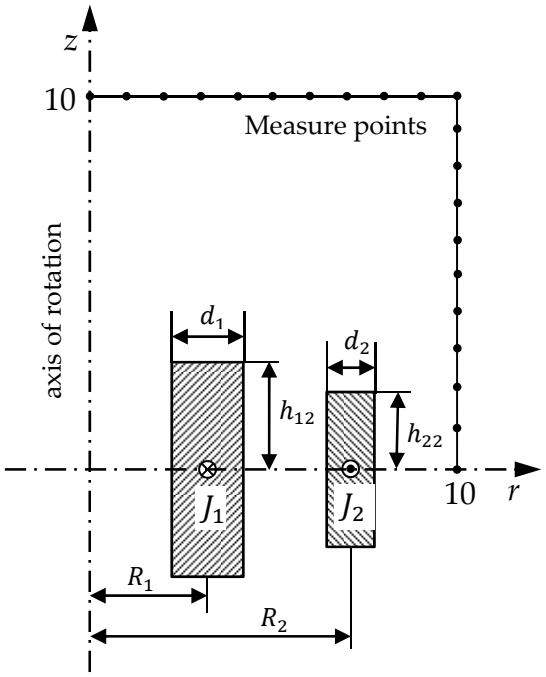


Figure 2.32 : Representation of the right-half transverse cut over the SMES device

There are two formulations of the *TEAM22* benchmark, based on the number of design variables of the optimization problem. The first formulation consists of finding the optimal dimensions of the outer coil (R_2, h_{22}, d_2), while considering known the geometry of the inner coil and the current densities in both coils. The second formulation is more general, having as purpose the optimal sizing of both coils, thus accounting for all 8 design variables ($R_1, R_2, h_{12}, h_{22}, d_1, d_2, J_1, J_2$). While the first formulation considers discrete values for the three variables, for the second formulation all design variables are considered continuous.

Table 2.5 : Design variables and constants for the three-parameter *TEAM22* benchmark

Parameter	$\boldsymbol{R_1}$	$\boldsymbol{R_2}$	$\boldsymbol{h_{12}}$	$\boldsymbol{h_{22}}$	$\boldsymbol{d_1}$	$\boldsymbol{d_2}$	$\boldsymbol{J_1}$	$\boldsymbol{J_2}$
Unit	$[m]$	$[m]$	$[m]$	$[m]$	$[m]$	$[m]$	$[MA/m^2]$	$[MA/m^2]$
Min	-	2.6	-	0.204	-	0.1	-	-
Max	-	3.4	-	1.1	-	0.4	-	-
Fixed	2.0	-	0.8	-	0.27	-	22.5	-22.5

However, the work in this study does not address discrete optimization problems, thus for both formulations of the *TEAM22* benchmark we will consider all design variables as continuous.

Table 2.6 : Design variables for the eight-parameter *TEAM22* benchmark

Parameter	$\boldsymbol{R_1}$	$\boldsymbol{R_2}$	$\boldsymbol{h_{12}}$	$\boldsymbol{h_{22}}$	$\boldsymbol{d_1}$	$\boldsymbol{d_2}$	$\boldsymbol{J_1}$	$\boldsymbol{J_2}$
Unit	$[m]$	$[m]$	$[m]$	$[m]$	$[m]$	$[m]$	$[MA/m^2]$	$[MA/m^2]$
Min	1.0	1.8	0.1	0.1	0.1	0.1	10	-30
Max	4.0	5.0	1.8	1.8	0.8	0.8	30	-10

For all optimal designs that are to be found, the superconductive state of the superconductors forming both coils should be preserved. This condition is called the "quench condition" and it is graphically presented in Figure 2.33 through the dashed line, which is a linearization of the NbTi superconductor curve.

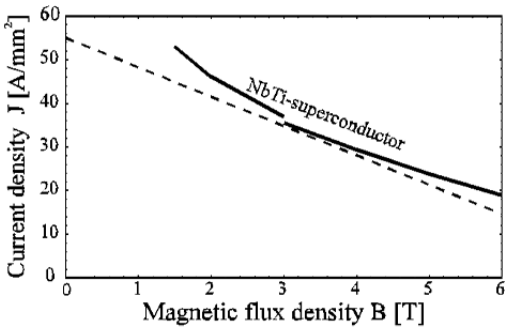


Figure 2.33 : Critical curve for ensuring the superconductivity state of the superconductor

Therefore, the optimization problem is constrained and the following two constraint functions are to be respected:

$$|J_1| \leq (-6.4|B_{max1}| + 54) \tag{2.39}$$

$$|J_2| \leq (-6.4|B_{max2}| + 54) \tag{2.40}$$

where B_{max1} and B_{max2} represent the maximum values of the magnetic induction in the inner, respectively the outer coil.

Furthermore, for the 8-parameter formulation, since all geometrical parameters of both coils are considered variable, a geometrical constraint which is meant to prevent the overlapping of the two coils is present within the optimization problem. This constraint function is expressed in (2.41).

$$R_1 + \frac{d_1}{2} < R_2 - \frac{d_2}{2} \tag{2.41}$$

2.6.2 Modeling of the SMES device

A finite element model (FEM) of the SMES device has been developed in 2D using the commercial FE software Opera®, product of Cobham. The FE model is parameterized; the model construction through command lines allows thus the automated modification of the parameters of the device within an optimization process. The mesh of the geometry is also parameterized, depending on the different values of the geometrical parameters for a given configuration of the SMES device.

The SMES device presents an axis-symmetry property, which is accounted for within the FE modeling of the device. A first symmetry, following the axis of rotation z, is accounted for by considering only the right half-side of the device. Moreover, given the symmetry following the plane $z = 0$, the modeled region of the device was reduced to the superior half of the coils. A global view over the device geometry and mesh for a given configuration of the SMES device is presented in Figure 2.34. For this particular design, the Opera 2D FEM mesh consists of 9877 elements and 19938 nodes.

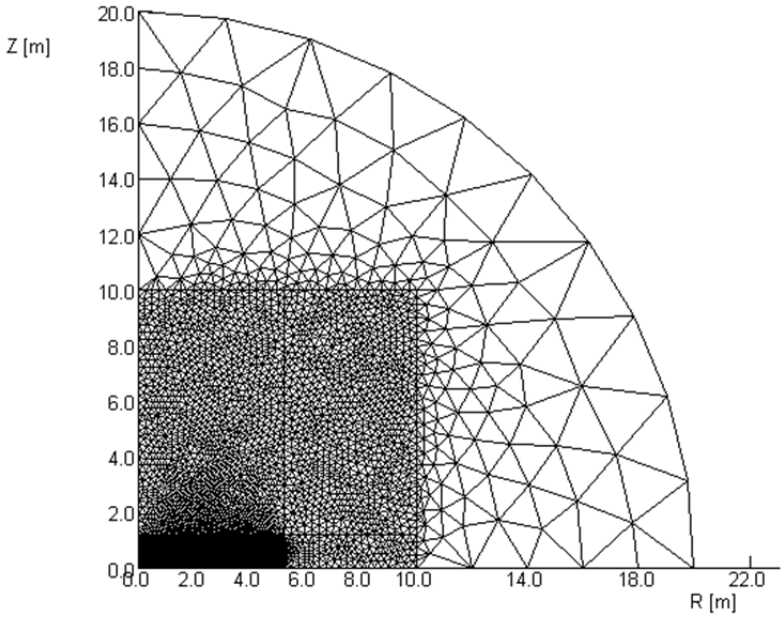


Figure 2.34 : General view over the FEM of the SMES device for a given configuration

In the area surrounding the two coils, the mesh was densified and rectangular mesh elements were chosen in order to accurately represent the physical phenomena. Elsewhere, the mesh was constructed using triangular elements, and is less denser than in the region of the coils. Figure 2.35 presents a zoomed view over the denser-meshed region where the two coils lie.

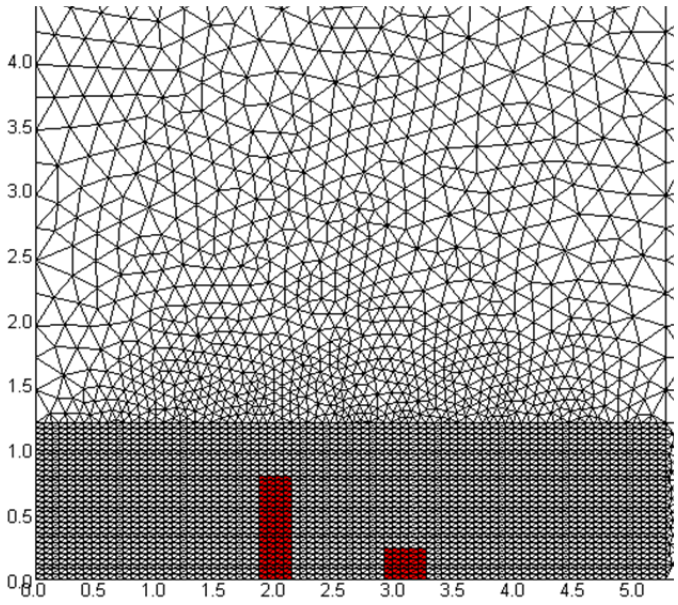


Figure 2.35 : Zoom over the area containing the two coils of the SMES device

2.6.3 Single-objective optimization of the 3-parameter TEAM22 benchmark problem

The three MBDO strategies previously presented in this chapter are used to solve the optimal design problem of the *TEAM22*'s SMES device. For comparison reasons, a trust-region optimization algorithm (SQP) with a multi-start strategy, along with a genetic algorithm (GA) and a hybrid algorithm (GA hybridized with SQP) are also used to find the optimal configurations of the SMES device. The three-parameter *TEAM22* optimization problem formulation consists of finding the optimal dimensions of the outer coil of the SMES device, with regard to the stored energy and the stray field. The inner coil configuration and the current density of the second coil are fixed parameters.

The single-objective mathematical formulation of the three-parameter *TEAM22* optimization problem is given in (2.42).

$$
\begin{aligned}
&\underset{\mathbf{x}}{\text{Minimize}} \quad f(\mathbf{x}) = \frac{B^2_{stray}(\mathbf{x})}{B^2_{norm}} + \frac{\left|E(\mathbf{x}) - E_{ref}\right|}{E_{ref}} \\
&\text{with} \quad B^2_{stray}(\mathbf{x}) = \frac{\sum_{i=1}^{22}\left|B_{stray,i}(\mathbf{x})\right|^2}{22} \\
&\qquad E_{ref} = 180MJ,\ B_{norm} = 3mT \\
&\qquad \mathbf{x} = [R_2, h_{22}, d_2] \\
&\qquad R_1 = 2m \qquad h_{12} = 0.8m \qquad d_1 = 0.27m \qquad J_1 = 22.5MA/m^2 \\
&\qquad R_2 \in [2.6,3.4]m \qquad h_{22} \in [0.204,1.1]m \qquad d_2 \in [0.1,0.4]m \qquad J_2 = -22.5MA/m^2 \\
&\text{such as} \quad g_{exp}(\mathbf{x}) = |J_2| - (-6.4|B_2^{max}(\mathbf{x})| + 54) \leq 0
\end{aligned}
\tag{2.42}
$$

where $B_{stray,i}$ is the stray field calculated at the location of the i-th point (among the 22 measure points), g_{exp} is the expensive constraint function that ensures the superconductivity state of the materials which form the outer coil.

The two goals of the optimization problem, the minimal stray field respectively the given level of energy stored by the SMES device are aggregated within the expression of a single objective function f using a weighting formulation, where equal weights are given to the two goals.

2.6.3.1 Optimal SMES design using classical global optimization strategies

Three classical global optimization strategies, a trust-region algorithm (SQP) with an LHS-based multi-start, a genetic algorithm (GA) and a hybrid of a genetic algorithm and a trust-region algorithm (GA hybridized with SQP) were first used to find the optimal design configuration of the SMES device's outer coil. The numerical results of the optimal SMES configurations found by the three global optimization strategies are presented in Table 2.7.

Table 2.7 : Optimal configuration of the SMES found by classical global optimization strategies

Classical global optimizer	**Type of optimum**	**Design variables**			**Outputs**		**Objective**	**Nb. model evaluations**
		R_2	h_{22}	d_2	E	B_{stray}	f	
	loc./glob.	$[m]$	$[m]$	$[m]$	$[MJ]$	$[mT]$	$[-]$	$[-]$
	Target	-	-	-	= 180	< 3	-	-
SQP30runs	global	3.0825	0.2462	0.3812	179.9999	0.8886	0.0877	14 264
GA	local	3.0918	0.2747	0.3396	179.9996	0.8920	0.0884	12 510
hybrid	global	3.0827	0.2480	0.3784	179.9999	0.8885	0.0877	13 087

The histogram of the 30 runs of the SQP algorithm is presented in Figure 2.36.

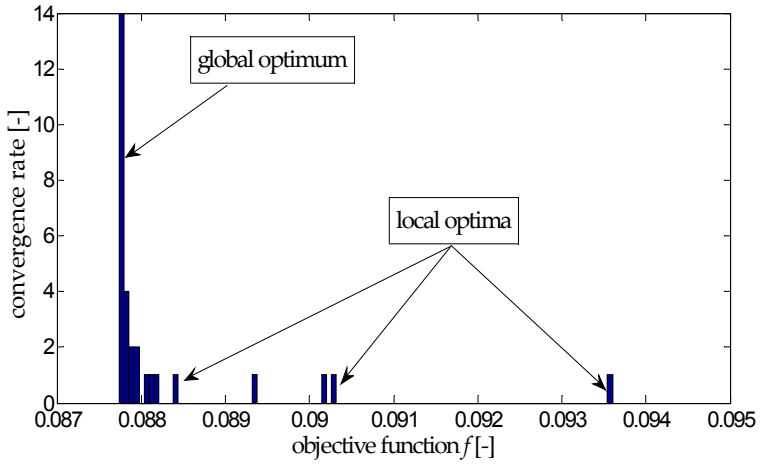


Figure 2.36 : Histogram of the SQP algorithm with 30 runs

From the histogram in Figure 2.36 one can see that the three-parameter *TEAM22* optimization problem is multi-modal. The multi-start strategy has identified an important number of local optima and one global optimum corresponding to the design vector $\mathbf{x}^* = [3.0825, 0.2462, 0.3812]$, and with a value of $f^* = 0.0877$ for the objective function. Also, it can be observed that the convergence rate to the global optimum of the problem is relatively low (under 50%), with only 14 out of 30 runs converging to this optimum. This global optimum has been also confirmed by the result of the hybrid optimization strategy.

The total number of SMES fine model evaluations is about the same for all three optimization strategies, and is relatively high. Considering a mean time of 5 seconds per fine model evaluation, each of the three optimizations took about 2 hours, in order to find the optimal values for only three parameters of the outer coil. Hence, for the full eight-parameter *TEAM22* optimization benchmark, the total number of fine model evaluations and the total optimization time would be, using a simple extrapolation, of about 35000 fine model evaluations, respectively ~6 hours. These classical global optimization strategies become very quickly prohibitive for an optimization problem with a time-consuming fine model (superior to a few minutes), as for the case of the LIM design optimization problem presented in paragraph 2.4.4. Thus, we will next focus our attention on the performance of the three metamodel-based optimization strategies developed in this chapter, applied for the single-objective optimization of the SMES device. Using a metamodel to guide the search for optimal

designs, these global optimization strategies propose to find the global solution of an optimization problem with a reduced budget of fine model evaluations.

2.6.3.2 Sequential metamodel-based optimization of the SMES

The first-presented metamodel-based optimization technique (paragraph 2.3) and also the most basic – the sequential approach – is used here to find the optimal configuration of the SMES device using the single-objective problem formulation in (2.42). For this, six different optimization runs were performed using different sets of Kriging metamodels of objective and constraint functions constructed based on six experimental designs of varying size. The experimental designs were selected using the Latin Hypercube Strategy (LHS) and are composed of 10, 30, 100, 300, 500 and 1000 designs. The optimal results of the optimization problem corresponding to the six runs are presented in Table 2.8.

Table 2.8 : Optimal configurations of the SMES device's outer coil obtained using the six runs of the sequential metamodel-based optimization approach

LHS	**Design variables**			**Outputs**		**Objective function**			**Constraint function**		
	R_2	h_{22}	d_2	E	B_{stray}	$\hat{f}$	f	ε_f	$\hat{g}_{exp}$	g_{exp}	$\varepsilon_{g_{exp}}$
	$[m]$	$[m]$	$[m]$	$[MJ]$	$[mT]$	$[-]$	$[-]$	[%]	$[-]$	$[-]$	[%]
10 pts.	2.6000	0.5693	0.3730	193.29	8.904	-150.93	8.88	1799	9.6e-13	13.29	99
30 pts.	2.9685	0.2040	0.1000	179.47	11.196	-986.94	13.93	7180	-12.28	-18.2	32
100 pts.	2.6002	0.6779	0.1159	127.00	5.553	-2.52	3.72	167	-4.73	-10.98	57
300 pts.	3.3101	0.8087	0.1009	173.23	0.977	-0.039	0.144	127	-15.96	-17.1	7
500 pts.	3.3808	0.7853	0.1001	176.46	0.933	0.114	0.143	20	-15.18	-17.51	13
1000 pts.	3.0767	0.2657	0.3550	179.48	0.882	0.0902	0.0893	1.01	-2.61	-1.79	46

From the results presented in Table 2.8, looking at the values of the objective and constraint function errors, ε_f respectively $\varepsilon_{g_{exp}}$, one can remark the fact that the global accuracy of a metamodel depends directly on the size of the experimental design used for fitting the metamodel. While the first two tests, using metamodels created on experimental designs of 10 and 30 points are far from representing the true SMES model, the last one, created using an LHS of 1000 points predicts with a good accuracy the outputs of the true SMES model. This makes the optimization algorithm employed for searching the minimum of the function, find the real optimum of the *TEAM22* problem ($f^* = 0.0877$), with a high accuracy ($f = 0.0893$). Nevertheless, the necessary time for constructing such a model (approx. 2h for 1000 LHS evaluations) is very important, making this approach very unpractical. This becomes much more critical when the employed simulation model needs to perform transient or dynamic simulations for complex electrical devices, such as a transformer or a motor, when a single model evaluation time is of the order of hours, even days.

To better visualize the global accuracy of the 6 previously created objective function metamodels, a test set of 100 randomly generated designs is considered. These designs are then evaluated both using the true (FEM) model of the SMES, and the 6 metamodels, their objective function values represented by the variables Y, respectively $\hat{Y}_k$, where $k \in \{10,30,100,300,500,1000\}$. For each of the 6 metamodels, the error measure expressed in (2.43) is calculated.

$$\varepsilon_k = \left| \frac{Y - \hat{Y}_k}{Y} \right| \tag{2.43}$$

where ε_k is a column vector of 100 elements, corresponding to the relative error of each of the 100 selected test points.

The 6 error measures corresponding to the 6 metamodels are represented graphically in Figure 2.37 using a box-plot representation, which was introduced in Chapter 1. However, due to the large error presented by the first two metamodels, the error measure ε_k cannot be properly displayed, thus its logarithm, $\log \varepsilon_k$ was instead represented in the box-plot.

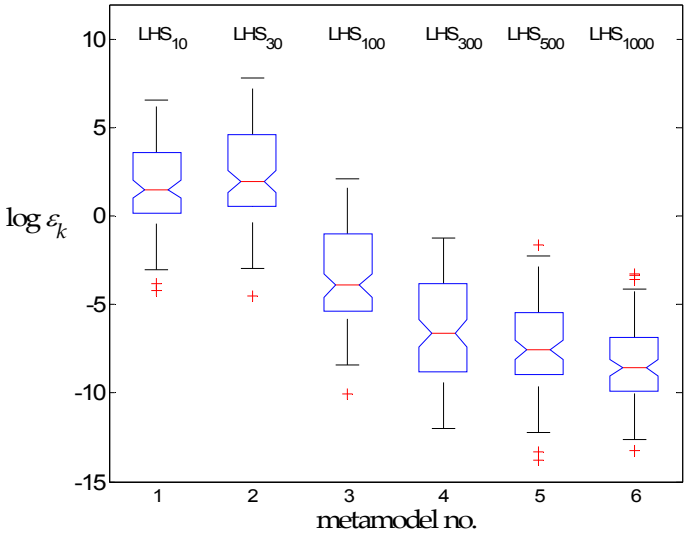


Figure 2.37 : Box-plot representation of the metamodels error measure, $\log \varepsilon_k$

The box-plot representation confirms the previous observations, that the global metamodel accuracy increases with the size of the support data.

In the attempt to reduce the *TEAM22* optimization time, by reducing the necessary number of design evaluations for attaining a proper metamodel accuracy, we will next investigate the second MBDO approach presented in paragraph 2.4.

2.6.3.3 Adaptive metamodel-based optimization of the SMES

The adaptive metamodel-based optimization technique presented in paragraph 2.4 is considered here in order to find the optimal configuration of the SMES device's outer coil. Several optimization tests, using different sizes of the initial experimental design (LHS of 10, 30, 100, 300, 500 and 1000 points) were performed. A number of 100 infill designs were imposed for each optimization run.

The numerical results of the different optimization runs are regrouped in Table 2.9.

Table 2.9 : Optimal configurations of the SMES device's outer coil obtained using the adaptive MBDO approach and the six metamodels

LHS	**Design variables**			**Outputs**		**Objective function**			**Constraint function**		
	R_2	h_{22}	d_2	E	B_{stray}	$\hat{f}$	f	ε_f	$\hat{g}_{exp}$	g_{exp}	$\varepsilon_{g_{exp}}$
	$[m]$	$[m]$	$[m]$	$[MJ]$	$[mT]$	$[-]$	$[-]$	[%]	$[-]$	$[-]$	[%]
10 pts.	3.0999	0.3098	0.3017	179.89	0.8015	0.1026	0.0916	1.2e-1	-3.3567	-3.3551	4.7e-4
30 pts.	3.0668	0.2425	0.3921	179.37	0.7788	0.0959	0.0895	7.1e-2	-1.0019	-1.0248	2.2e-2
100 pts.	3.0664	0.2363	0.4000	179.00	0.7704	0.0923	0.0905	2.0e-2	-0.9994	-0.9995	7.4e-5
300 pts.	3.1249	0.3172	0.2878	180.37	0.8060	0.1027	0.0948	8.4e-2	-4.1794	-4.1772	5.2e-4
500 pts.	3.0808	0.2595	0.3613	179.59	0.7791	0.0872	0.0890	1.9e-2	-1.7710	-1.7656	3.0e-3
1000 pts.	3.0896	0.2759	0.3393	179.97	0.7888	0.0902	0.0886	1.8e-2	-2.3150	-2.2645	2.2e-2

From the results presented in Table 2.9, it can be seen that all six optimization runs performed well, with optimal values of the objective function very close to each other and to the real optimum of the problem. In comparison with the previous approach, analyzed in paragraph 2.6.3.2, where the size of the experimental design had a crucial impact on the performance of the optimization process, here we can remark that the initial experimental design size practically does not influence the performance of the optimization process. Also, it can be remarked that for the optimal sites, the error between the fine model and the metamodels of the objective, respectively constraint function, does not depend on the size of the initial experimental design. This can be explained by the fact that systematically sampling at metamodel-predicted optimal design sites and adding these designs to the list of support points for new metamodels, the global accuracy of metamodels is also increased. However, it must be stated that the good performance of this MBDO approach is also given by the reduced size of the design space, i.e. low number of design variables, and the fact that the unique constraint function is not active at the site of the optimum, i.e. it is different from zero.

Next, the criterion-based EGO algorithm will be analyzed, with regard to several infill sampling criteria which balance between the search of improving designs and the exploration of the design space, and using different constraint handling techniques.

2.6.3.4 Criterion-based metamodel optimization of the SMES (EGO algorithm)

The optimization of the SMES device's outer coil configuration using the EGO algorithm using different infill criteria described in paragraph 2.5.1 is referred here. The probability of improvement (*PI*), expected improvement (*EI*), generalized expected improvement (*GEI*) and the cooling strategy infill point selection criteria were considered with the EGO algorithm for finding the optimal dimensions of the outer coil of the device. The expensive constraint function of the optimization problem was handled both through the penalty formulation and its *PF* measure. Each optimization run was launched with an initial experimental design of 50 points and a total budget of 150 SMES fine model evaluations was imposed. The different optimal configurations found using each optimization run are presented in Table 2.10.

Table 2.10 : Optimal configurations of the SMES device's outer coil obtained using the EGO algorithm with different infill point selection criteria

Infill Criterion	**Constr. handling**	**Design variables**			**Outputs**		**Obj. fun.**	**Constr.**	**SMES evals.**
		R_2	h_{22}	d_2	E	B_{stray}	f	g_{exp}	
		$[m]$	$[m]$	$[m]$	$[MJ]$	$[mT]$	$[-]$	$[-]$	$[-]$
PI	Penalty	3.0671	0.2431	0.3880	178.91	0.7685	0.0909	-1.18	83
	PF	3.1110	0.2585	0.3538	180.78	0.7969	0.0966	-2.39	99
EI	Penalty	3.0937	0.2503	0.3722	180.50	0.7921	0.0922	-1.75	123
	PF	3.0556	0.2596	0.3704	178.81	0.7764	0.0920	-1.13	113
GEI ($g = 2$)	Penalty	3.0929	0.2921	0.3184	179.47	0.7834	0.0907	-2.82	79
	PF	3.0606	0.2472	0.3829	178.46	0.7624	0.0922	-1.16	59
GEI ($g = 5$)	Penalty	3.0856	0.2951	0.3190	179.47	0.7887	0.0911	-2.64	136
	PF	3.0958	0.2737	0.3384	179.91	0.7855	0.0893	-2.44	60
Cooling strategy	Penalty	3.0932	0.3103	0.3029	179.61	0.7989	0.0926	-3.23	140
	PF	3.0782	0.2682	0.3524	179.69	0.7856	0.0892	-1.83	150

From the results presented in Table 2.10, it can be seen that all tested infill criteria performed well, with no distinction of a best criterion. Similar results were obtained for all five tested criteria. Since EGO is a global algorithm, one can remark that the optimal results were not found with an increased precision. For an increased accuracy of the optimum, a trust-region algorithm such as SQP might be further employed, using as start point the optimum found by the EGO algorithm, hence ensuring the required accuracy of results. Both the penalty method and the *PF* constraint handling techniques performed well. Compared to the other two MBDO strategies, presented in the previous paragraphs, the number of fine model evaluations is drastically reduced (a mean of 104 fine model evaluations per optimization run). This fine model evaluations reduction is translated into the overall SMES optimization time reduction from a couple of hours, as in the case of the sequential MBDO, paragraph 2.6.3.2, to only 20-25 minutes.

For a better visualization of the optimization results, the parallel coordinates representation introduced in Chapter 1 of the manuscript was considered in Figure 2.38.

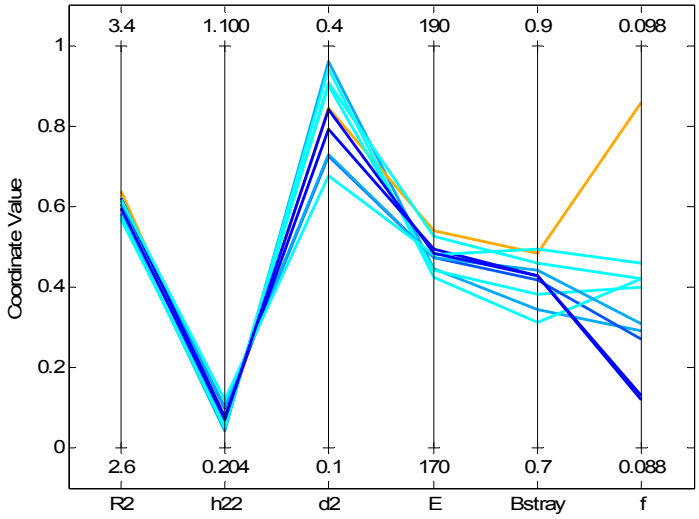


Figure 2.38 : Parallel coordinates representation of EGO algorithm runs with tested infill criteria

One interesting remark that can be drawn from Figure 2.38 is that for the obtained optimal results, the energy E and the stray field B_{stray} are somehow directly correlated. The color of the polylines in Figure 2.38 is associated to the value of the objective function f. Better results are represented using darker blue color. While the optimal values of R_2 and h_{22} are concentrated around one point, the optimal values of d_2 are displaced within an interval as large as about 30% of the variation range of the variable. The objective function is very sensitive to the values of d_2.

After overviewing the single-objective optimization of the SMES device, we will next focus our attention on the multi-criteria optimization of the SMES. The optimal trade-off between the two distinct optimization criteria of the SMES device is sought using the different presented multi-objective MBDO strategies in the next paragraph.

2.6.4 *Bi-objective optimization of the SMES device*

The 8-parameter formulation of the *TEAM22* optimization benchmark was considered in order to test the behavior of the developed MEGO algorithm. Instead of aggregating the two objectives of the problem within the formulation of one objective function, as with the initial single-objective formulation, the energy, respectively the stray field objective functions are handled individually, within a bi-objective optimization problem formulation [ALO 08]. The mathematical formulation of the bi-objective optimization problem is expressed in (2.44).

$$\begin{aligned}
&\underset{\mathbf{x}}{\text{Minimize}} \quad f_1(\mathbf{x}) = OF_1(\mathbf{x}) = \frac{\left|E(\mathbf{x}) - E_{ref}\right|}{E_{ref}} \\
&\qquad\qquad f_2(\mathbf{x}) = OF_2(\mathbf{x}) = \frac{B^2_{stray}(\mathbf{x})}{B^2_{norm}} = \frac{1}{B^2_{norm}} \cdot \frac{\sum_{i=1}^{22}\left|B_{stray,i}(\mathbf{x})\right|^2}{22} \\
&\text{with} \quad E_{ref} = 180MJ,\ B_{norm} = 200\mu T \\
&\qquad \mathbf{x} = [R_1, R_2, h_{12}, h_{22}, d_1, d_2, J_1, J_2] \\
&\qquad R_1 \in [1,4]m \qquad h_{12} \in [0.1,1.8]m \quad d_1 \in [0.1,0.8]m \quad J_1 \in [10,30]MA/m^2 \\
&\qquad R_2 \in [1.8,5]m \qquad h_{22} \in [0.1,1.8]m \quad d_2 \in [0.1,0.8]m \quad J_2 \in [-30,-10]MA/m^2 \\
&\text{such as} \quad g_{inexp}(\mathbf{x}) = \left(R_1 + \frac{d_1}{2}\right) - \left(R_2 - \frac{d_2}{2}\right) \le 0 \\
&\qquad g_{exp1}(\mathbf{x}) = |J_1| - (-6.4|B_1^{max}(\mathbf{x})| + 54) \le 0 \\
&\qquad g_{exp2}(\mathbf{x}) = |J_2| - (-6.4|B_2^{max}(\mathbf{x})| + 54) \le 0
\end{aligned} \tag{2.44}$$

where $B_{stray,i}$ is the stray field calculated at the location of the *i*-th point (among the 22 measure points), g_{inexp} represents the geometrical constraint that prevents the two coils from overlapping, g_{exp1} and g_{exp2} are the expensive constraint functions that ensure the superconductivity state of the materials which form the two coils.

The developed multi-objective criterion-based optimization algorithm MEGO is used for finding the bi-objective non-dominated trade-off solutions of the *TEAM22* benchmark problem.

2.6.4.1 Bi-objective SMES device optimization using MBDO

As we have seen in paragraph 2.6.3.2, a very large number of sample points (~1000) were necessary to obtain a globally accurate metamodel of the 3-parameter SMES device. For the bi-objective 8-parameter design problem of the SMES device, a prohibitive number of sample points are expected for attaining a sufficiently accurate model which could be able to determine true Pareto front. Therefore, this MBDO method is not considered here, since it would require a huge amount of computational effort, and in revenge the metamodels global accuracy is not even guaranteed. Since the performance of the multi-objective adaptive MBDO approach has been already addressed for the optimization of the LIM in paragraph 2.4.4, it will not be discussed here. The presentation of the multi-objective SMES optimization is centered here on the developed multi-objective MEGO algorithm and the features of this algorithm.

2.6.4.2 SMES optimization using the multi-objective MEGO algorithm

The developed multi-objective MEGO algorithm presented in paragraph 2.5.6 implementing the multi-objective pseudo-distance infill criterion presented in paragraph 2.5.5.2 was used here to find the optimal trade-off designs of the SMES device. For the initialization step of the MEGO algorithm, an initial experimental design of 100 points ($\sim 10 \cdot n_{var}$) was selected using a Latin Hypercube strategy and evaluated using the FE model of the device. The MEGO algorithm was launched with an imposed total budget of 300 fine model (FEM) evaluations, part of them (100) distributed to the initial experimental design and the rest of 200 infill points added by the MEGO algorithm. The resulting Pareto front of *TEAM22* problem is presented in Figure 2.39.

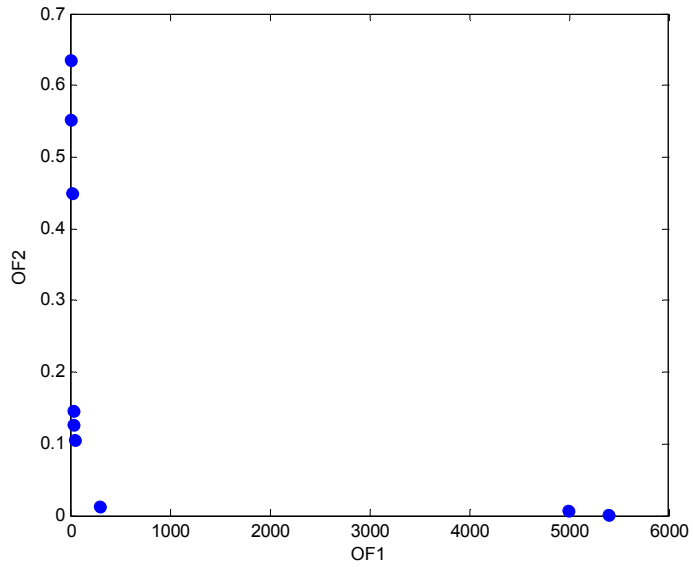


Figure 2.39 : Pareto front of the 8-parameter *TEAM22* optimization benchmark

In Figure 2.39 one can remark the "L"-shaped form of the obtained Pareto front, which is specific to this optimization benchmark [ALO 08]. Moreover, there is still an important scale difference between the two objective functions, in spite the fact that the functions have been scaled.

The entire optimization process takes about six hours of computation. For this case, the FE model is relatively fast, with a mean evaluation time for one configuration of about 4.3 seconds. A

particularly time-consuming step of the algorithm is represented by the Kriging metamodels construction phase, whose time demand augments with each point added to the list of sampled data ($\mathcal{L}$) used for training new Kriging metamodels.

In order to reduce the computation time, the MEGO optimization process was tested using the two proposed approaches for generating a set of infill designs at each iteration of the algorithm. The optimization results as well as the time gain realized through these approaches are next presented.

2.6.4.3 SMES optimization using MEGO and the distributed computation

Several optimization tests were conducted using different core configurations of a server, from 2 up to 8 cores, in order to assess the performance of the MEGO algorithm and the potential time-gain obtained by distributing the computation of FE model of the SMES. The two approaches, "hybrid model" and "weighted pseudo-distance" presented in paragraph 2.5.7.1, respectively 2.5.7.2, for generating a set of infill designs at each iteration of the MEGO algorithm have been tested. The corresponding Pareto fronts for the 8-parameter *TEAM22* optimization benchmark problem are presented in Figure 2.40. For each optimization run, the same initial set of points was used. For a better visualization of the results and due to the fact that all other Pareto fronts were equivalent, for the "weighted pseudo-distance" approach only one server configuration was considered for presentation in Figure 2.40 (8 core configuration).

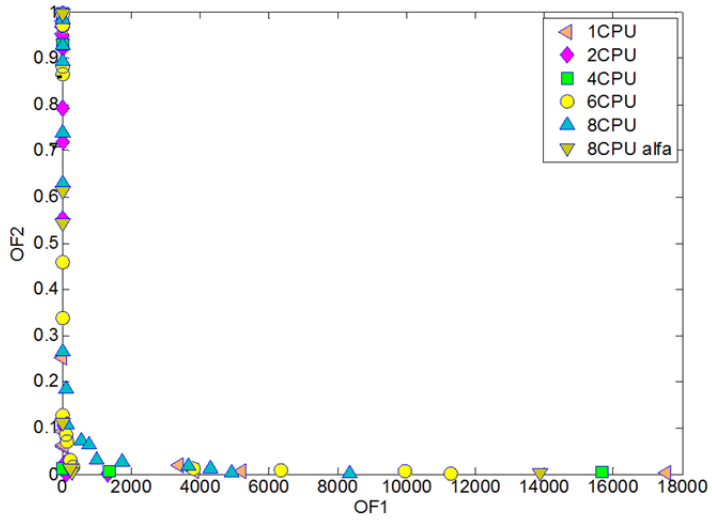


Figure 2.40 : Pareto front results with distributed computation on different server configurations

It can be seen from Figure 2.40 that all Pareto fronts obtained using different core configurations of the server are equivalent between each other and also equivalent to the initial sequential MEGO Pareto front (without distributed computation) previously presented in Figure 2.39. Both tested approaches offer good results and thus can be used with as many cores as there are available. The assessment of the computational time gain for each configuration of cores is next presented.

Speedup assessment for several server configurations

The time-gain obtained by performing the FE model computation on more than one core of the server was assessed using the speedup measure. The speedup (i.e. computation acceleration) was

calculated for all configurations (1 to 8 cores) of the server and the results are given in Table 2.11 and Table 2.12 for the DOE phase, respectively the iteration phase of the MEGO algorithm.

Table 2.11 : FEM evaluation time and speedup for the initial DOE phase (LHS of 80 points)

No. of cores used			**1**	**2**	**4**	**6**	**8**
FEA time	$t_1 - t_0$	$[s]$	281	152	87	74	68
FEA time/eval	$(t_1 - t_0)/n_{DOE}$	$[s]$	3.5	1.9	1.1	0.9	0.9
Speedup	$T_{(1\_core)}/T_{(n\_cores)}$	$[-]$	1	1.85	3.22	3.83	4.09

A similar speedup was obtained for the iteration phase of the MEGO algorithm. The time markers ($t_0 \div t_5$) can be found on the Figure 2.28 of MEGO workflow with distributed computation.

Table 2.12 : FEM evaluation time and speedup for the iteration phase of the MEGO algorithm

No. of cores used			**1**	**2**	**4**	**6**	**8**
FEA time	$\sum_{iter}(t_4 - t_3)$	$[s]$	3034	1846	991	804	757
FEA time/eval	$\sum_{iter}(t_4 - t_3)/n_{IP}$	$[s]$	4.3	2.6	1.4	1.15	1.1
Speedup	$T_{(1\_core)}/T_{(n\_cores)}$	$[-]$	1	1.64	3.06	3.77	4.01

The speedup for both the initial DOE step and the iteration phase of the MEGO algorithm is presented graphically in Figure 2.41.

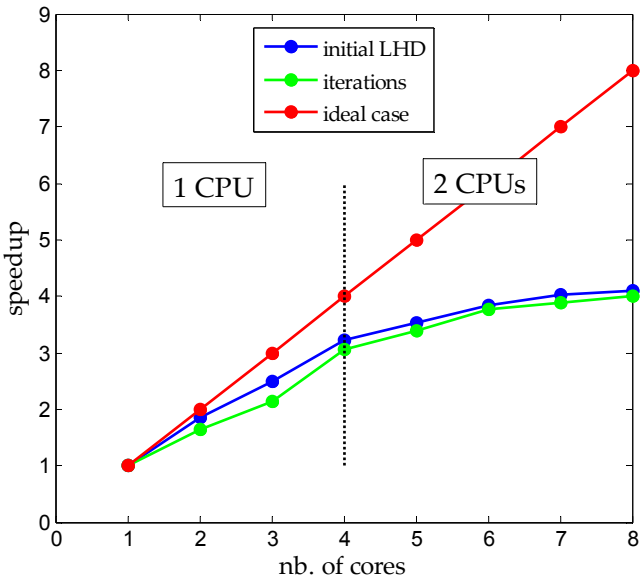


Figure 2.41 : Speedup for the initial DOE and iteration step of the MEGO algorithm in the case of the SMES device optimization

A maximum speedup of approximately 4 was obtained when using a total number of 8 cores of the server for distributing the FE model evaluation. This is far from the ideal case of a speedup of 8 (i.e. when distributing the FEM evaluation on n cores, ideally the computation should be n times faster). The cause of this resides in the fact that the time of one FEM evaluation is dependent of the geometry of the device, as the FE mesh is adaptive. This means that large device configurations are

more demanding in terms of computational time. Therefore, at a given iteration of the MEGO algorithm, the time consumed by the n FEM evaluations using n available cores of the server is not $1/n$ of the time of one FEM evaluation, but the time of the slowest of the n FEM evaluations; all other already finished FEM evaluations wait up for the slowest computation to finish.

Moreover, one can remark from Figure 2.41 that the speedup follows approximately the trend of the ideal case up to 4 cores. For server configurations of 5 to 8 available cores, the speedup is practically the same; very little computational time-gain is obtained. This can be explained through the physical configuration of the server, which has 2 CPUs with 4 cores on each CPU. Therefore, from 1 to 4 cores, a single CPU of the server is used to perform the FE model computation. When a 5th core is used for the computation, the 2nd CPU of the server is solicited, thus the interaction between the two CPUs which address the same memory slows down the computation.

2.7 Conclusion

In this chapter, the integration of metamodels into the optimization process was addressed. The study of integrating metamodels into the optimization process started with the most basic approach, where a response surface replaces completely the fine model during the whole duration of the optimization process. One important feature, which is used by the subsequent approaches, consists of the fact that the global accuracy of the metamodels is directly related to the size of the experimental design. It has been shown that constructing metamodels of a fine model with a high level of global accuracy requires an important amount of computation, which is translated into a prohibitive overall optimization time. Moreover, for higher dimensionality problems, presenting highly nonlinear trends of the objective and/or constraint functions, the required global accuracy of metamodels is impossible to attain, even with a large experimental design. This limits the application of this strategy for optimization purposes.

The analysis of metamodels integration into the optimization process continued then with an adaptive metamodel optimization process. The initially created metamodel is improved along several iterations, by using the optimal predicted points as support points for the new metamodels. A special selection process of new infill points was set up for the multi-objective case. The algorithm was used to find the optimal configurations of a linear induction motor (LIM) with respect to three antagonist criteria (mass, efficiency and developed force). The LIM was represented during the optimization process using a 2D FE model constructed using the commercial FEA software Opera2D. The optimization process can either be automatic or manual, hence being decoupled into several steps, represented by the algorithms iterations. Within a company's structure, this allows for a dialogue between the designer or the team addressing the optimization process and the specialists responsible with the development of the product. This is a common practice in the industry, as the case of the Renault Company, which employs a similar approach for addressing the crash and vibratory studies for the produced vehicles.

The latter presented method is also the most complex and integrates most of the author's personal contribution to the field. It consists of an adaptive method of optimization based on the use of Kriging metamodels and their interpolating feature. The original EGO algorithm was proposed in the early 2000s and is based on the use of a statistics-based criterion mixing the metamodels prediction and an error estimation of the Kriging metamodel. The goal of this

algorithm is to balance between the search for improving designs and the exploration of the design space, associated to the improvement of the metamodels global accuracy. A number of classical single-objective infill selection criteria have been reviewed. Several methods for handling the constraint functions were overviewed for the use within the EGO algorithm and their appropriation is analyzed both using analytical test functions and a classical electromagnetic test problem, consisting of the optimal sizing of a superconducting magnetic energy storage (SMES) device. The mechanism of the MEGO algorithm is sequential, allowing for the computation of a single infill design at a time. In order to fully benefit from the computational power available at hand, two structural modifications of the multi-objective MEGO algorithm have been proposed for generating a set of infill designs at each iteration of the algorithm, instead of a single one, as in the classical case. In consequence, an overall optimization time reduction was obtained by an order of about 4, when using a server configuration of two CPUs and eight cores.

The integration of metamodels into the optimization process shows very useful, allowing for handling optimal design problems of complex electrical devices or other products in general, with accurate analysis tools such as CAD/CAE software. In the following chapter, the optimization of complex systems by decomposition approaches will be addressed. The complex system is decomposed following his different sub-systems and components, and an optimization problem is defined for each component or sub-system. A coordination strategy manages the information exchanges between the optimizers of each sub-system/component. The employment of metamodel based design optimization approaches (MBDO) seems highly adapted for coping with such complex system design structures and will be addressed in the next chapter.

Chapter 3 Decomposition-based complex system design optimization

In the previous chapter, there were presented the different ways of addressing the optimization of electromagnetic devices using expensive simulation and analysis tools, by means of creating metamodels for these high fidelity models. Employing a more global vision, these devices represent only parts of more complex engineering systems. The optimization of such final systems or products, while accounting for all the functional relationships between all its components, represents a difficult task. This chapter addresses the optimization of complex engineering products or systems, by means of decomposition of the system to be designed and coordinated optimization of the different sub-systems and components of the complex system.

First of all, the benefits of decomposing a complex system are studied. The different single and multi-level decomposition strategies are then overviewed and the focus is set on the multi-level optimization approaches. Three most common categories of complex system decomposition approaches: model-based, discipline-based, and object-based with one most representative strategy from each category are then presented in detail. The model-based decomposition approach, Space Mapping (SM) and particularly a variant named Output Space Mapping (OSM) is presented through application to the optimal design of a safety-isolation transformer using the FE method and analytical modeling. The results are compared to those obtained using a similar approach, the MEGO algorithm presented in the previous chapter. The attention is directed next to the Collaborative Optimization (CO) strategy, belonging to the discipline-based decomposition category. The same safety-isolation transformer optimization benchmark is addressed using the CO strategy, based on the analytical model, and an efficiency improvement of the CO approach is proposed in order to allow the integration of the FE model into the optimization process with affordable computational cost.

A particular multi-level object-based decomposition strategy – Analytical Target Cascading (ATC) – which was found to comply very well with the hierarchical organization of the Alstom Transport Company is then presented in detail. A mathematical optimization problem is used to present the general ATC formulation and the different concepts associated with ATC. The convergence of the ATC strategy in the presence of both attainable and unattainable design targets is discussed. Some efficiency enhancements of the ATC strategy, based on the integration of metamodels within the hierarchy of the ATC optimization strategy, are developed. A multi-level optimization framework developed under Matlab® and implementing the ATC strategy with the presented enhancements was used for addressing an Alstom railway application consisting of the optimal design of a tramway with an on-board ultra-capacitor energy storage system (UC-ESS).

3.1 Why need to decompose a complex system?

Generally, the products designed by engineers are too complex to be handled holistically[16] by a person or even a team of engineers [ALL 08]. Products such as aircrafts, automobiles, trains, tramways or metros, to cite only a few, but the most representative examples of such complex systems, cannot be designed as a whole, due to panoply of reasons.

The design process of such complex systems implies combined decisions from a broad area of both technical and non-technical disciplines, from mechanics, hydraulics, thermic, vibration and acoustics, electrics and electronics, continuing with environmental impact, social and economic impact, and up to finance, human resources, marketing and management. Expressing all the needs and requirements from the above-cited domains in a general, global manner is unpractical, difficult and usually it is never realized.

The system to be designed is generally composed of different physical sub-systems, such as body-case, powertrain, traction system, heating and ventilation system, auxiliary systems, etc. Moreover, each of these sub-systems can be further seen as being composed of several different components, e.g. a tramway traction system is composed of motor, converter, transformer, gear-box, braking resistor, etc. Following the company's organization scheme, usually each sub-system design is handled by a different department of the company and the design of each component of a given sub-system is addressed by an engineer or a team of engineers with a given domain of expertise. Each team of engineers possesses a particular expertise and uses its specific engineering tools for the simulation, analysis and design of the product to be developed. Thus, the initial design task is split into several, smaller and more manageable design problems, allocated each to a different team of specialists which use their proper tools. The collaboration between all these specialists within the design cycle of a product is crucial and should be done in a coordinated manner in order to accomplish a competitive design.

The complex systems decomposition also helps the better understanding of the functional relationships between the different elements of the decomposed structure of the system to be optimally designed.

Dealing with several smaller optimization problems at each element of the decomposed structure, which are solved independently, it becomes possible to employ a different optimization method for solving each optimization problem of the hierarchy. One may assign the most appropriate optimization tool for each of the sub-system optimization problems. This way, while one sub-system problem having a smooth model is solved using a gradient-based algorithm such as SQP, other sub-problem having a lighter computation model with a-priori known multi-modality might be assigned a heuristic optimization algorithm, such as GA, PSO or Simulated Annealing, or if a heavy simulation code with inherent numerical noise is employed, a metamodel-based optimization algorithm might be used.

However, an important drawback of the decomposition-based design optimization methods consists in the additional computational cost related to the coordination strategy, which has the key role of ensuring the consistency between the different sub-systems of the complex system [GUA 11].

[16] The term "holistic" refers to addressing the system as a whole, rather than the sum of its different components. Hence, holistically handling a complex system implies addressing at the same time all the functional relationships between the different components of the system.

Therefore, for the above-cited and many more reasons, such complex systems require the use of a decomposition approach for their optimal design. In the following paragraph, the decomposition of a complex system will be addressed.

3.2 Complex system partitioning

The partitioning of a complex system can be done in a number of different ways, following different criteria. Choudhary et al. [CHO 05] suggested several main categories of complex system decomposition strategies existing in the literature: object-based, aspect or discipline-based and model-based decompositions. A fourth category, less current, a sequential or workflow-based decomposition strategy was equally suggested.

Following the object-based decomposition, the system is divided by physical components, e.g. for a railway traction system: motor, transformer, power electronics, auxiliary systems etc. The aspect or discipline-based decomposition implies dividing the system according to different disciplines or specialties, and is employed when multiple performance aspects of a device or system are assessed together. The model-based decomposition is based on the functional dependencies between the variables and the functions of a problem. The sequential decomposition addresses sub-problem division of problems using a workflow-based organization. In the electromagnetic and, at a larger extent, the electrical engineering domain, the optimal design of complex systems deals mainly with the former three decomposition strategies. The decomposition of the complex system following these three most-employed strategies is presented schematically in Figure 3.1.

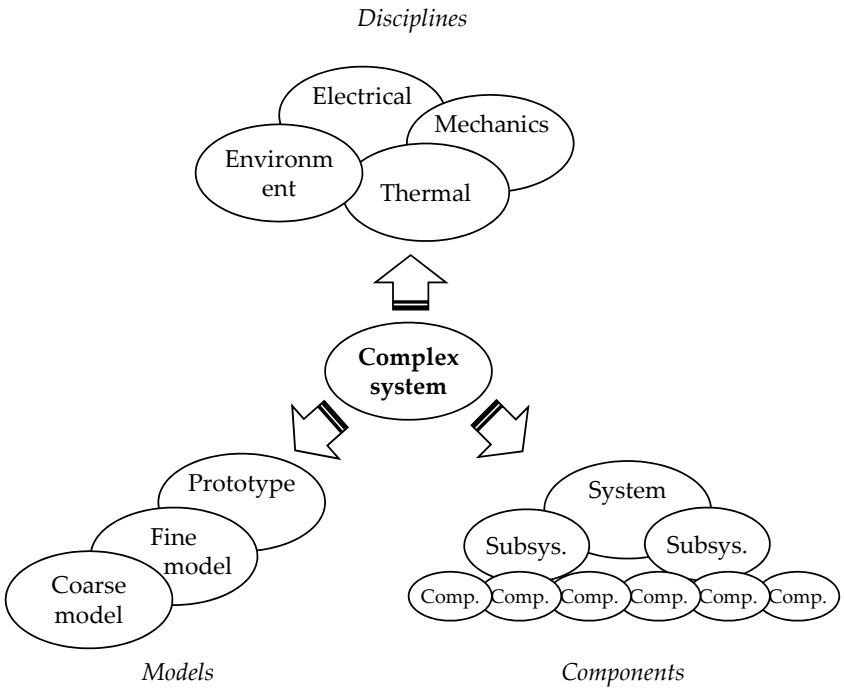


Figure 3.1 : Complex system decomposition

In industrial practice, the decomposition of a complex system is mainly done in accordance with the organization of the company whose complex system design is studied. Thus, in a company there might be several teams of engineers, each specialized in a field, such as electrical, mechanical, structural, thermal, etc. working at the development of a complex system. In this case, MDO approaches are well suited for this kind of company organization. Other companies, such as the case of Alstom Company, are organized hierarchically, where the design of a complex system (metro, tramway, train etc.) is the result of an assembly of the different constituent parts of the designed system. In this case, the teams of engineers are specialized and assigned each to address the design of one component of the complex system. Other departments or teams deal with the integration of the different designs into the complex system structure. For this later case, object-based complex system decomposition approaches are well-suited to handle such hierarchically structured designs. Moreover, this decomposition approach is also motivated by the organization of the company over several different production sites, each in charge of producing a component of the railway traction system.

Depending on the complexity level of the system, there are two types of formulations which can be employed for the optimization of a complex system: single-level respectively multi-level formulations. The difference between the two approaches consists in the way the optimization task is assigned and performed. For single-level methods, a unique optimizer manages the optimization task of the decomposed structure of the system. When using multi-level methods, the optimization task is distributed across the system, by employing a different optimizer for each element of the decomposed structure. Many strategies exist for coordinating the different optimizers engaged in the optimization of a complex system. While the methods from the former category have been studied for a long period of time and have already attained the maturity, the latter approaches are only in their youth period [DEP 07]; a lot of research is dedicated lately to the development, the improvement and the application of such strategies. In the following paragraphs, the two types of approaches will be briefly overviewed, presenting the differences and the particularities of each method belonging to these categories.

3.2.1 Single-level optimization strategies

In the decomposition-based design optimization context, the single-level methods refer to multidisciplinary design optimization methods (MDO) [CRA 94], [BAL 94], [AGT 10], [BAL 11a]. The single-level aspect refers to the fact that a single optimizer is used to manage the different interconnected disciplinary models. Using the MDO formulations, the problem is decomposed following different disciplines, more or less strongly interconnected. Depending on the level of coupling between the disciplines considered in the optimal design process, three main MDO methods exist in the literature: multidisciplinary feasible (MDF), individual discipline feasible (IDF) and all-at-once (AAO).

The MDF method is the most usual MDO method and it is also known as the all-in-one (AIO) method [BAL 11]. For each iteration of the system optimizer, a complete multi-disciplinary analysis is run. Hence, the feasibility of all disciplines is enforced at each iteration of the system optimizer. The coupling between disciplines is usually solved using a FPI method. MDF is also the most easy to implement and does not need to break down the system, being handled as a whole.

The IDF formulation reduces the computational effort of the optimizer by avoiding a complete multidisciplinary analysis at each system optimization iteration. Therefore, at each iteration, the different disciplines are individually feasible, but the coupling between them is not guaranteed during the optimization process. In fact, the consistency of the solution is guaranteed only at the end of the optimization process if convergence is achieved. The IDF formulation makes use of additional coupling variables for handling the coupling between disciplines. Breaking down the coupling between disciplines, the computational effort is reduced and furthermore, parallelizing the discipline evaluations is possible with IDF.

The AAO formulation goes further with the decomposition of the multi-disciplinary model, by breaking down any internal loop of a disciplinary model. The additional variables and the associated equality constraints increase the complexity of the optimization problem and might face convergence difficulties with classical optimization algorithms. With the AAO formulation, the design configuration is consistent only at the end of the optimization process, when the residuals of the additional equality constraints are null; the interdisciplinary and the disciplinary feasibilities are not ensured at intermediate iterations of the system optimization.

The single-level optimization strategies have been thoroughly studied for a long period of time and have already proven their capacities and limits. At the L2EP laboratory at Ecole Centrale de Lille, these techniques have made the object of several researches lately [KRE 08], [BEN 12b]. The comparison of the different single-level MDO methods on a safety-isolation transformer optimization benchmark was addressed by Ben-Ayed et al. in [BEN 12]. Thus, the single-level MDO strategies are only cited here for completion purpose and will not be studied herein. The main focus of this chapter of the thesis is set on the multi-level optimization strategies, which have not yet made proof of their full capacities and thus these will be analyzed in the following paragraphs.

3.2.2 Multi-level optimization strategies

Multi-level decomposition-based optimization strategies imply the partitioning of the complex system into different elements or sub-systems, displaced hierarchically onto several interconnected levels. The multi-level optimization strategies used with such decomposed structures employ a different optimizer for each sub-system of the hierarchy. Hence, the burden of the optimization is distributed along the different elements of the decomposed structure. The “hierarchical” attribute of a system partitioning refers to the functional dependency among the different elements of the architecture. A coordination strategy must be set in place in order to manage the information flow between the different elements of the decomposed structure. The role of the coordination strategy is to drive the different quantities exchanged between elements to agreement and obtain in the end a design which is consistent and optimal. The coordination strategy has equally the role of defining the order in which the different element optimizations are executed and passing the information from one element to another.

The decomposition of the system can be done in a number of different ways and the way it is decomposed imposes the selection of an appropriate optimal design strategy. As mentioned earlier, the model-based, discipline-based and the object-based decomposition strategies are the most frequently encountered in the industrial practice, especially the latter two, which are intended to comply with the structural organization of the company (see Figure 3.1).

3.2.2.1 Model-based complex system decomposition

The model-based decomposition of a complex system implies the existence of several models of the complex system or device, of different fidelity levels. A complex system or device can be represented, depending on the purpose of the study, using different types of modeling. Thus, in the industrial practice, during the preliminary phases of the design process, a more global vision of the system to be designed is required. This way, a simple response surface or a coarse model, supplying with the main trends or tendencies of the system's outputs can be used to represent the system to be designed. The advantage of such models consists in the fast obtaining of the system's main tendencies, in exchange for their lack of accuracy. However, during the later stages of an industrial design process, high fidelity simulations are employed by the engineers in order to analyze and validate the behavior of the system to be optimally designed. During the last two decades, with the important development of powerful computational resources, the use of CAD and CAE software has become very frequent in product design. These tools, as the FEA are often used in order to simulate and validate the potentially optimal designs. Such analysis usually requires an important amount of computational effort and a lot of evaluation time, which makes prohibitive the simulation of many potential configurations. Depending on the type of application, the accuracy of the results supplied by the simulations might not be sufficient and the design process can be extended even to the development of physical reduced-scale prototypes. It is obvious that such prototypes are very costly, time and effort demanding, thus a single or a much reduced number of prototypes should be developed. Techniques such as the Space Mapping (SM) are intended to address optimization problems dealing with several models of different fidelity of the same device, exploiting the synergies of the combined use of variable fidelity models of a system or device. The general representation of a complex system decomposition following several models of different accuracy is presented schematically in Figure 3.2. The model-based decomposition of the complex systems is organized under a hierarchical architecture, with a decreasing fidelity of models downwards the hierarchy.

The SM technique is a space projection method [BAN 94], [BAN 04]. It is a common approach for the optimization of systems represented through several models of different fidelity. This approach requires the use of at least two models of the same system, a fine and a coarse one, such as a FE model respectively an analytical or light-mesh FE model. The different fidelity optimizations are nested one into the other in a hierarchical architecture. This method was applied with success in the electromagnetic domain to the optimization of electromagnetic converters [BAN 94], [ENC 07]. Several different variants of the Space Mapping technique have been developed. Among these, the most common are the Aggressive Space Mapping (ASM) [BAN 95], the Output Space Mapping (OSM) [ENC 07], [ENC 08], [TRA 09] and the Manifold Mapping (MM) [ECH 08]. The first-mentioned approach (ASM) proposes a technique for the acceleration of the convergence of this strategy, leading to a reduced computational effort. However, ASM makes use of a complicated parameter extraction process, which is very sensitive and difficult to implement. The latter two variants, OSM and MM, avoid the delicate parameter extraction process and are therefore easily implementable. The OSM variant aligns iteratively the coarse model to the fine model by the use of a correcting vector of coefficients. The MM method uses a so-called manifold mapping function to modify the target values at each iteration.

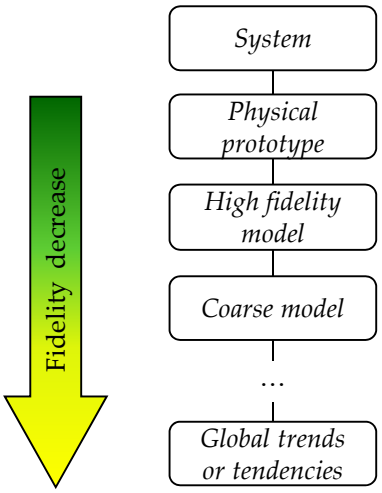


Figure 3.2 : Different models that can be used in model-based decomposition

3.2.2.2 Discipline-based complex system decomposition

The development of optimization strategies dealing with multi-level discipline-based decomposed systems came as a solution for the limits faced by the single-level MDO strategies. When the multidisciplinary design problem becomes very complex, with many disciplinary aspects involved in the optimal design process and very strong interdisciplinary connections, as the case of most industrial problems, the classical single-level MDO formulations cannot manage such problems. Several multi-level MDO strategies have been developed during the last two decades. The most common is the CO strategy proposed by Braun and Kroo in [BRA 96]. The method proposes a higher autonomy of the different sub-systems of the decomposed structure in order to satisfy the interdisciplinary compatibility constraints. Hence, giving the disciplinary autonomy, this method allows the intervention of the specialists of a discipline to the corresponding sub-system, without any regard to the other disciplines involved in the optimal design process. Another multi-level MDO method is the Concurrent Subspace Optimization (CSSO), proposed earlier by Sobieszczanski-Sobieski in 1988 in [SOB 88] and developed at the NASA Langley Research Center. As with the previous CO method, the CSSO strategy also advocates for the disciplinary autonomy. A system level optimizer manages the coupling between disciplines and guides the optimization process towards a consistent optimal solution. In each disciplinary sub-system, an approximation of the coupling variables is used in order to estimate their influence on the objective and constraint functions. During the concurrent optimizations of the sub-systems, the shared variables are considered as constants. The approximation of the coupling variables is obtained through the use of metamodels or approximate models. Bi-level system synthesis (BLISS) is another multi-level MDO strategy proposed also by Sobieszczanski-Sobieski a decade after CSSO, in 1998 in [SOB 98], and also developed at the NASA Langley Research Center. As with the previous two methods, a system optimizer is in charge with the coordination of the different sub-system optimization processes. To improve the interdisciplinary consistency constraint satisfaction, the system level also employs a multi-disciplinary analysis to better estimate the values of the shared and local disciplinary variables. The BLISS method employs a gradient-based approach to optimize the contributions of

the shared and discipline related design variables to the objective function. An improved version of this method, named BLISS2000 employs metamodels for the different disciplinary analysis, thus improving the computational efficiency of the method [SOB 00a]. A similar method for multi-level MDO problems, named DIVE (Disciplinary Interaction Variable Elimination Approach) has been developed by Parte et al. in [PAR 09]. The DIVE method makes use of metamodels for the different disciplinary optimizations. This method proposes a complete autonomy of individual disciplines to handle their disciplinary variables.

A common feature of all multi-level discipline-based complex system decomposed optimization strategies is that they are all bi-level. Since there is no priority between disciplines, all disciplinary sub-systems are displaced on the same level, being evaluated concurrently. A system level optimizer is in charge with coordinating the different sub-system level disciplinary optimizers.

The representation of multi-level complex system decomposition into the different disciplines involved is presented schematically in Figure 3.3.

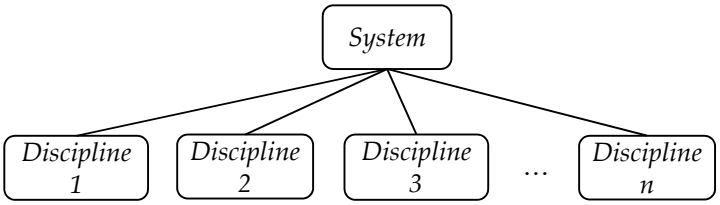


Figure 3.3 : Discipline-based complex system decomposition

3.2.2.3 Object-based (physical) complex system decomposition

The object-based decomposition addresses the complex system viewed as the set of its constituent physical sub-systems and components. Therefore, the complex system can be decomposed into its corresponding sub-systems, at their turn the sub-systems can be further decomposed into components and the decomposition can continue down to the elementary components of the analyzed structure. All the components and sub-systems function interconnected together, producing the functionalities of the system. The optimization strategies addressing object-based decomposed systems have the role of optimally designing the different components of the decomposed structure, while managing the relationships between all these components and sub-systems, ensuring thus the consistency of the system as a whole. Based on the level of desired decomposition, the structure can be partitioned into several elements displaced hierarchically over two or more levels. For each element of the hierarchy, an optimization problem is formulated. Hence, as with the discipline-based decomposition methods, the computational burden of the optimization is distributed across the hierarchical structure. Compared to the disciplinary decomposition of a system, presented in paragraph 3.2.2.2, where no priority of evaluation could be established between disciplines and thus the decomposition was done on a bi-level basis, the object-based decomposition of systems allows a generalization of the decomposition to as many levels as required. In such a decomposed structure, an element at an intermediate level of the hierarchy directly depends only on the sub-system which it belongs to, called parent element and which is a unique element and also on its constituent elements, situated at the inferior level and which are considered its children elements. Among the strategies addressing object-based decomposed

complex systems, two are the most representative, the Analytical Target Cascading (ATC), or simply Target Cascading (TC), and the Augmented Lagrangian Coordination (ALC). The ATC strategy introduced by Michelena et al. in the early 2000s [MIC 99], [MIC 03], [KIM 01] was proposed as a technique for formalizing the industrial product development process. It consists of propagating the general targets of the product specification to all elements across the decomposed hierarchy. The different elements of the hierarchy aim to attain the imposed targets by modifying their local design variables. Therefore, the targets are propagated downwards to the lowermost elements of the hierarchy and their responses are re-balanced upwards to the top of the hierarchy. Different coordination techniques are employed to ensure the consistency and the convergence to the optimal design. According to the ATC strategy, an element of the hierarchy exchanges information only with its parent and children elements from the adjacent levels. No direct horizontal coupling between elements of the same level or from different levels, other than its parent and children elements is allowed by the ATC formulation. Nonetheless, in practice, object-based decomposed design problems presenting stronger couplings between the elements of the hierarchy may exist. To address such problems, a modified version of ATC, named ALC, presenting the same structure as that of ATC was proposed by Tosserams et al. [TOS 07], [TOS 08]. This version allows coupling between different elements across the decomposed structure, representing thus a generalization to non-hierarchical organized structures of ATC. The coordination method employed by ALC uses system-wide functions and basis it's mechanism on the Lagrangian duality theory to ensure the convergence of the optimization process and the consistency of the design.

The object-based decomposition of a complex system into several elements organized hierarchically is presented in Figure 3.4.

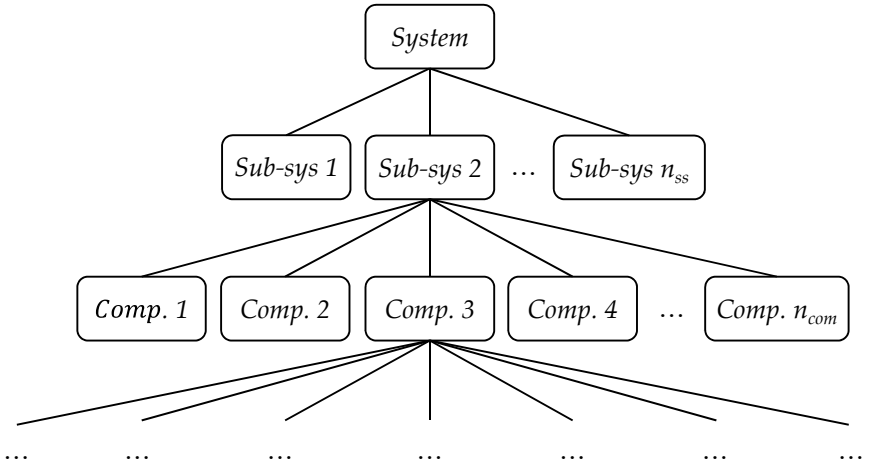
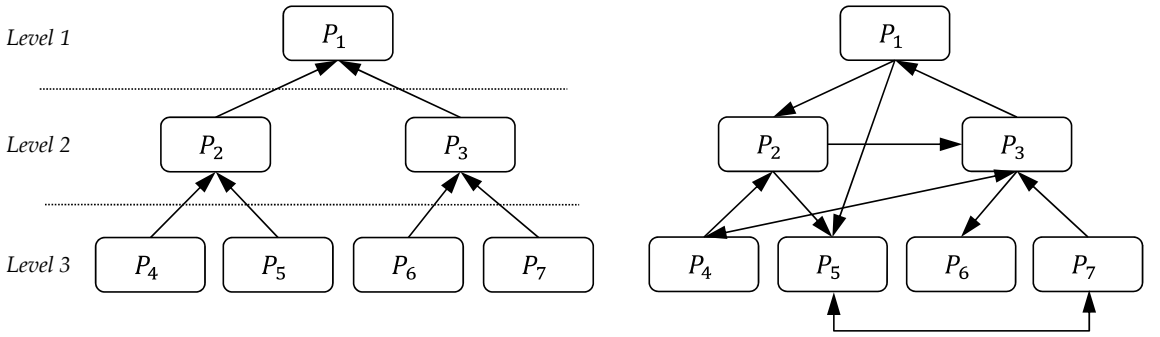


Figure 3.4 : Object-based (physical) complex system decomposition

An example of a complex system decomposed into its seven constituent elements is presented graphically in Figure 3.5. The first case, presented in Figure 3.5a shows a purely hierarchical decomposition of the system following three levels. Figure 3.5b shows the general case of non-hierarchical multi-level decomposition of the system, where coupling between any elements of the decomposed structure is allowed.



a) Hierarchical decomposition b) General non-hierarchical decomposition

Figure 3.5 : Example of complex system hierarchically and non-hierarchically decomposed

3.3 Output Space Mapping (OSM) multi-model optimization strategy

The Output Space Mapping (OSM) strategy is a variant of the Space Mapping (SM) method. It involves the use of two models of the device to be conceived, a fine (e.g. FE model) and a coarse one (e.g. analytical, light-mesh FE model). The OSM method bases its principle on the systematical alignment of the coarse model to the fine one, consisting of two optimization loops – a high fidelity and a low fidelity one, with a hierarchical connection. The low fidelity optimization loop makes use of a classical optimization process (such as SQP) with the coarse model which benefits from a fast evaluation. Its role is to supply the upper high fidelity loop with designs to be evaluated and validated using the fine model. In the upper loop, there is no formal optimization problem; it consists only of a single-passage step of fine model evaluation of the design previously found by the lower loop. Once the design is evaluated using the fine model, a vector of correcting coefficients is calculated in order to align the coarse model to the fine one. The process continues until the two models, fine and coarse are aligned at the site of the optimum, within a prescribed tolerance.

The OSM technique and the SM strategy at a larger extent were originally intended for solving bi-level model-based decomposed problems. A three level OSM strategy has been proposed by Ben-Ayed et al. [BEN 12a] for the optimization of a safety-isolation transformer and a linear induction motor (LIM) using 3D FE models as fine models. An analytical model and a Kriging metamodel were considered as coarse models for the transformer, respectively the LIM device. For the transformer application, a 3D FE model with a light mesh was considered as intermediate model, having an accuracy level situated between the fine and the coarse model. A 2D FE model with a light mesh was selected as intermediate model for the LIM application case. A significant overall optimization time reduction was indicated for both applications, due to the use of the intermediate models. A generalization of the OSM strategy for application to problems consisting of n models of different fidelity levels has been proposed by Ben-Ayed et al. in [BEN 11a]. Practical details of the implementation of these generalized techniques can be found in [BEN 12b] and [GON 11a]. However, in practical applications a maximum number of three fidelity levels seem sufficient for the multi-level model configuration of the OSM strategy.

The information flow between the high fidelity and the low fidelity loop of the OSM strategy is presented graphically in Figure 3.6.

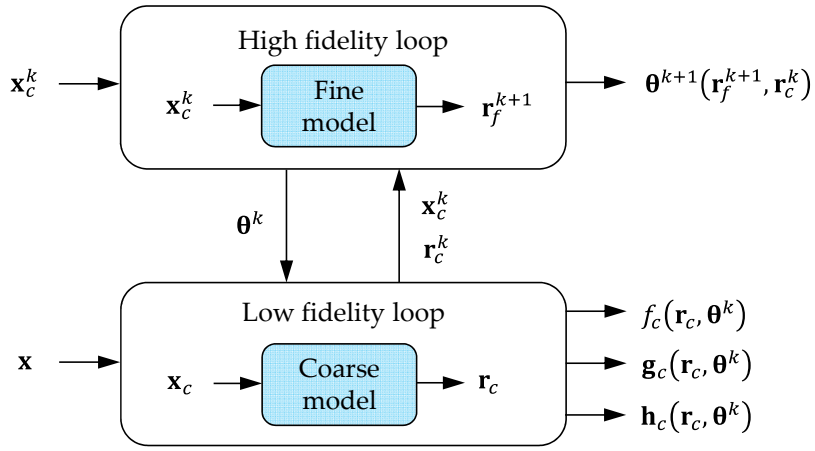


Figure 3.6 : Output Space Mapping (OSM) information flow

Upper loop (high fidelity loop)

At the first iteration of the optimization process, the vector of correcting coefficients $\boldsymbol{\theta}^1$ is initialized to the unit vector $\mathbf{1}$.

$$\boldsymbol{\theta}^1 = \mathbf{1} = [1,1,\cdots,1] \tag{3.1}$$

At any iteration k, other than the first one, the correcting coefficients vector $\boldsymbol{\theta}^{k+1}$ is adapted according to the expression in (3.2).

$$\boldsymbol{\theta}^{k+1} = \mathbf{r}_f^{k+1} \circ \frac{1}{\mathbf{r}_c^k} \tag{3.2}$$

Lower loop (low fidelity loop)

At a given iteration k of the upper loop, the lower loop receives the values of the coefficients of the correcting vector $\boldsymbol{\theta}^k$. An optimization process is then launched using a gradient-based optimization algorithm (such as SQP) on the coarse model whose outputs are adjusted by the vector $\boldsymbol{\theta}^k$. The mathematical formulation of the lower loop optimization problem is expressed in (3.3).

$$\begin{aligned} \underset{\mathbf{x}}{\text{Minimize}} \quad & f_c(\mathbf{r}_c(\mathbf{x}), \boldsymbol{\theta}^k) = f_c(\mathbf{r}_c(\mathbf{x}) \circ \boldsymbol{\theta}^k) \\ \text{such as} \quad & g_c(\mathbf{r}_c, \boldsymbol{\theta}^k) \leq 0 \\ & h_c(\mathbf{r}_c, \boldsymbol{\theta}^k) = 0 \\ \text{where} \quad & \mathbf{r}_c = \mathbf{r}_c(\mathbf{x}) \end{aligned} \tag{3.3}$$

The optimal output vector of the adjusted coarse model $\mathbf{r}_c^k$, as well as the corresponding optimal design vector $\mathbf{x}_c^k$ are re-balanced to the upper loop for validation with the fine model, in order for it to calculate a new correcting vector $\boldsymbol{\theta}^{k+1}$.

The optimization process is completed when the two models, fine and coarse are aligned on their output variables within a prescribed tolerance value ε. The expression of the stopping criterion is given in (3.4).

$$\left\| \frac{\mathbf{r}_f^{k+1} - \mathbf{r}_c^k}{\mathbf{r}_f^{k+1}} \right\|_\infty \leq \varepsilon \tag{3.4}$$

where $\|\cdot\|_\infty$ represents the infinite norm or the l_∞-norm.

Giving the dual model character of this approach, a number of similarities can be found between the OSM approach and the MBDO approaches presented in the previous chapter. While in the present case a coarse model is systematically improved by aligning it to the fine one, using the vector of correcting coefficients $\boldsymbol{\theta}$, the MBDO approaches of the previous chapter propose the use of a metamodel which is progressively improved by the addition of new support points. At a certain extent, the MBDO approaches can therefore be perceived as multi-level model-based decomposition optimization strategies. This entitles thus a comparison between two similar approaches, OSM strategy presented here and the MEGO strategy developed in the previous chapter. Using the same application of the safety-isolation transformer optimization problem addressed by Ben-Ayed et al. in [BEN 12], a preliminary comparison between the OSM and the MEGO algorithm has been presented in [BEN 11] on a single-objective optimization basis. The results showed an advantage of the OSM method over the MEGO approach, which was mainly due to the good quality of the analytical model used as coarse model by OSM. For a complete comparison between the two presented approaches, a multi-objective comparison was equally addressed on the same safety-isolation transformer benchmark problem by Berbecea et al. in [BER 12]. These comparisons allowed identifying the different similarities and differences between the two approaches both within a single and a multi-objective framework, as well as drawing useful guidelines for the future improvement of both methods.

Mathematical example

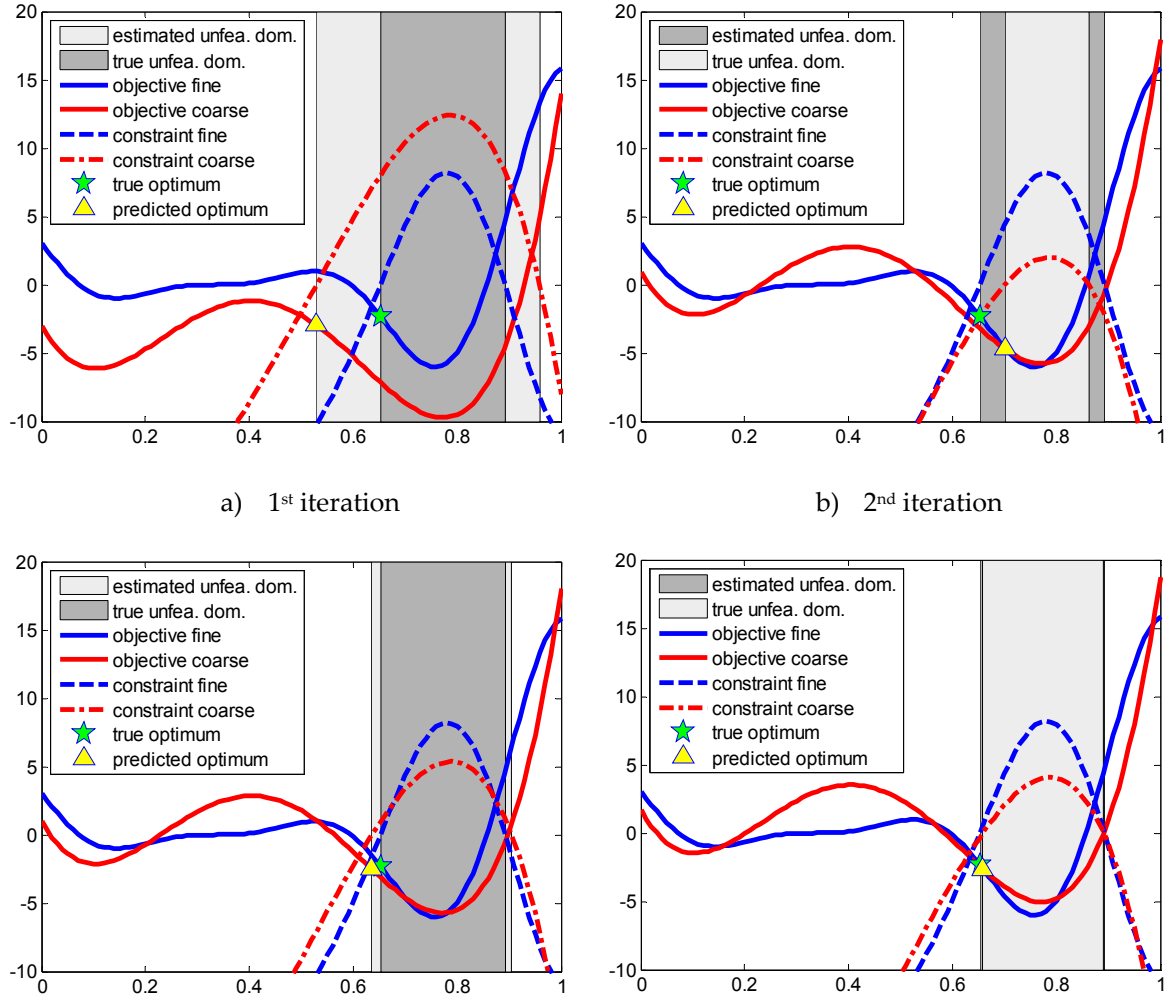
In order to illustrate the mechanism of the OSM strategy, a simple mathematical example problem has been constructed starting from Forrester et al.'s test function found in [FOR 09], to which a 2nd order polynomial function has been added. Two polynomials, a four-degree and a two-degree one, respectively have been constructed and are considered as coarse models. The mathematical expressions of the fine and the coarse models are given in (3.5).

$$\begin{aligned} f_f(x) &= (6x-2)^2 \sin(12x-4) \\ g_f(x) &= -75x^2 + 145x - 65 - f_f(x) \\ f_c(x) &= 532x^4 - 908x^3 + 462x^2 - 69x - 3 \\ g_c(x) &= -40x^2 + 82x - 35 - f_c(x) \end{aligned} \tag{3.5}$$

where $x \in [0,1]$.

The optimization process has been launched using the OSM optimization technique. The optimization process converged after only 8 OSM iterations, therefore 8 total fine model ($f_f(x)$

respectively $g_f(x)$) evaluations. The evolution of the adjusted coarse objective and constraint functions during the OSM optimization process is presented in Figure 3.7 for the first 3 iterations and the last iteration of the optimization process.



a) 1st iteration

b) 2nd iteration

c) 3rd iteration

d) 8th (last) iteration

Figure 3.7 : OSM optimal results for the analytical test-case

The true optimum of the optimization problem (given by the fine models of the objective and constraint functions), represented by a green star, is situated on the border of the feasibility constraint function, as can be seen from Figure 3.7a. At the beginning of the OSM optimization process, an important difference can be remarked between the fine and the coarse models of both objective and constraint function, reflected in the difference between the true and the predicted optimal designs, represented by a green star and a yellow triangle, respectively. Along the OSM iterations, the coarse models of both objective and constraint function are aligned progressively in the region which fosters the true optimum of the problem, as can be seen from Figure 3.7a to Figure 3.7d, ending up by aligning the fine and the coarse models at the spot of the true optimum. From the four representations of Figure 3.7 one can remark a certain global discrepancy between the fine and the coarse models of both objective and constraint functions. Nevertheless, the global trend of

the fine models is successfully captured by the coarse models, representing the key to the success of the optimization process employing the OSM algorithm.

3.4 Collaborative Optimization (CO) multi-discipline optimization strategy

The Collaborative Optimization (CO) is a class of bi-level multi-disciplinary optimization strategies, consisting of a superior level (system level) and a lower level (disciplinary level). It was first introduced in 1996 by Braun in [BRA 96]. The CO architecture promotes disciplinary autonomy while maintaining interdisciplinary compatibility [BRA 96a], [KRO 00]. The first applications of CO addressed multi-disciplinary optimal design problems from the aerospace and aeronautical industry for solving aircraft optimal design applications [BRA 96], [MOO 96], [ALL 06a], [WAN 09]. A number of similarities exist between CO and another decomposition-based optimization approach, ATC presented in paragraph 3.5. Allison et al. [ALL 05] review the terminologies used by CO and ATC and clarify the differences between the two methods by application to a structural optimal design problem and an electric water pump design problem.

The structure and the information flow between the different elements of the architecture of the CO process are presented graphically in Figure 3.8.

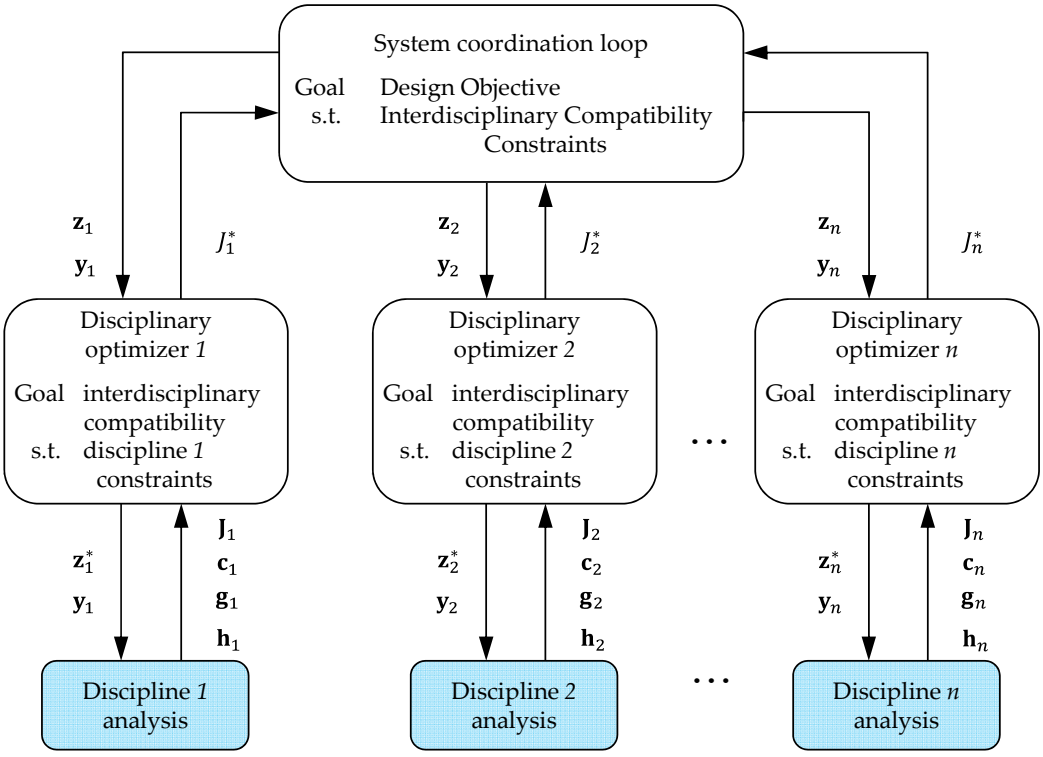


Figure 3.8 : Structure and information flow of the collaborative optimization (CO) strategy

In the CO structure presented in Figure 3.8, $\mathbf{z}_i$ represent the system level design variables for the i-th sub-system, $\mathbf{y}_i$ represent the interdisciplinary design variables relative to the i-th sub-system, $\mathbf{z}_i^*$ represent the local copies of $\mathbf{z}_i$, J_i represents the objective function value of the i-th sub-system, J_i^* is

the *i*-th sub-system optimal objective function value issued from the disciplinary optimization, $\mathbf{c}_i$ represent the interdisciplinary coupling variables which are outputs of the *i*-th discipline, $\mathbf{g}_i$ are the inequality constraints specific to the *i*-th sub-system, and $\mathbf{h}_i$ are the equality constraints specific to the *i*-th sub-system.

3.4.1 Basic mathematical formulation of CO

A different optimization problem is formulated at the system and the sub-systems from the discipline level of the bi-level CO hierarchy.

The formulation of the system level optimization problem is expressed in (3.6).

$$\begin{aligned} &\underset{\mathbf{y},\mathbf{z}}{\text{Minimize}} && f(\mathbf{y},\mathbf{z}) \\ &\text{subject to} && \boldsymbol{J}^* = J_i^*\left(\mathbf{z}^*,\mathbf{z},\mathbf{y}_i,\mathbf{c}_i(\mathbf{y}_i,\mathbf{z}_i^*)\right) = \mathbf{0} \quad \forall i \in [1,n_{ss}] \end{aligned} \tag{3.6}$$

where $\mathbf{y}$ is the vector of all interdisciplinary coupling variables, $\mathbf{z}$ is the vector of all design variables, J_i^* represents the optimal objective function value found by the *i*-th sub-system, and n_{ss} is the number of sub-systems of the hierarchy.

The role of the system level optimization loop is to coordinate the different disciplinary optimization problems situated at the lower disciplinary level. At the system level, no formal analysis is run; the goal of the system optimization problem is to minimize the global objective function of the problem. Hence, no specific local inequality or equality constraints are expressed here. The system level design variables are divided into the global design variables $\mathbf{z}$ and the interdisciplinary coupling variables $\mathbf{y}$, which are shared between the different disciplines involved in the optimal design process. The consistency between the different disciplines involved in the optimal design process is ensured by the interdisciplinary constraints J_i^*.

The formulation of sub-system problem of a discipline *i* is expressed in (3.7).

$$\begin{aligned} &\underset{\mathbf{z}_i^*}{\text{Minimize}} && J_i\left(\mathbf{z}_i,\mathbf{z}_i^*,\mathbf{y}_i,\mathbf{c}_i(\mathbf{y}_i,\mathbf{z}_i^*)\right) = \|\mathbf{z}_i - \mathbf{z}_i^*\|_2^2 + \left\|\mathbf{y}_{ij} - \mathbf{c}_{ij}(\mathbf{y}_i,\mathbf{z}_i^*)\right\|_2^2 \\ &\text{subject to} && \mathbf{g}_i(\mathbf{y}_i,\mathbf{z}_i^*) \le \mathbf{0} \\ & && \mathbf{h}_i(\mathbf{y}_i,\mathbf{z}_i^*) = \mathbf{0} \end{aligned} \tag{3.7}$$

where $\mathbf{y}_{ij}$ represent the coupling variables which are outputs of the *i*-th sub-system and inputs to the *j*-th one, $\mathbf{g}_i$ represent the local inequality constraints specific to the *i*-th discipline, $\mathbf{h}_i$ represent the local equality constraints specific to the *i*-th discipline.

Each sub-system optimization problem of the CO multi-disciplinary hierarchy acts on its local design variables in order to find an agreement with the other sub-systems upon the coupling variables, while respecting the local constraints specific to the discipline.

The multi-level CO formulation is intended to comply very well with companies which are organized into several departments, each being in charge with addressing a disciplinary aspect of the system to be designed. This formulation allows the different discipline specialists or team of specialists of the company to work on the sub-system of their expertise without any regard and in parallel to the other disciplinary sub-systems.

3.4.2 Coordination of the CO process

The coordination of the CO process is realized by the system level optimization problem, by the management of the interdisciplinary consistency constraints. At convergence, the interdisciplinary constraints must be active in order to have consistency between the different disciplines.

A known issue of the CO process is related to its non-robustness due to the fact that it faces instability at the convergence [BAL 11]. In addition, the CO process was reported a slow convergence rate, even non-convergence, due to the interdisciplinary compatibility constraints expressed as equality constraints in (3.6). The potential non-smoothness and the noise in the disciplinary analysis models of the sub-system level, combined with the equality constraints introduce computational difficulties [ALE 00], [KRO 00], [ZAD 09].

A certain level of inconsistency must be accepted with the system-level problem in order for the process to be able to converge to an optimum. For that, the interdisciplinary equality constraints may be relaxed [ALE 00], [ALL 05], by transformation into inequality constraints, the system-level optimization problem taking the expression (3.8).

$$\begin{aligned} &\underset{\mathbf{y},\mathbf{z}}{\text{Minimize}} \quad f(\mathbf{y},\mathbf{z}) \\ &\text{subject to} \quad \boldsymbol{J}^* = J_i^*\left(\mathbf{z}^*,\mathbf{z},\mathbf{y}_i,\mathbf{c}_i(\mathbf{y}_i,\mathbf{z}_i^*)\right) \le \boldsymbol{\varepsilon} \qquad \forall i \in [1,n_{ss}] \end{aligned} \tag{3.8}$$

where $\boldsymbol{\varepsilon}$ is an inconsistency tolerance vector representing the allowed level of inconsistency between the different disciplines.

Using the formulation in (3.8), interdisciplinary inconsistency is allowed during system-level iterations. A loose-enough tolerance must be accepted in order for the optimization process to converge. However, a very loose tolerance value will cause the obtained design to be physically inconsistent. Thus, a variable-tolerance strategy might be used with CO, starting the system level iterative process with a higher value of $\boldsymbol{\varepsilon}$ and reducing it towards the end of the optimization process, in order to ensure consistency between disciplines.

3.4.3 Existing efficiency enhancements of the CO formulation

As with most of the decomposition-based complex system optimization strategies, the main issues of the CO strategy consist of the computational difficulties encountered in the coordination process of the different quantities exchanged between the diverse elements of its hierarchy [BRA 96a], [KRO 00], [ALE 00], [MIN 11]. A number of studies addressed these coordination issues and tried, by different means, to improve the convergence of the coordination process. In [LI 08], Li et al. identified and made use of some geometric properties of the CO process. The inconsistency between two sub-systems is seen as the geometrical distance between two optimum points of the two sub-systems. Instead of the classical fixed coefficients, a variable tolerance method for the relaxed consistency constraints was set in place in order to increase the convergence of the CO process.

The iterations of the CO process required for the interdisciplinary coordination convergence and the increased number of variables at each discipline sub-system problem following the CO decomposition strategy augment the computational complexity for the CO optimization process

compared to the non-decomposed problem. To alleviate the burden of the accurate but heavy computations of the disciplinary analysis, Sobieski and Kroo [SOB 00] and Jang et al. [JAN 05] proposed to replace the disciplinary simulation models by metamodels within the CO framework. A combination of neural networks and Kriging metamodels are used in [JAN 05] both in single-objective and multi-objective CO optimization context as substitute for the time consuming disciplinary simulations. To improve the accuracy of the metamodels prediction, these are improved progressively by additional support points along the iterations of the CO process. A similar approach was proposed by Zhao and Cui in [ZHA 07] and analyzed this time the combined use of an optimal LHS and a RBF metamodel to alleviate the long computations of the different disciplinary simulations. As within the approach proposed by [JAN 05], in their implementation the metamodels are also improved progressively along the iterations of the CO process in order to increase the prediction accuracy of RBF metamodels towards the final design. The LHS was used to successfully generate a feasible initial start point for the CO process. An important computational reduction was reported for the application of their implementation to solving the optimal design problem of a speed reducer. Another metamodel-based CO framework has been proposed recently by Zadeh et al. in [ZAD 09]. In order to improve the computational efficiency of the CO multi-disciplinary strategy, the authors propose the integration of metamodels within the disciplinary level of the CO structure to alleviate the computational burden generated by the heavy disciplinary analysis model. An important overall optimization time reduction is obtained for a cantilever beam test problem, with practically the same accuracy level of the results. Another study addressing the metamodel integration within the CO strategy was addressed by Wang et al. in [WAN 09] this time for the pop-up attack planning problem of a tactical aircraft. The response surface-based CO strategy proposed dealt with different laws of the tactical aircraft's flight regime, similar to the different disciplines accounted for by a classical MDO problem.

The classical CO strategy employs a nested formulation, meaning that for each iteration of the system-level optimization loop, a complete sub-system optimization is performed. This formulation implies an important number of disciplinary model evaluations of the sub-system level, which are usually time expensive simulation models. In order to reduce the computational cost of the traditional CO formulation, an Enhanced Collaborative Optimization (ECO) method was proposed by Roth et al. in [ROT 08], which eliminates the need of a complete sub-system optimization for each system iteration. This ECO formulation employs an augmented Lagrangian relaxation for the handling of the interdisciplinary constraints and makes use at each disciplinary optimization of an approximation of the other disciplines constraint functions. An important computational reduction was reported through application of the ECO formulation to both analytical test-problems and an aircraft design problem [ROT 08a]. An enhancement of ECO, employing Kriging metamodels for the approximation of the constraint functions of the other disciplines in the optimization process of a disciplinary sub-system was proposed by Xiao et al. in [XIA 10].

The integration of the multi-criteria feature within the CO process was presented by Rabeau et al. in [RAB 07], which was implemented under their COSMOS optimization platform. Their approach, based on the conjoint use of the classical CO strategy and multi-objective evolutionary algorithms for finding the non-dominated Pareto solutions of each discipline, allows finding a set of compromise solutions between the different disciplines involved in the optimization process. The set of compromise solutions comes in exchange for the inherent additional computational

complexity added by the classical evolutionary algorithms which require a large number of disciplinary model evaluations.

Combining the features of traditional CO and BLISS, Zhao and Cui recently proposed in [ZHA 11] a new multi-level MDO strategy entitled BLISCO, which uses the collaborative mechanism of classical CO and replaces the sub-system objective way for handling the compatibility constraints with the sum of coupled outputs of the disciplines. Two engineering problems, a speed reducer optimization and an electronic packaging system, were used for testing and validating the proposed method and their algorithm implementation was coupled to the commercial optimization software iSIGHT™.

3.4.4 Efficiency enhancement of the CO process by integration of the Output Space Mapping (OSM) technique

The CO process performs a multidisciplinary optimization of complex systems or devices. However, the integration of accurate simulation models into the CO process cannot be done without a consequent computational burden increase, mainly due to the nested character of the optimization loops of the CO process. In the upper, system-level optimization loop, each system-level objective function evaluation implies one optimization run of each disciplinary model from the lower level of the CO structure. When the disciplinary models employed are fast-evaluation models, such as analytical models – as the case of the isolation transformer benchmark – or approximate models, such as response surfaces, the optimization can be run at reasonable computational cost, regardless of the numerous calls to the disciplinary models within the CO process. Integrating the accuracy offered by simulation models, such as FEM, into the CO multidisciplinary optimization process by simply replacing the call to the fast-evaluation models with a call to the simulation models is injudicious.

A lot of research effort has been directed towards the integration of expensive simulation models within the disciplinary optimization loops of the CO process. As mentioned previously in paragraph 3.4.3, the most common way of reducing the computational burden introduced by the simulation models consists of creating and using metamodels of the expensive simulation models. In this paragraph, a novel idea is introduced, which is based on the existence of two different levels of modeling for the same device. The proposed idea consists in implementing a space-mapping technique for solving the disciplinary optimizations from the lower level of the CO structure of the isolation transformer benchmark problem. The implementation of this idea is based on the existence of two distinct models, an analytical and a FE model for each discipline involved in the simulation of the transformer. The OSM technique presented in paragraph 3.3 has been proven a good performance for solving the AIO optimization problem of the isolation transformer [BEN 11], [BER 12]. Using the OSM technique, the convergence of the AIO transformer optimization has been attained with a much reduced number of FEM calls. Thus, the OSM technique is proposed here for solving each disciplinary optimization problem of the isolation transformer. The idea fostered in this paragraph has been proposed for presentation in [BER 12a]. The bi-level structure of the CO formulation implementing the OSM technique (CO-OSM) for the disciplinary optimization is presented in Figure 3.9.

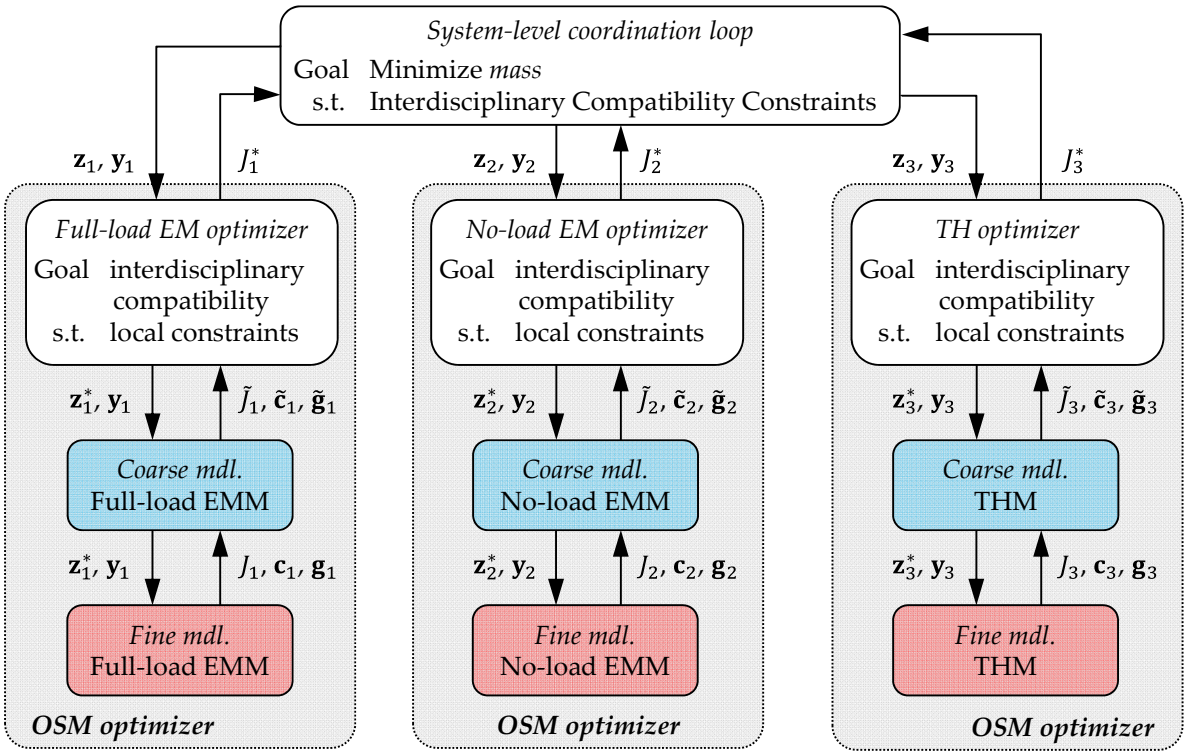


Figure 3.9 : Bi-level CO structure implementing the OSM technique for the optimization of the disciplinary models of the isolation transformer

The three fine models in Figure 3.9 represent the different FE models of the transformer (full-load and no-load electromagnetic models and the thermal model). A single analytical model, integrating both the electromagnetic and the thermal aspect is used for simulating the transformer. However, the different equations corresponding to the full-load EM model, no-load EM model and the thermal model have been identified and separated, in order to form three distinct disciplinary models. In Figure 3.9, J_i represents the objective function value of the i-th discipline calculated using the FE model, $\mathbf{c}_i$ is the vector of coupling variables which are outputs of the i-th discipline and calculated by the FE model and $\mathbf{g}_i$ is the vector of discipline-specific inequality constraints. The variables presenting the "$\tilde{}$" sign suggest the fact that their values are calculated this time using the corrected analytical model within the OSM optimization loop.

Within the disciplinary optimization using the OSM technique, the FE disciplinary model is used for correcting the outputs of the analytical model. Hence, the values of the variables received by the system-level optimization loop from the disciplinary optimization loops are those computed using the corrected analytical models. The implementation of this idea however requires further investigations.

3.5 Analytical Target Cascading (ATC) multi-component optimization strategy

The Analytical Target Cascading (ATC) or more simply Target Cascading (TC) is a multi-level decomposition-based complex system design optimization strategy introduced by a team of research scientists from the Mechanical Department at the University of Michigan, USA in the early 2000s as a technique of formalization of the industrial product development process [MIC 99a]. It consists of propagating the general targets of the product specification to all elements across the decomposed hierarchy. The different elements of the hierarchy aim to attain the imposed targets by modifying their local design variables. Thus, the targets are propagated downwards to the lowermost elements of the hierarchy and their responses are re-balanced upwards to the top of the hierarchy. Different coordination techniques are employed to ensure the consistency and the convergence to the optimal design. According to the ATC strategy, an element of the hierarchy exchanges information only with its parent and children elements from the adjacent levels. No direct horizontal coupling between elements of the same level or from different levels, other than its parent and children elements is allowed by the ATC formulation.

The first applications of ATC addressed the automobile industry, for the optimal design of automotive vehicles [KIM 01], [KIM 03], [KIM 03a]. Later on, the application of the ATC optimization strategy was extended to the aircraft design domain, being employed for the optimization of an aircraft design [ALL 06a]. Also, this method was applied with success not just for the optimization of one product, but also a family of products, e.g. design of a family of automobiles [KOK 02], design of a family of aircrafts [ALL 06], design of a family of racecars [FER 09]. A wide range of applications, such as simulation-based building design [CHO 05], structural design problems [MIC 06], and electrical water pump design [ALL 08], [ALL 09] were equally addressed using the ATC formulation. A multi-disciplinary design problem involving decision both from engineering and marketing domains was addressed in [MIC 05]. At the L2EP laboratory at Ecole Centrale de Lille, previous works of Moussouni-Messad and Kreuawan carried on the optimal design of railway systems [MOU 09a], [KRE 08].

The ATC optimization strategy was initially conceived as a hierarchical construction. Nonetheless, non-hierarchical complex system decompositions can be fostered by the ATC architecture. The integration of MDO problems, which deal with interconnected models from several disciplines, thus with no obvious hierarchy between disciplines, was addressed by Tosserams et al. in [TOS 08a]. To increase the flexibility of the ATC formulation, they proposed two extensions of the traditional ATC formulation to integrate the MDO structure within the ATC strategy for the design optimization of a supersonic business jet.

Most of the real engineering design problems naturally contain design variables which present a discrete nature, e.g. number of conductors, standard wire sections, number of electric circuits, etc. Usually, these variables are accounted for as continuous within the ATC process. Michalek and Papalambros proposed in [MIC 06] a modified version of ATC, integrating a classical discrete optimization algorithm, Branch-and-Bound (BB), which they call BB-ATC, for addressing a mixed-integer nonlinear structural design problem. However, this adds a considerable computational burden to the ATC optimization process, and it is stated to be practical for large hierarchical problems with relatively few discrete variables.

A critical difference between ATC and the MDO methods is that while the different MDO methods use nested optimization, within the ATC process the lower-level problems are completely solved before exchanging any responses and/or targets with their parent and/or children in the hierarchy [ALL 06].

The convergence properties of the ATC architecture have been proven for a given class of coordination strategies [MIC 03]. The ATC process convergence has been proven under given assumptions regarding the target matching or attaining. More complex coordination strategies have been developed in the literature for the case when the global targets of the top-level optimization problem are unattainable. Within the management community, intentionally-imposed unattainable targets are known as "stretch targets" or "stretch goals", having as goal to motivate employees [MIC 05].

Even though ATC does not belong to the MDO class of problems due to its object-based decomposition, rather than discipline-based as in the case of multidisciplinary problems, many authors consider it within the same class of methods as MDO. The ATC strategy employs elements common to other MDO methods, and the most resembling MDO method is the CO strategy. Apart from the decomposition of handled problems following objects instead of disciplines, the ATC strategy distinguishes also through the length of the hierarchy addressed, i.e. ATC is intended for addressing hierarchical decomposed problems over two or more levels, compared to the MDO methods which are only bi-level.

3.5.1 Basic mathematical formulation

Using the multi-level ATC strategy, the complex optimization problem is decomposed into several, smaller and more manageable interconnected problems, displaced on two or more hierarchical levels. The information, expressed as targets and responses, is communicated hierarchically from the top to the bottom of the hierarchy. Hence, no direct information exchange is allowed between elements situated on levels that are not directly connected. The information exchange between elements situated at the same level is possible, but not directly. This information is expressed in terms of shared variables between elements from the same level, and the exchange between two or more elements that share it will be done only through the intermediate of the parent level, which is in charge with ensuring the coordination.

In the literature, several different notations have been employed for addressing the different quantities exchanged between the elements of the ATC hierarchy. Nevertheless, a consensus has emerged on two standardized notations, used by the majority of research scientists in the field. The first notation convention belongs to the parents of the ATC strategy [KIM 03], [MIC 03]. The full set of notations is presented in [MIC 05b]. The second notational convention was later introduced by Tosserams et al. [TOS 06], [TOS 08]. The equivalence between the two notational conventions is presented in the appendix of [TOS 06]. Both notational conventions are used in parallel, the adoption of one or another being just a question of preference. The notations employed in this work are mainly adapted from the latter case, with a few slight differences.

3.5.2 *All-in-One (AIO) optimization problem formulation*

The general, non-decomposed optimization problem of the complex system to be designed, written under the most basic general form, is expressed as in (3.9). This form of the optimization problem is also known as the AIO problem [TOS 06], [LI 08a].

$$\begin{aligned} \underset{\mathbf{z}}{\text{Minimize}} \quad & f(\mathbf{z}) \\ \text{subject to} \quad & \mathbf{g}(\mathbf{z}) \leq \mathbf{0} \\ & \mathbf{h}(\mathbf{z}) = \mathbf{0} \end{aligned} \tag{3.9}$$

The design vector $\mathbf{z}$ regroups all the decision variables of the optimization problem. Decomposing the optimization problem following the hierarchy presented in Figure 3.5a implies distributing the design variables along the hierarchy, each set $\mathbf{x}_{ij} \in \mathbf{z}$ to its corresponding sub-problem P_{ij} of the hierarchy. Since a natural object-based decomposition of the problem is assumed, each sub-problem of the hierarchy representing a physical sub-system or component of the complex system to be designed, the design variables allocation is done in a natural manner. Each sub-problem has its own design variables $\mathbf{x}_{ij}$ and also its associated design feasibility constraints, $\mathbf{g}_{ij}$ respectively $\mathbf{h}_{ij}$, which are local to sub-problem P_{ij}.

An important assumption which is made is that the objective function f is additively separable; each sub-problem P_{ij} might have its own local objective function f_{ij}. In expression (3.9) the global objective function f of the problem is thus an aggregation of all these local objective functions f_{ij}.

The hierarchical problem before decomposition, which is known as the structured AIO problem, is expressed in (3.10). In this expression, the global objective function is separated into its corresponding set of objectives from each element optimization. The distinction is made between the local variables, specific to each element, and the exchanged quantities between elements.

$$\begin{aligned} \underset{\mathbf{z}}{\text{Minimize}} \quad & f = \sum_{i=1}^{N} \sum_{j \in \mathcal{E}_i} f_{ij}\left(\mathbf{x}_{ij}, \mathbf{t}_{ij}, \mathbf{t}_{(i+1)k_1}, \cdots, \mathbf{t}_{(i+1)k_{c_{ij}}}\right) \\ \text{subject to} \quad & \mathbf{g}_{ij}\left(\mathbf{x}_{ij}, \mathbf{t}_{ij}, \mathbf{t}_{(i+1)k_1}, \cdots, \mathbf{t}_{(i+1)k_{c_{ij}}}\right) \leq \mathbf{0} \\ & \mathbf{h}_{ij}\left(\mathbf{x}_{ij}, \mathbf{t}_{ij}, \mathbf{t}_{(i+1)k_1}, \cdots, \mathbf{t}_{(i+1)k_{c_{ij}}}\right) = \mathbf{0} \\ \text{where} \quad & k \in \mathsf{C}_{ij} \qquad j \in \mathcal{E}_i \qquad i = 1, \cdots, N \qquad \mathbf{x}_{ij} \in \mathbf{z} \end{aligned} \tag{3.10}$$

The three-level, seven-element hierarchical decomposition presented in Figure 3.5a is considered here and presented in Figure 3.10 after the introduction of the local variables for each sub-problem and the quantities exchanged between each parent-child couple.

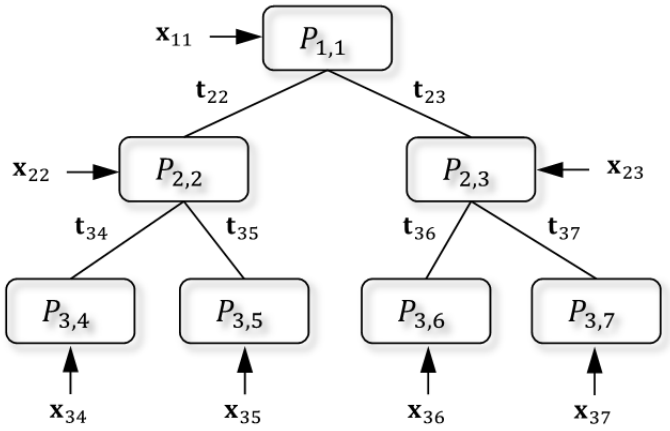


Figure 3.10 : Decomposed structure with local and exchanged variables

Since all objective and constraint functions depend on the quantities $\mathbf{t}_{ij}$ exchanged between different sub-problems, and which are common for two sub-problems, the AIO optimization problem cannot yet be decomposed.

In order to be able to decompose the AIO problem into several hierarchically-connected sub-problems, a differentiation must be made between the values of the same exchanged variable at the two sub-problems (i.e. parent-child couple) linked by it. Thus, the notion of response variable is introduced, in order to address the value of an exchanged variable which is calculated at a lower level and re-balanced up to the parent element. The vector of all shared and coupling variables of a sub-problem P_{ij}, which are re-balanced to its parent sub-problem, are regrouped under the variable $\mathbf{r}_{ij}$, denoting the sub-problem's responses to the previously received targets $\mathbf{t}_{ij}$. After introducing copies of the exchanged variables in all sub-problems, in order to differentiate between the quantities sent and received by each sub-problem, the AIO optimization problem takes the expression given in (3.11).

$$
\begin{aligned}
\underset{\bar{\mathbf{x}}_{11},\cdots,\bar{\mathbf{x}}_{NM}}{\text{Minimize}} \quad & f = \sum_{i=1}^{N} \sum_{j \in \mathcal{E}_i} f_{ij}\left(\bar{\mathbf{x}}_{ij}\right) \\
\text{subject to} \quad & \mathbf{g}_{ij}\left(\bar{\mathbf{x}}_{ij}\right) \leq \mathbf{0} \\
& \mathbf{h}_{ij}\left(\bar{\mathbf{x}}_{ij}\right) = \mathbf{0} \\
& \mathbf{c}_{ij} = \mathbf{t}_{ij} - \mathbf{r}_{ij} = \mathbf{0} \\
\text{where} \quad & \bar{\mathbf{x}}_{ij} = \left[\mathbf{x}_{ij}, \mathbf{r}_{ij}, \mathbf{t}_{(i+1)k_1}, \cdots, \mathbf{t}_{(i+1)k_{c_{ij}}}\right] \\
& k \in \mathsf{C}_{ij} \qquad j \in \mathcal{E}_i \qquad i = 1, \cdots, N
\end{aligned}
\tag{3.11}
$$

In order for the decomposed optimization problem to be consistent, the response variables $\mathbf{r}_{ij}$ of all sub-problems must match their corresponding target variables $\mathbf{t}_{ij}$. This condition is imposed in the expression of (3.11) by means of the additional constraint functions $\mathbf{c}_{ij} = \mathbf{t}_{ij} - \mathbf{r}_{ij} = \mathbf{0}$, which represent the consistency constraints of the decomposed optimization problem.

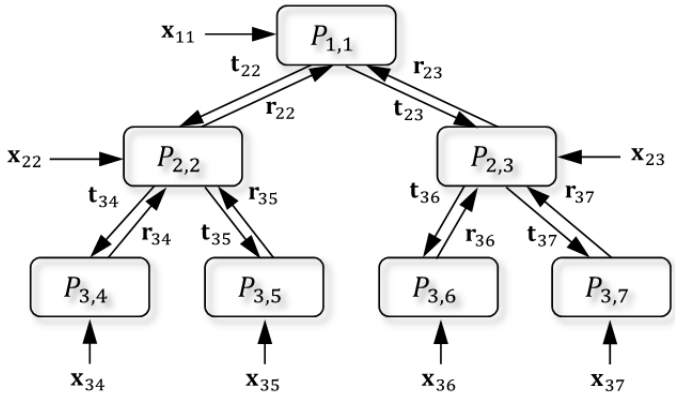
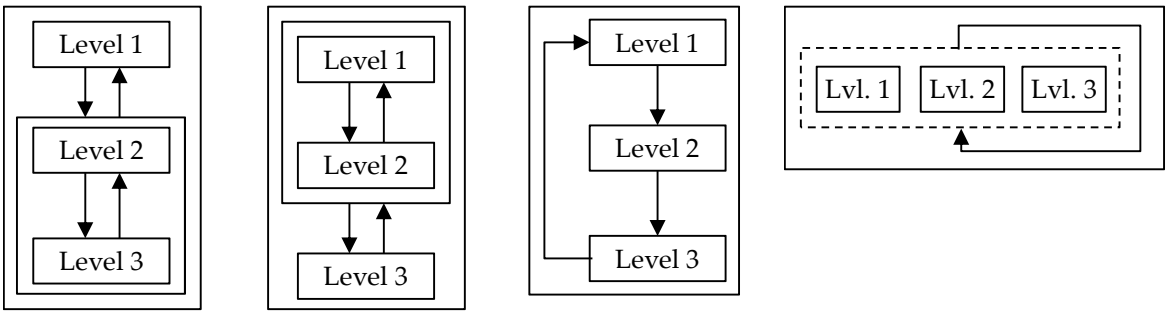


Figure 3.11 : Decomposed structure with local and exchanged variables with their copies

The key element of the optimization of decomposed problems using the ATC formulation consists in the management of the constraint functions ensuring the consistency of the ATC process. In comparison with the general objective function f, which is assumed to be additively separable into M objective functions f_{ij} and the inequality and equality constraint functions, $\mathbf{g}_{ij}(\bar{\mathbf{x}}_{ij})$ respectively $\mathbf{h}_{ij}(\bar{\mathbf{x}}_{ij})$, which are specific to each sub-problem of the hierarchy, the consistency constraint functions $\mathbf{c}_{ij}$ depend on all the exchanged variables (i.e. targets and responses, $\mathbf{t}_{ij}$ respectively $\mathbf{r}_{ij}$) between the different sub-problems of the hierarchy. Therefore, these consistency constraints are not separable following each sub-problem, and a special coordination strategy must be set in place in order to force the matching of these constraints and implicitly ensure the convergence of the ATC optimization process.

3.5.3 Coordination strategies for ATC decomposed systems

The coordination strategies associated with ATC have the role of organizing the information flux within the hierarchy, by defining the order in which the elements of a given level of the ATC hierarchy are evaluated. Different convergent ATC coordination strategies might be employed for the coordination of the ATC optimization process. Several nested coordination methods are presented in Figure 3.12. Some of them have convergence proofs (Figure 3.12a, Figure 3.12b), while others were proven convergent empirically, through different applications.



a) Nested-down b) Nested-up c) Simple loop d) Simple loop with parallel evaluation levels

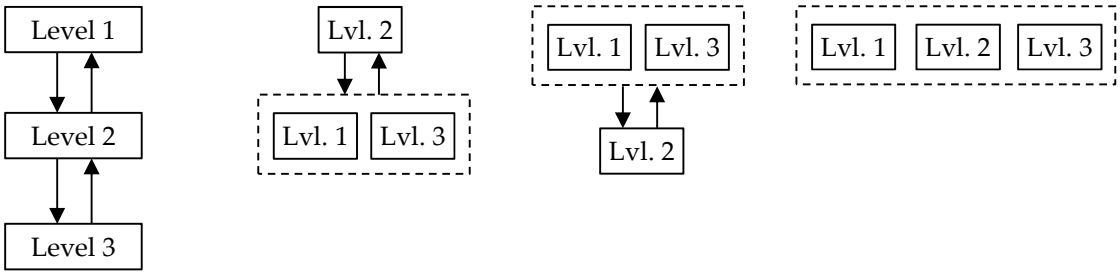
Figure 3.12 : Nested and simple loop coordination methods

The closed contours in Figure 3.12 suggest the loops existing between the different levels in the decomposed problem hierarchy. The first two coordination strategies, represented graphically in Figure 3.12a and Figure 3.12b consist of two nested loops, the inner loop being imbricated within the outer loop. Convergence proofs for these two coordination strategies and several other similar strategies have been offered by Michelena et al. in [MIC 03]. The first coordination strategy presented in Figure 3.12a is named nested-down and implies the complete resolution of the inner loop between the two lower levels of the hierarchy for each iteration of the outer loop. A full inner loop evaluation is thus completed for each outer loop iteration. The inner loop ensures the consistency of the constraints matching the responses from the bottom-level elements to the targets imposed by the middle-level elements. The outer loop run is in charge with ensuring the consistency between upper and middle-level elements and ends when an agreement is attained between the target and response variables of the upper respectively middle-level elements. The second coordination strategy, named nested-down and presented in Figure 3.12b, is similar to the first one, with the difference that the inner loop ensures this time the consistency between upper and middle-level elements, while the outer loop is in charge with ensuring the agreement between middle and lower-level elements of the hierarchy.

The third coordination strategy, presented in Figure 3.12c, makes use of a single loop for ensuring the consistency between the different levels of the hierarchy. This time, a unique loop is in charge of matching the responses to the targets between both upper-middle and middle-lower level elements. Here, the different levels of the hierarchy are evaluated sequentially, descending from the top to the bottom of the hierarchy. The coordination loop ensures the convergence of this sequence. This type of coordination strategy is called block coordinate descent (BCD) [LI 08].

The fourth coordination strategy, presented in Figure 3.12d, is somehow similar to the strategy presented in Figure 3.12c but has a great advantage that all levels are evaluated concurrently. The parallel evaluation of the levels of the hierarchy is suggested by the dashed-line contour in Figure 3.12d and it allows performing the task of optimizing all levels in parallel. Equally, a coordination loop ensures the consistency between all levels of the hierarchy. In order to obtain separable problems, Li et al. [LI 08] propose the use of a diagonal quadratic approximation (DQA), which linearizes the quadratic term of the augmented Lagrangian constraint relaxation presented later on in paragraph 3.5.6.3.

All previously-presented coordination strategies employ one or more loops for ensuring the consistency between the values of the targets and responses generated by each sub-problem. The presence of these loops implies a consequent computational effort, which penalizes the overall time dedicated to the complex system optimization task. To overcome the drawback represented by the computational complexity added by the coordination loops of the previous strategies, the coordination strategies presented in Figure 3.13 propose the elimination of these loops and therefore the reduction of the computational burden. However, as in the case of the previous strategies, these strategies still use a "meta-loop", in charge with coordinating the global ATC optimization process.



a) Each level evaluation at a time

b) Even-odd level evaluation

c) Odd-even level evaluation

d) Simultaneous evaluation of all levels

Figure 3.13 : No-loop coordination methods

The coordination strategy presented in Figure 3.13a uses a single-pass evaluation of each level of the hierarchy. The global "meta-loop" of the ATC optimization process is in charge with ensuring the consistency between targets and responses generated at each level. The targets are cascaded from the top of the hierarchy down to the bottom of the hierarchy and then re-balanced upwards to the top level, following the same path.

The strategies presented in Figure 3.13b and Figure 3.13c represent two versions of the previous coordination strategy of Figure 3.13a. These strategies exploit the specificity of the information flux between levels, mainly the fact that middle-level elements require information from both upper and lower levels, in order to compute the responses respectively targets for the upper, respectively bottom level. Hence, top and bottom-level elements can be evaluated concurrently. In the general case of a n-level hierarchy, this is translated into the even-odd level evaluation strategy in Figure 3.13b, respectively the odd-even level evaluation strategy in Figure 3.13c. For these two latter cases, the sequential level evaluation from the strategy in Figure 3.13a is transformed into a two-step evaluation of the odd, respectively even levels of the hierarchy. The concurrent evaluation of odd, respectively even levels is suggested by the dashed-line contours. An appropriate technique employing these coordination methods was proposed by Tosserams et al. in [TOS 06].

The last presented coordination strategy from Figure 3.13d represents the most generalized coordination strategy, where all sub-problems from all elements are evaluated in parallel. As in the previous three strategies, the "meta-loop" is in charge of attaining an agreement between the different targets and responses, but this time all levels at the same time. The computational burden of the optimization problem is reduced at maximum in the detriment of the additional implementation complexity and the difficulty in interpreting the different exchanged quantities. For this coordination method, a truncated diagonal quadratic approximation (TDQA) method, which limits the number of inner-loop iterations of DQA was proposed by Li et al. in [LI 08].

All coordination strategies presented in Figure 3.12 and Figure 3.13 have their advantages and their drawbacks and they require the association of a certain method for handling the consistency constraints, local to each sub-problem of the hierarchy. Thus, the assessment of their performance and their properties cannot be done independently, but with a given local consistency constraint handling method. A number of different test problems have been used to empirically estimate the performance of the different coordination methods by Li et al. in [LI 08]. Different consistency constraint handling methods are possible, and these will be reviewed next in the following paragraph.

3.5.4 Consistency constraints handling

The additional consistency constraints introduced previously in the expression of (3.11) require a special attention. These consistency constraints cannot be considered and handled as strict constraints by each optimization algorithm associated to each sub-problem optimization, since they depend on all target and response variables and therefore are not separable. The non-respect of these constraints must be accepted during the ATC optimization process, i.e. "meta-loop" execution, but at the end of the optimization process, all constraints, including consistency constraints, must be satisfied in order to obtain a feasible and consistent complex system design.

In the first works addressing the ATC optimization method, the consistency constraints were handled using additional tolerance variables for ensuring the compatibility of responses with targets between two sub-problems of adjacent levels [KIM 03], [KOK 02], [CHO 05]. The formulation of the optimization problem of a sub-problem P_{ij} after adding the additional tolerance variables is expressed in (3.12).

$$
\begin{aligned}
\underset{\bar{\mathbf{x}}_{ij}}{\text{Minimize}} \quad & w_{ij}^{R}\left\|\mathbf{R}_{ij}^{i}-\mathbf{R}_{ij}^{i-1}\right\|+w_{ij}^{y}\left\|\mathbf{y}_{ij}^{i}-\mathbf{y}_{ij}^{i-1}\right\|+\epsilon_{ij}^{R}+\epsilon_{ij}^{y} \\
\text{subject to} \quad & \sum_{k\in \mathsf{C}_{ij}} w_{(i+1)k}^{R}\left\|\mathbf{R}_{(i+1)k}^{i}-\mathbf{R}_{(i+1)k}^{i+1}\right\|\leq \epsilon_{ij}^{R} \\
& \sum_{k\in \mathsf{C}_{ij}} w_{(i+1)k}^{y}\left\|\mathbf{y}_{(i+1)k}^{i}-\mathbf{y}_{(i+1)k}^{i+1}\right\|\leq \epsilon_{ij}^{y} \\
& \mathbf{g}_{ij}\left(\mathbf{x}_{ij}^{i},\mathbf{y}_{ij}^{i},\mathbf{R}_{ij}^{i}\right)\leq \mathbf{0} \\
& \mathbf{h}_{ij}\left(\mathbf{x}_{ij}^{i},\mathbf{y}_{ij}^{i},\mathbf{R}_{ij}^{i}\right)=\mathbf{0} \\
\text{where} \quad & \mathbf{R}_{ij}^{i}=\mathbf{r}_{ij}\left(\tilde{\mathbf{x}}_{ij}\right) \\
& \tilde{\mathbf{x}}_{ij}=\left[\mathbf{x}_{ij}^{i},\mathbf{y}_{ij}^{i},\mathbf{R}_{(i+1)k_1}^{i},\cdots,\mathbf{R}_{(i+1)k_{c_{ij}}}^{i}\right] \\
& \bar{\mathbf{x}}_{ij}=\left[\mathbf{x}_{ij}^{i},\mathbf{y}_{ij}^{i},\mathbf{y}_{(i+1)k_1}^{i},\cdots,\mathbf{y}_{(i+1)k_{c_{ij}}}^{i},\mathbf{R}_{(i+1)k_1}^{i},\cdots,\mathbf{R}_{(i+1)k_{c_{ij}}}^{i},\epsilon_{ij}^{R},\epsilon_{ij}^{y}\right] \\
& j\in\mathcal{E}_i \qquad\qquad i=1,\cdots,N
\end{aligned}
\tag{3.12}
$$

where $\mathbf{R}_{ij}^{i}$ represents the vector of coupling variables of sub-problem P_{ij} calculated at the i-th level, $\mathbf{y}_{ij}^{i}$ is the vector of shared variables, w_{ij}^{R} and w_{ij}^{y} are the weighting coefficients vectors of the coupling variables, respectively shared variables, ϵ_{ij}^{R} and ϵ_{ij}^{y} are the tolerance values for the target matching of the coupling variables, respectively shared variables, $\mathbf{r}_{ij}$ is the analysis model of the P_{ij} sub-problem, $\tilde{\mathbf{x}}_{ij}$ is the design variables vector for the analysis model, $\bar{\mathbf{x}}_{ij}$ is the local and exchanged variables vector, C_{ij} is the set of children of the P_{ij} sub-problem.

In order to distinguish between the values of the different copies of the same variables, calculated at different elements of the hierarchy, the superscript is used in this formulation. The value of the superscript denotes the hierarchical level of the architecture at which the respective variable is calculated. Thus, the values of the coupling or shared variables that are issued by the

sub-problem P_{ij} are distinguished by the superscript "*i*", while the same variables computed by the sub-problem's parent and children are represented using the superscript "*i-1*", respectively "*i+1*". A distinction is made here between the coupling and the shared variables of the sub-problem, $\mathbf{R}_{ij}^{i}$ respectively $\mathbf{y}_{ij}^{i}$. The coupling variables between the sub-problem and its parent are represented by the responses of the sub-problem analysis in this formulation through the **R** symbol.

This consistency constraint handling method was associated with the first two nested coordination strategies previously presented in paragraph 3.5.3, Figure 3.12a and Figure 3.12b. The convergence of these associations has been proved for these coordination strategies and other similar nested strategies, and it is available in [MIC 03] and [TZE 03]. An ATC decomposition-based optimization platform using this association of constraint handling and nested coordination strategy was proposed in [ETM 05].

The representation of the different variables of a sub-problem P_{ij}, specific or exchanged with its parent and children elements, is offered in Figure 3.14.

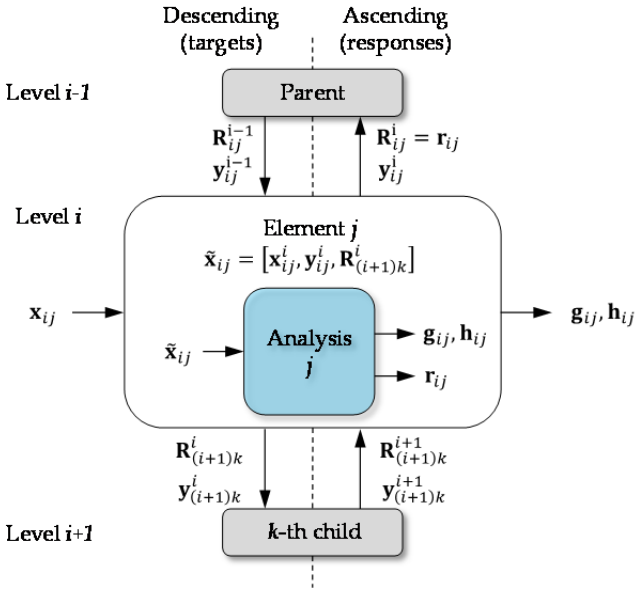


Figure 3.14 : Representation of the variables of a sub-problem P_{ij}

This ATC formulation does not accept the decomposition of the global objective function following the different sub-problems of the hierarchy, thus there are no local objective functions. Nevertheless, each sub-problem P_{ij} of the hierarchy has its own local constraint functions, $\mathbf{g}_{ij}$ respectively $\mathbf{h}_{ij}$ which it must satisfy. The global objective function is expressed under the form of design targets which are imposed to the root element of the hierarchy.

As with the multi-disciplinary CO formulation presented in paragraph 3.4, a certain level of inconsistency must be accepted at intermediate iterations of the optimization process in order for the ATC strategy to be able to converge. Hence, the consistency constraints are relaxed using a penalty function which is added to the objective function.

3.5.5 Relaxation of the consistency constraints of the AIO problem

The AIO optimization problem, after introducing the penalty terms in the objective function's expression, is formulated in (3.13). This formulation is also known as the relaxed AIO problem.

$$\begin{aligned}
&\underset{\bar{\mathbf{x}}_{11},\cdots,\bar{\mathbf{x}}_{NM}}{\text{Minimize}} && \sum_{i=1}^{N}\sum_{j\in\mathcal{E}_i} f_{ij}\left(\bar{\mathbf{x}}_{ij}\right) + \pi\left(\mathbf{c}(\bar{\mathbf{x}}_{11},\cdots,\bar{\mathbf{x}}_{NM})\right) \\
&\text{subject to} && \mathbf{g}_{ij}\left(\bar{\mathbf{x}}_{ij}\right) \leq \mathbf{0} \\
& && \mathbf{h}_{ij}\left(\bar{\mathbf{x}}_{ij}\right) = \mathbf{0} \\
&\text{where} && \bar{\mathbf{x}}_{ij} = \left[\mathbf{x}_{ij}, \mathbf{r}_{ij}, \mathbf{t}_{(i+1)k_1},\cdots,\mathbf{t}_{(i+1)k_{c_{ij}}}\right] \\
& && j \in \mathcal{E}_i \qquad i = 1,\cdots,N
\end{aligned} \tag{3.13}$$

where π represents the penalty function, having the role to reduce the discrepancies ($\mathbf{c}$) between the linking variables (coupling and shared variables), $\mathbf{r}_{ij}$ is the response variables vector of the sub-problem P_{ij} for its parent element and $\mathbf{t}_{(i+1)k_1}$ is the target variables vector of the sub-problem P_{ij} for its child element k_1 situated at level $(i+1)$.

After introducing the penalty term, the AIO problem can be finally decomposed following the different elements of the ATC hierarchy.

The general formulation of an optimization problem P_{ij} of the hierarchy is expressed in (3.14).

$$\begin{aligned}
&\underset{\bar{\mathbf{x}}_{ij}}{\text{Minimize}} && f_{ij}\left(\bar{\mathbf{x}}_{ij}\right) + \pi\left(\mathbf{c}(\bar{\mathbf{x}}_{11},\cdots,\bar{\mathbf{x}}_{NM})\right) \\
&\text{subject to} && \mathbf{g}_{ij}\left(\bar{\mathbf{x}}_{ij}\right) \leq \mathbf{0} \\
& && \mathbf{h}_{ij}\left(\bar{\mathbf{x}}_{ij}\right) = \mathbf{0} \\
&\text{where} && \bar{\mathbf{x}}_{ij} = \left[\mathbf{x}_{ij}, \mathbf{r}_{ij}, \mathbf{t}_{(i+1)k_1},\cdots,\mathbf{t}_{(i+1)k_{c_{ij}}}\right]
\end{aligned} \tag{3.14}$$

From (3.14), it can be seen that although every problem P_{ij} of the hierarchy has its own specific local objective and constraint functions, which depend on the local and linking variables, the penalty function is not separable, depending on all the variables of the AIO optimization problem.

A particular case appears when the response variables $\mathbf{r}_{ij}$ for the parent element are not decision variables for the P_{ij} problem, but instead, these responses are results of the analysis model of this sub-problem. In this case, the $\mathbf{r}_{ij}$ variable is not included into the decision vector, but it depends on the rest of the decision variables. In this case, the formulation of the P_{ij} optimization problem takes the form expressed in (3.15).

$$\begin{aligned}
&\underset{\bar{\mathbf{x}}_{ij}}{\text{Minimize}} && f_{ij}\left(\bar{\mathbf{x}}_{ij}\right) + \pi\left(\mathbf{c}(\bar{\mathbf{x}}_{11},\cdots,\bar{\mathbf{x}}_{NM})\right) \\
&\text{subject to} && \mathbf{g}_{ij}\left(\bar{\mathbf{x}}_{ij}\right) \leq \mathbf{0}
\end{aligned} \tag{3.15}$$

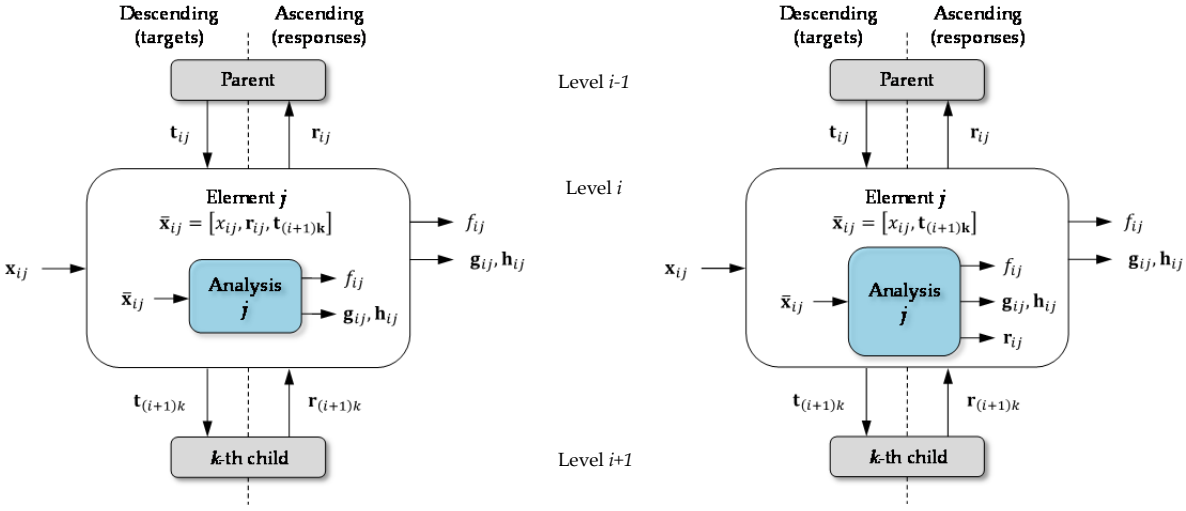
$$\mathbf{h}_{ij}(\bar{\mathbf{x}}_{ij}) = \mathbf{0}$$

where $\bar{\mathbf{x}}_{ij} = \left[\mathbf{x}_{ij}, \mathbf{t}_{(i+1)k_1}, \cdots, \mathbf{t}_{(i+1)k_{c_{ij}}}\right]$

$$\mathbf{r}_{ij} = \mathbf{a}_{ij}(\bar{\mathbf{x}}_{ij})$$

where $\mathbf{a}_{ij}$ represents the analysis model employed with the P_{ij} optimization problem.

According to the expressions of (3.14) and (3.15), the general optimization problem of an element P_{ij} of the hierarchical structure, the informational flow is presented in Figure 3.15.



a) Responses to parent element are inputs for the optimization problem of P_{ij}

b) Responses to parent element are calculated by the analysis model of P_{ij}

Figure 3.15 : General P_{ij} problem information flow

To simplify the annotation of elements in Figure 3.15, the notation $\mathbf{t}_{(i+1)\mathbf{k}} = \left[\mathbf{t}_{(i+1)k_1}, \cdots, \mathbf{t}_{(i+1)k_{c_{ij}}}\right]$ was employed to address all the target variables computed for all children of the P_{ij} problem. The first case represented graphically in Figure 3.15a represents the general formulation of a sub-problem P_{ij} of the ATC hierarchy. The input vector of the optimization problem and the analysis model of this element contain the local variables $\mathbf{x}_{ij}$ and all the variables exchanged with his parent and his children, $\mathbf{r}_{ij}$ respectively $\mathbf{t}_{(i+1)\mathbf{k}}$. In the second case, represented in Figure 3.15b, the answer for P_{ij} element's parent, the response $\mathbf{r}_{ij}$, is in fact calculated by the analysis model of this element this time. The use of one or the other formulation, depending on the need expressed through the decomposition of the initial problem, represents an important flexibility of the ATC formulation.

3.5.6 Penalty functions for the consistency constraint relaxation

Different general constraint relaxation methods exist in the literature, which have been adapted for the consistency ensuring of the ATC optimization process. A taxonomy of the different penalty functions employed with different coordination strategies is presented in [LI 08a]. The most basic of all constraint relaxation methods is represented by the quadratic penalty (Q) function. More

complex penalty methods, such as the Lagrangian relaxation (L) and the Augmented Lagrangian (AL) relaxation functions are intended to increase the robustness of the optimization.

3.5.6.1 Quadratic penalty relaxation

One of the most basic relaxation methods is represented by the quadratic penalty function. The convergence of the ATC optimization process employing the quadratic penalty function for relaxing the consistency constraints has been studied for different nested coordination schemes [TZE 03]. A multi-level optimization platform based on the specification language χ has been developed and presented in [ETM 05].

The expression of the quadratic penalty function is given in (3.16).

$$\pi(\mathbf{c}) = \pi_Q(\mathbf{c}) = \|\mathbf{w} \circ \mathbf{c}\|_2^2 = \sum_{i=1}^{N} \sum_{j \in \mathcal{E}_i} \left\|\mathbf{w}_{ij} \circ \mathbf{c}_{ij}\right\|_2^2 = \sum_{i=1}^{N} \sum_{j \in \mathcal{E}_i} \left\|\mathbf{w}_{ij} \circ \left(\mathbf{t}_{ij} - \mathbf{r}_{ij}\right)\right\|_2^2 \tag{3.16}$$

where the "$\circ$" sign denotes the term-by-term multiplication, also known as the Hadamard product, and the $\|\cdot\|_2^2$ symbol denotes the square of the l_2-norm, or the Euclidean norm, justifying the presence of the "quadratic" term in the denomination of the formulation.

The difficulty of the application of the quadratic penalty relaxation consists in the proper selection of the weighting coefficients $\mathbf{w}$. In order for the ATC process to converge and to attain the consistency of the design, large penalty weights must be used. However, using too large weights can cause ill-conditioning of the problem [BER 99], [BER 03], [LI 08a]. For problems which have attainable targets, it has been proven that using any finite weights could lead to convergence to a consistent design [MIC 05a]. Also, for problems with unattainable targets, there is no guarantee that the ATC process will converge to a consistent solution. The convergence properties of ATC employing this penalty formulation in combination with the nested coordination strategies presented in Figure 3.12a and Figure 3.12b have been analyzed in [MIC 03].

Classically, the weighting coefficients are determined using a trial-and-error method. Nevertheless, this is a very expensive approach, especially when the analysis models employed with each element optimization are computational intensive analysis codes, leading thus to a much increased computational cost than for the AIO problem. A weighting update method (WUM) has been proposed in [MIC 05a]. An important inconvenient of this method is that it requires the analytical expressions of the gradients of the objective and constraint functions. In practice, when using simulation codes, the analysis model is expressed as a black-box function, therefore no expression is available for these gradients, limiting thus the application of this method. To overcome this drawback, Moussouni et al. proposed in [MOU 09] a technique which transforms the quadratic weighting penalty function of the single-objective upper-most optimization problem of the hierarchy into a multi-objective problem, which is handled using a multi-objective genetic algorithm NSGA-II. An optimal solution which presents a minimal tolerance, decreased at each iteration, is extracted from the obtained Pareto front and used as target for the optimization problems at the inferior level of the ATC hierarchy. This technique brings however an additional computational complexity related to obtaining the Pareto front and it is difficultly generalizable.

3.5.6.2 Lagrangian relaxation

In order to enforce the consistency between different sub-problems and to improve the convergence of the ATC optimization process, the Lagrangian relaxation has been proposed for relaxing the consistency constraints of ATC decomposed problems. The Lagrangian relaxation method has been used for solving ATC decomposed problems in [LAS 05], [KIM 06]. This method is based on the Lagrangian duality theory, hence benefitting of convergence proofs under certain assumptions. The penalty function using the Lagrangian relaxation takes the form presented in (3.17).

$$\pi(\mathbf{c}) = \pi_L(\mathbf{c}) = \mathbf{v}^T\mathbf{c} = \sum_{i=2}^{N}\sum_{j\in\mathcal{E}_i}\mathbf{v}_{ij}^T\mathbf{c}_{ij} = \sum_{i=2}^{N}\sum_{j\in\mathcal{E}_i}\mathbf{v}_{ij}^T\left(\mathbf{t}_{ij}-\mathbf{r}_{ij}\right) \tag{3.17}$$

where $\mathbf{v}$ is the set of Lagrangian multipliers $\mathbf{v}_{ij}$ of all elements from the ATC hierarchy.

The AIO problem in (3.13), after introducing the Lagrangian relaxation penalty function, takes the form (3.18). This is known as the Lagrangian relaxed problem or the Lagrangian problem (LP).

$$\begin{aligned}
&\underset{\bar{\mathbf{x}}_{11},\cdots,\bar{\mathbf{x}}_{NM}}{\text{Minimize}} \quad \Psi(\mathbf{v}) = \sum_{i=1}^{N}\sum_{j\in\mathcal{E}_i} f_{ij}\left(\bar{\mathbf{x}}_{ij}\right) + \sum_{i=2}^{N}\sum_{j\in\mathcal{E}_i}\mathbf{v}_{ij}^T\left(\mathbf{t}_{ij}-\mathbf{r}_{ij}\right)\\
&\text{subject to} \quad \mathbf{g}_{ij}\left(\bar{\mathbf{x}}_{ij}\right) \le \mathbf{0}\\
&\qquad\qquad \mathbf{h}_{ij}\left(\bar{\mathbf{x}}_{ij}\right) = \mathbf{0}\\
&\text{where} \quad \bar{\mathbf{x}}_{ij} = \left[\mathbf{x}_{ij}, \mathbf{r}_{ij}, \mathbf{t}_{(i+1)k_1}, \cdots, \mathbf{t}_{(i+1)k_{c_{ij}}}\right]\\
&\qquad\qquad j \in \mathcal{E}_i \qquad i = 1, \cdots, N
\end{aligned} \tag{3.18}$$

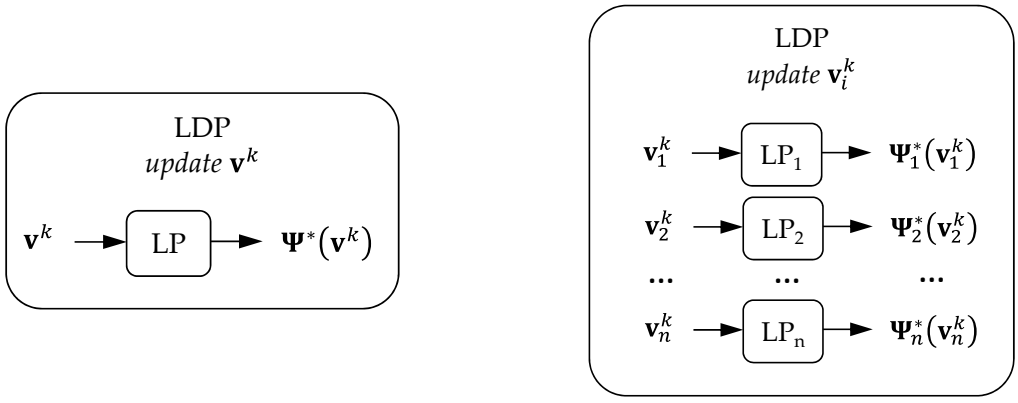
where Ψ represents the objective function of the Lagrangian problem.

According to the Lagrangian duality theory [BER 99], solving the original AIO problem is equivalent to solving the Lagrangian dual problem (LDP) given in (3.19).

$$\underset{\mathbf{v}}{\text{Maximize}} \quad \Psi^*(\mathbf{v}) \tag{3.19}$$

where $\Psi^*(\mathbf{v})$ represents the optimal value of the objective function obtained from problem (3.18), for a given vector of Lagrange multipliers $\mathbf{v}$.

Solving the original optimization problem using the ATC formulation and the Lagrangian relaxation consists thus of a nested formulation of the Lagrangian dual problem (LDP) – Lagrangian relaxed AIO problem (LP). Schematically this formulation is represented in Figure 3.16.



a) Nested LDP – AIO LP problem b) Nested LDP – decomposed LP problem

Figure 3.16 : ATC formulation with the nested Lagrangian dual problems

Figure 3.16a presents the nested LDP – AIO relaxed problem. In Figure 3.16b, the LP_i loop represents the set of optimization problems of the *i*-th level of the ATC hierarchy. The ATC coordination meta-loop is represented thus by the LDP problem, managing the resolution of the optimization problems of all levels of the hierarchy and updating the values of the Lagrange multiplier estimates. Since all Lagrangian multiplier estimates are calculated altogether at a given iteration of the ATC process, the optimization problems from every level of the hierarchy can be all executed in parallel. This ATC formulation is hence suited for application with the coordination technique presented in Figure 3.13d.

The iterative process of ATC starts with an initial guess for the Lagrange multipliers $\mathbf{v}^0$. At each iteration of the coordination meta-loop of the ATC optimization process, the values of the estimates of the Lagrange multipliers are updated based on the values from the previous iteration, using the expression (3.20).

$$\mathbf{v}^{k+1} = \mathbf{v}^k + \alpha^k \mathbf{c}^k = \mathbf{v}^k + \alpha^k (\mathbf{t}^k - \mathbf{r}^k) \tag{3.20}$$

where α is a scalar variable representing the step size.

The convergence of the ATC process is ensured if the conditions given in (3.21) are satisfied.

$$\begin{cases} \alpha^k \to 0 & \text{as} \quad k \to \infty \\ \sum_{k=1}^{\infty} \alpha^k = \infty \end{cases} \tag{3.21}$$

A special attention is required for the selection of an appropriate value for the step size α. In [LAS 05] a value of $\alpha = 1/k$ was suggested.

The convergence of the ATC process is attained when the values of the inconsistency between each interconnected elements of the hierarchy at a certain iteration of ATC become inferior to a pre-imposed tolerance value ε, condition which is expressed in (3.22).

$$\|\mathbf{c}^k\| = \|\mathbf{t}^k - \mathbf{r}^k\| \leq \varepsilon \tag{3.22}$$

At the convergence of the ATC process, the optimal values of $\mathbf{v}^*$ represent the values of the Lagrangian multipliers.

An important advantage of the ATC formulation with the Lagrangian relaxation consists of the completely parallelizable character of the coordination strategy. The optimization problems for the elements from all levels of the ATC hierarchy can all be launched in parallel, which generates an important reduction of the overall ATC optimization time. The inconvenient of this method consists in the possibility of existence of duality gaps in the expression of the non-relaxed AIO problem. Also, convergence difficulties have been reported for mathematical test problems [LI 08a].

3.5.6.3 Augmented Lagrangian relaxation

A lot of research effort has been addressed lately to the use of AL coordination for the ATC decomposed problems, [BLO 05], [KIM 06], [ALL 06], [ALL 10], [TOS 08], [TOS 09]. This represents an improved version of the Lagrangian relaxation previously presented in paragraph 3.5.6.2, which is meant to improve the convergence of the ATC strategy employing it. As with the previous relaxation method, Lagrangian duality theory rules also apply to for the AL relaxation method. The expression of the regular Lagrangian penalty function is augmented by a quadratic term. The expression of the AL penalty function takes the form presented in (3.23).

$$\begin{aligned}\pi(\mathbf{c}) = \pi_{AL}(\mathbf{c}) = \mathbf{v}^T\mathbf{c} + \|\mathbf{w}\circ\mathbf{c}\|_2^2 &= \sum_{i=1}^{N}\sum_{j\in\mathcal{E}_i}\left(\mathbf{v}_{ij}^T\mathbf{c}_{ij} + \left\|\mathbf{w}_{ij}\circ\mathbf{c}_{ij}\right\|_2^2\right)\\ &= \sum_{i=1}^{N}\sum_{j\in\mathcal{E}_i}\left(\mathbf{v}_{ij}^T(\mathbf{t}_{ij}-\mathbf{r}_{ij}) + \left\|\mathbf{w}_{ij}\circ(\mathbf{t}_{ij}-\mathbf{r}_{ij})\right\|_2^2\right)\end{aligned} \tag{3.23}$$

The additional quadratic term acts as a convexifier for the original AIO problem, enhancing thus the robustness of the optimization process [BLO 05].

The AIO relaxed problem using the AL penalty function takes thus the form of (3.24).

$$\begin{aligned}&\underset{\bar{\mathbf{x}}_{11},\cdots,\bar{\mathbf{x}}_{NM}}{\text{Minimize}} \quad \Psi'([\mathbf{v},\mathbf{w}]) = \sum_{i=1}^{N}\sum_{j\in\mathcal{E}_i} f_{ij}(\bar{\mathbf{x}}_{ij}) + \sum_{i=2}^{N}\sum_{j\in\mathcal{E}_i}\left(\mathbf{v}_{ij}^T(\mathbf{t}_{ij}-\mathbf{r}_{ij}) + \left\|\mathbf{w}_{ij}\circ(\mathbf{t}_{ij}-\mathbf{r}_{ij})\right\|_2^2\right)\\ &\text{subject to} \quad \mathbf{g}_{ij}(\bar{\mathbf{x}}_{ij}) \le \mathbf{0}\\ &\qquad\qquad \mathbf{h}_{ij}(\bar{\mathbf{x}}_{ij}) = \mathbf{0}\\ &\text{where} \quad \bar{\mathbf{x}}_{ij} = \left[\mathbf{x}_{ij}, \mathbf{r}_{ij}, \mathbf{t}_{(i+1)k_1}, \cdots, \mathbf{t}_{(i+1)k_{c_{ij}}}\right]\\ &\qquad\qquad j\in\mathcal{E}_i \qquad\qquad i = 1,\cdots,N\end{aligned} \tag{3.24}$$

where Ψ' represents the objective function of the AIO relaxed problem with AL penalty function.

As for the simple Lagrangian relaxation of paragraph 3.5.6.2, the dual problem is given in (3.25).

$$\underset{[\mathbf{v},\mathbf{w}]}{\text{Maximize}} \quad \Psi'^*([\mathbf{v},\mathbf{w}]) \tag{3.25}$$

where $\Psi'^*([\mathbf{v},\mathbf{w}])$ represents the optimal value of the objective function obtained from problem (3.24), for a given vector of the augmented Lagrange multipliers $\mathbf{v}$ and $\mathbf{w}$.

Compared to the Lagrangian relaxation of paragraph 3.5.6.2, where the term containing the Largrange multiplier estimates is separable, allowing thus the decomposition of the relaxed AIO problem into several optimization problems, for the AL relaxation this is no longer possible. The cause of the inseparability is given by the quadratic term of AL. Therefore, this does not allow a parallel evaluation of all optimization problems from all levels of the ATC hierarchy and it cannot be employed with such a coordination strategy. A nested coordination strategy, as the ones presented in Figure 3.12a and Figure 3.12b need to be employed with the AL relaxation.

The values of the linear term coefficients of the AL relaxation at a given iteration of the ATC process are updated using the expression given in (3.26).

$$\mathbf{v}^{k+1} = \mathbf{v}^k + 2\mathbf{w}^k \circ \mathbf{w}^k \circ \mathbf{c}^k \tag{3.26}$$

At an ATC iteration, the values of the new linear coefficients of the AL relaxation depend on the values of the linear and quadratic term coefficients, as well as the inconsistency from the previous iteration. At convergence of ATC, the values of $\mathbf{v}$ are in fact the values of the Lagrange multipliers.

The combined use of the AL relaxation with the update scheme for the coefficient of the linear AL relaxation term is known in the nonlinear programming theory as the "method of multipliers" [BER 99]. Hence, convergence properties exist for the method of multipliers under general assumptions and provided that $\mathbf{v}^k$ respects relation (3.27).

$$\mathbf{0} < \mathbf{v}^k < \mathbf{v}^{k+1} \tag{3.27}$$

For the quadratic term coefficients of the AL relaxation, a linear scheme is usually used for updating the value of $\mathbf{w}$ [TOS 06], [ALL 06], as given in (3.28).

$$\mathbf{w}^{k+1} = \beta \mathbf{w}^k \qquad \beta \geq 1 \tag{3.28}$$

where β is a constant parameter.

The β parameter has a direct influence on the convergence of the optimization process. In [TOS 06], a value of $\beta \in [2,3]$ has been suggested for accelerating the convergence. Also, the initialization of $\mathbf{v}^0 = \mathbf{0}$ and $\mathbf{w}^0 = \mathbf{1}$ was suggested.

The ATC process with the AL relaxation represents a robust implementation of the ATC optimization process, and also benefits of mathematical convergence proofs. It has been successfully applied for solving both abstract mathematical problems and industrial design problem, such as aircraft design problems [ALL 06a]. Nonetheless, the use of a nested coordination technique impacts on the total computational effort of the ATC optimization process.

3.5.6.4 Augmented Lagrangian relaxation with the alternating direction method of multipliers

To overcome the nested coordination feature of the AL relaxation method of the previous paragraph, the combination of AL relaxation and the alternating direction method of multipliers (AL-AD) was proposed for the ATC strategy by Tosserams et al. in [TOS 06]. The AL-AD method consists of alternating the evaluation of the optimization problems for the elements of each level of the ATC hierarchy. The AL-AD method is thus designed to work with the even-odd and odd-even level alternating coordination schemes presented in Figure 3.13b, respectively Figure 3.13c.

The optimization problem of each element of the ATC hierarchy takes the expression of (3.29).

$$\begin{aligned}
\underset{\bar{\mathbf{x}}_{ij}}{\text{Minimize}} \quad & f_{ij}(\bar{\mathbf{x}}_{ij}) + \mathbf{v}_{ij}^T(-\mathbf{r}_{ij}) + \left\|\mathbf{w}_{ij} \circ (\mathbf{t}_{ij} - \mathbf{r}_{ij})\right\|_2^2 \\
& + \sum_{k \in \mathsf{C}_{ij}} \left(\mathbf{v}_{(i+1)k}^T \mathbf{t}_{(i+1)k} + \left\|\mathbf{w}_{(i+1)k} \circ (\mathbf{t}_{(i+1)k} - \mathbf{r}_{(i+1)k})\right\|_2^2\right) \\
\text{subject to} \quad & \mathbf{g}_{ij}(\bar{\mathbf{x}}_{ij}) \leq \mathbf{0} \\
& \mathbf{h}_{ij}(\bar{\mathbf{x}}_{ij}) = \mathbf{0} \\
\text{where} \quad & \bar{\mathbf{x}}_{ij} = \left[\mathbf{x}_{ij}, \mathbf{r}_{ij}, \mathbf{t}_{(i+1)k_1}, \cdots, \mathbf{t}_{(i+1)k_{c_{ij}}}\right] \\
& k \in \mathsf{C}_{ij} \qquad j \in \mathcal{E}_i \qquad i = 1, \cdots, N
\end{aligned} \tag{3.29}$$

Separating the relaxed AIO problem, the expression of a P_{ij} problem of the hierarchy makes a distinction between the elements exchanged with its parent element from the upper level and the children elements from the lower level of the ATC hierarchy. Hence, an optimization problem P_{ij} of the hierarchy is in charge with finding the appropriate response values for the targets received from its parent, and at the same time, suited targets for its children, in accordance with the received responses from its children. A mathematical example is next employed to exemplify the purposes.

3.5.7 Mathematical example

The original non-decomposed optimization problem is formulated using the expression (3.30).

$$\begin{aligned}
\underset{\mathbf{x}}{\text{Minimize}} \quad & f(\mathbf{x}) = x_1^2 + x_2^2 \\
\text{with} \quad & \mathbf{x} = [x_1, x_2, \cdots, x_{14}] \\
\text{subject to} \quad & g_1(\mathbf{x}) = x_3^{-2} + x_4^2 - x_5^2 \leq 0 \qquad g_2(\mathbf{x}) = x_5^2 + x_6^{-2} - x_7^2 \leq 0 \\
& g_3(\mathbf{x}) = x_8^2 + x_9^2 - x_{11}^2 \leq 0 \qquad g_4(\mathbf{x}) = x_8^{-2} + x_{10}^2 - x_{11}^2 \leq 0 \\
& g_5(\mathbf{x}) = x_{11}^2 + x_{12}^{-2} - x_{13}^2 \leq 0 \qquad g_6(\mathbf{x}) = x_{11}^2 + x_{12}^2 - x_{14}^2 \leq 0 \\
& h_1(\mathbf{x}) = x_3^2 + x_4^{-2} + x_5^2 - x_1^2 = 0 \qquad h_2(\mathbf{x}) = x_5^2 + x_6^2 + x_7^2 - x_2^2 = 0 \\
& h_3(\mathbf{x}) = x_8^2 + x_9^{-2} + x_{10}^{-2} + x_{11}^2 - x_3^2 = 0 \qquad h_4(\mathbf{x}) = x_{11}^2 + x_{12}^2 + x_{13}^2 + x_{14}^2 - x_6^2 = 0 \\
\text{where} \quad & 0 \leq x_1, x_2, \cdots, x_{14} \leq 5
\end{aligned} \tag{3.30}$$

Using a trust-region algorithm, SQP, with a multi-start strategy to ensure the obtaining of the global optimum of the test problem, the solution of this problem is found and the optimal values for the variables are presented in Table 3.1.

Table 3.1 : Optimal configuration of the analytical test problem

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	f
2.84	3.09	2.36	0.76	0.87	2.81	0.94	0.97	0.87	0.8	1.3	0.84	1.76	1.55	17.59

The analytical test problem has one global optimum, presented in Table 3.1, which is the same as the one indicated in [KIM 03].

Multi-level ATC formulation with unattainable targets

The multi-level decomposition of the test problem can be done in several different ways. Figure 3.17 presents a bi-level decomposition of the initial problem into three elements [KIM 03].

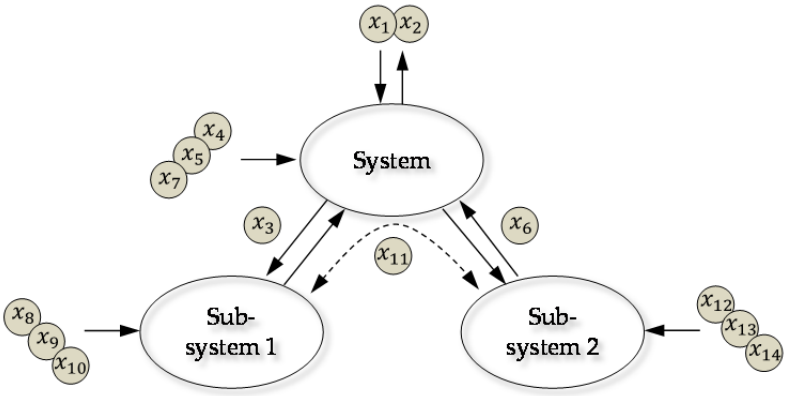


Figure 3.17 : Multi-level representation of the analytical test problem

Using the un-relaxed multi-level formulation, the system element from the top level of the analytical test problem can be expressed as in (3.31).

$$
\begin{aligned}
P_{sys}: \quad & \underset{\bar{\mathbf{x}}_{sys}}{\text{Minimize}} && f_{sys}(\bar{\mathbf{x}}_{sys}) = \left\|\mathbf{T}_{sys} - \mathbf{R}_{sys}\right\|_2^2 \\
& \text{with respect to} && \bar{\mathbf{x}}_{sys} = \left[\mathbf{x}_{sys}, \mathbf{t}_{ss}\right] \\
& \text{where} && \mathbf{x}_{sys} = [x_4, x_5, x_7] \qquad \mathbf{t}_{ss} = [x_{11}, x_3, x_6] \\
& && \mathbf{T}_{sys} = [0,0] \qquad \mathbf{R}_{sys} = [x_1, x_2] \\
& && h_1 \mapsto x_1 = \sqrt{x_3^2 + x_4^{-2} + x_5^2} \qquad h_2 \mapsto x_2 = \sqrt{x_5^2 + x_6^2 + x_7^2} \\
& \text{subject to} && \mathbf{c}_{sys} = \bar{\mathbf{t}}_{ss} - \bar{\mathbf{r}}_{ss} = [\bar{\mathbf{t}}_{ss1}, \bar{\mathbf{t}}_{ss2}] - [\bar{\mathbf{r}}_{ss1}, \bar{\mathbf{r}}_{ss2}] = \mathbf{0} \\
& && \bar{\mathbf{t}}_{ss1} = [x_{11}, x_3] \qquad \bar{\mathbf{t}}_{ss2} = [x_{11}, x_6] \\
& && \bar{\mathbf{r}}_{ss1} = [x_{11}^L, x_3^L] \qquad \bar{\mathbf{r}}_{ss2} = [x_{11}^L, x_3^L] \\
& && g_{sys,1}(\bar{\mathbf{x}}_{sys}) = g_1 = x_3^{-2} + x_4^2 - x_5^2 \leq 0 \\
& && g_{sys,2}(\bar{\mathbf{x}}_{sys}) = g_2 = x_5^2 + x_6^{-2} - x_7^2 \leq 0 \\
& && 0 \leq x_3, x_4, x_5, x_6, x_7, x_{11} \leq 5
\end{aligned}
\tag{3.31}
$$

To minimize the objective function $f = f_{sys}$, the global target $\mathbf{T}_{sys}$ is imposed null. The optimization problem is thus formulated using unattainable targets.

The two variables from the expression of the objective function, x_1 and x_2 and the associated equality constraint functions h_1 and h_2 were suppressed from the optimization problem formulation. Instead, these variables are considered outputs of the system level analysis model.

At the first iteration of the ATC optimization process, the consistency constraints are not accounted for, since the lower level elements were not yet launched to obtain response values for the shared and coupling variables, $\bar{\mathbf{r}}_{ss} = [\bar{\mathbf{r}}_{ss1}, \bar{\mathbf{r}}_{ss2}] = \left[[x_{11}^L, x_3^L], [x_{11}^L, x_6^L]\right]$. The L superscript denotes the response received from the **L**ower level. Once the values for the shared and coupling variables are determined at the system level, these are next "cascaded" as targets to the two children of the system element, P_{ss1} and P_{ss2}. The un-relaxed expressions of the optimization problems for the sub-system level elements P_{ss1} and P_{ss2} are given in (3.32) and (3.33), respectively.

$$
\begin{aligned}
P_{ss1}: \quad & \underset{\bar{\mathbf{x}}_{ss1}}{\text{Minimize}} && f_{ss1}(\bar{\mathbf{x}}_{ss1}) = 0 \\
& \text{with respect to} && \bar{\mathbf{x}}_{ss1} = [\mathbf{x}_{ss1}, \bar{\mathbf{r}}_{ss1}] \\
& \text{where} && \mathbf{x}_{ss1} = [x_8, x_9, x_{10}] \qquad \bar{\mathbf{r}}_{ss1} = [\mathbf{r}_{ss1}, x_3] \qquad \mathbf{r}_{ss1} = x_{11} \\
& && h_3 \mapsto x_3 = \sqrt{x_8^2 + x_9^{-2} + x_{10}^{-2} + x_{11}^2} \\
& \text{subject to} && \mathbf{c}_{ss1} = \bar{\mathbf{t}}_{ss1} - \bar{\mathbf{r}}_{ss1} = [x_{11}^U, x_3^U] - [x_{11}, x_3] = \mathbf{0} \\
& && g_{ss1,1}(\bar{\mathbf{x}}_{ss1}) = g_3 = x_8^2 + x_9^2 - x_{11}^2 \leq 0 \\
& && g_{ss1,2}(\bar{\mathbf{x}}_{ss1}) = g_4 = x_8^{-2} + x_{10}^2 - x_{11}^2 \leq 0 \\
& && 0 \leq x_8, x_9, x_{10}, x_{11} \leq 5
\end{aligned}
\tag{3.32}
$$

The superscript U of the shared and coupling variables x_{11}^U respectively x_3^U denotes the targets for these variables, received from the element's parent at the **U**pper level.

$$
\begin{aligned}
P_{ss2}: \quad & \underset{\bar{\mathbf{x}}_{ss2}}{\text{Minimize}} && f_{ss2}(\bar{\mathbf{x}}_{ss2}) = 0 \\
& \text{with respect to} && \bar{\mathbf{x}}_{ss2} = [\mathbf{x}_{ss2}, \bar{\mathbf{r}}_{ss2}] \\
& \text{where} && \mathbf{x}_{ss2} = [x_{12}, x_{13}, x_{14}] \qquad \bar{\mathbf{r}}_{ss2} = [\mathbf{r}_{ss2}, x_6] \qquad \mathbf{r}_{ss2} = x_{11} \\
& && h_4 \mapsto x_6 = \sqrt{x_{11}^2 + x_{12}^2 + x_{13}^2 + x_{14}^2} \\
& \text{subject to} && \mathbf{c}_{ss2} = \bar{\mathbf{t}}_{ss2} - \bar{\mathbf{r}}_{ss2} = [x_{11}^U, x_6^U] - [x_{11}, x_6] = \mathbf{0} \\
& && g_{ss2,1}(\bar{\mathbf{x}}_{ss2}) = g_5 = x_3^{-2} + x_4^2 - x_5^2 \leq 0 \\
& && g_{ss2,1}(\bar{\mathbf{x}}_{ss2}) = g_6 = x_5^2 + x_6^{-2} - x_7^2 \leq 0 \\
& && 0 \leq x_{11}, x_{12}, x_{13}, x_{14} \leq 5
\end{aligned}
\tag{3.33}
$$

Since this partitioning of the problem implies only two levels, there is no intermediary level, but only top and bottom levels.

ATC optimal results

For the multi-level optimization of a problem, two figure-cases may appear, depending on the values of the pre-imposed targets of the upper-most (root) element of the hierarchy. The ATC formulation might thus have attainable or unattainable targets. If there exists a feasible, consistent design which matches the imposed targets, these are called attainable. Otherwise, targets such as $\mathbf{t} = \mathbf{0}$ are considered unattainable, and the resolution of the multi-level problem using the ATC formulation requires special attention.

Among the coordination strategies and consistency constraint relaxation techniques presented previously in paragraphs 3.5.3 respectively 3.5.6, two implementations were tested for the bi-level decomposed structure of the mathematical test-problem. First, the quadratic penalty relaxation technique introduced in paragraph 3.5.6.1 was employed for solving the hierarchically decomposed problem for the ATC problem with attainable targets. To override the difficulties related to the selection of proper values for the weighting coefficients and the requirement of providing attainable targets, the augmented Lagrangian penalty function was then implemented along with the alternating direction method of multipliers of paragraph 3.5.6.4 for solving the optimization problem with the general formulation employing unattainable targets.

ATC optimization with attainable targets

When attainable targets are known for the optimization problem, meaning that a feasible design exists such as the responses of the upper-most element of the ATC hierarchy match the pre-imposed attainable targets, the solution to the optimization problem can be easily identified by the ATC optimization process using the quadratic penalty relaxation.

For the analytical test-problem previously introduced, such attainable targets can be considered, for example, the system-level response values of the optimal design obtained by the SQP method, $\mathbf{t} = \left[x_1^{SQP}, x_2^{SQP}\right] = [2.84,3.10]$. The ATC optimization process with attainable targets converged in only 17 iterations. The optimal results are presented in Table 3.2 along with the SQP results.

Table 3.2 : Multi-level analytical test-problem optimal results obtained by ATC with QP, for the attainable targets $\mathbf{t} = \left[x_1^{SQP}, x_2^{SQP}\right] = [2.84,3.10]$

Algo.	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	f
SQP	2.835	3.090	2.356	0.760	0.870	2.812	0.940	0.972	0.865	0.796	1.301	0.841	1.763	1.549	17.589
ATC	2.848	3.120	2.272	0.656	0.790	2.892	0.862	0.955	0.967	0.865	1.359	0.841	1.806	1.598	17.845

From the results in Table 3.2 it can be remarked that the obtained ATC optimal design presents values for the target variables x_1 and x_2 very close to the pre-imposed attainable targets, and consequently, to those of the optimum obtained by the SQP method.

Nevertheless, in industrial applications, it is rarely the case when attainable targets are known. A technique for the general case of unattainable targets imposes. The ATC formulation with the AL-AD method offers the possibility of finding the optimal design with such unattainable targets.

ATC optimization with unattainable targets

The ATC implementation using the AL-AD method [TOS 06] presented in paragraph 3.5.6.4 was run for solving the multi-level analytical test-problem with unattainable targets. This ATC implementation requires no setting or tuning of any weighting coefficients or any other coefficients, which represents a great advantage for the designer. An initial feasible design was supplied to the ATC process and the optimization was run to find a design with a high level of consistency (10^{-5}) between the values of the targets and responses exchanged between the elements of the hierarchy.

The convergence was attained after 123 iterations of the ATC process. The optimal results are presented in Table 3.3 for comparison with the already known optimal design, obtained using the SQP algorithm implementation of Matlab®.

Table 3.3 : Multi-level analytical test-problem optimal results obtained by ATC with AL-AD

Algo.	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	f
SQP	2.835	3.090	2.356	0.760	0.870	2.812	0.940	0.972	0.865	0.796	1.301	0.841	1.763	1.549	17.589
ATC	2.595	3.117	2.322	0.759	0.873	2.839	0.942	0.966	0.901	0.821	1.321	0.841	1.778	1.566	17.603

One interesting thing that can be remarked from the results comparison in Table 3.3 is that although the two optimal values of the objective function have practically the same value, these values were obtained with different values of the x_1 and x_2 underlying variables. While the values of the local variables of each sub-system of the ATC structure are almost identical for the two optima, the difference between the two designs is given by the values of the linking variables.

The evolution history of the different design variables of the analytical problem during the ATC optimal design process is presented graphically in Figure 3.18.

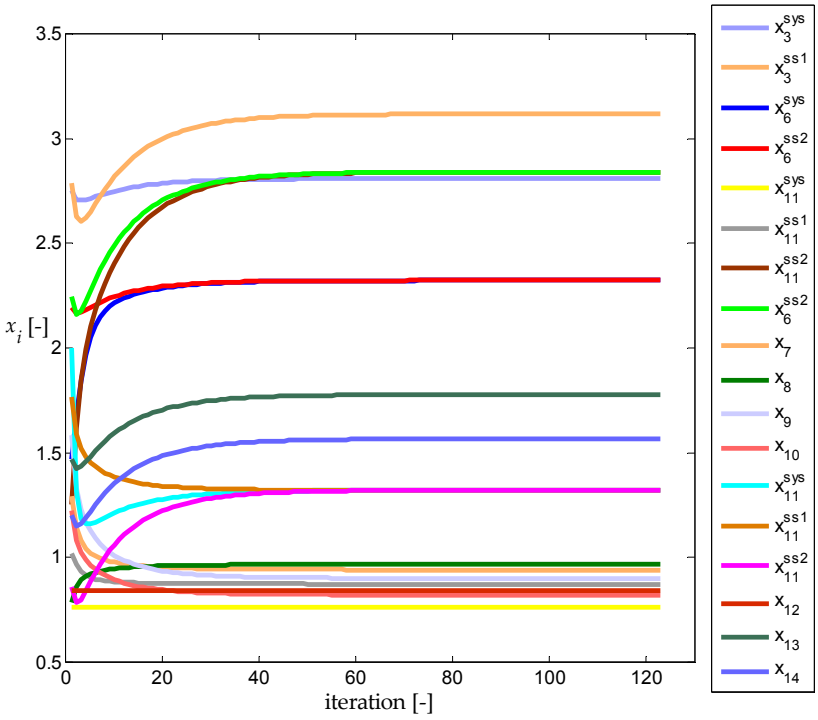
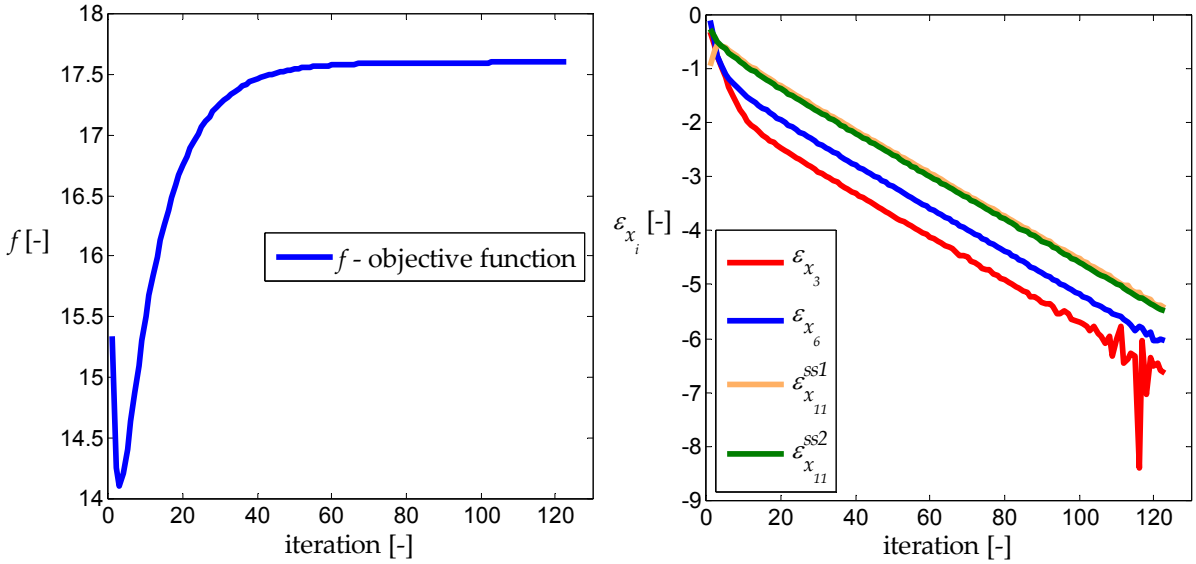


Figure 3.18 : Design variables evolution history for ATC employing the AL-AD method

The evolution of the objective function value during the ATC optimization process is presented graphically in Figure 3.19a. An important element to be followed during the ATC optimization process is the consistency of the design proposed by the optimization process. The consistency of the design refers to the discrepancy between the values of the same linking variable at different levels of the ATC multi-level structure. To quantify this discrepancy, an error measure is introduced, having the expression given in (3.34).

$$\varepsilon_{x_i} = \log_{10}\left|\frac{x_i^U - x_i^L}{x_i^U}\right| \tag{3.34}$$

The evolution of the inconsistency between the targets and responses exchanged between the elements of the ATC structure is presented graphically in Figure 3.19b for the two coupling variables x_3 and x_6 and the variable shared between the two lower-level sub-systems, x_{11}.

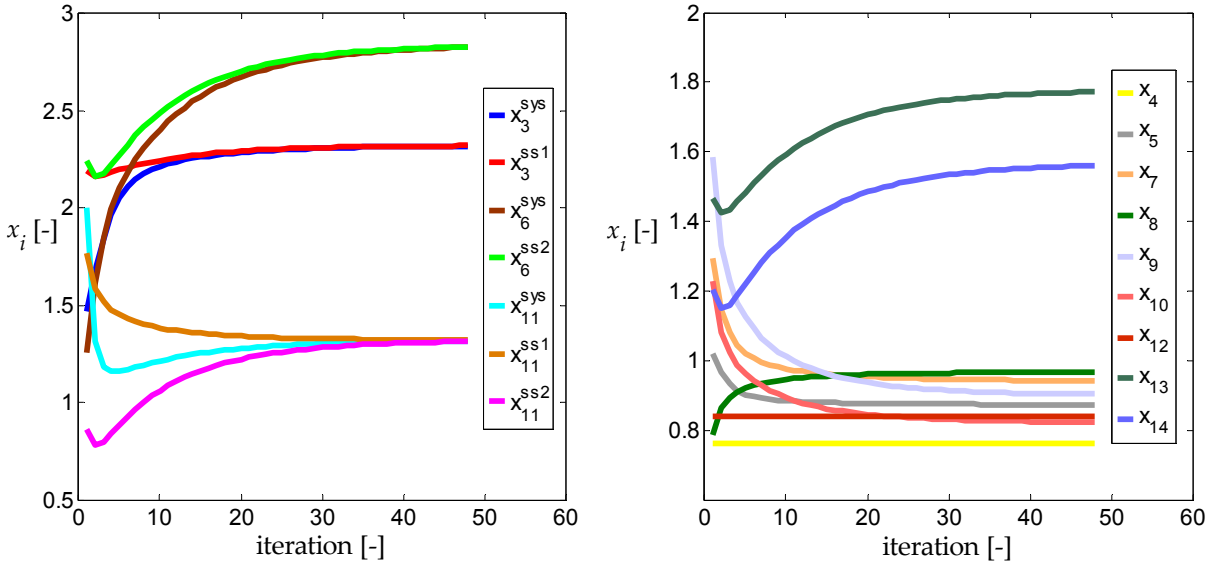


a) Objective function history b) Linking variables consistency history

Figure 3.19 : Objective function and linking variables history representation

An important characteristic to be remarked from this representation is that the ATC optimization process starts with designs having a value for the objective function inferior to the one of the optimum design. During the ATC iterations, the value of the objective function increases, converging towards the value of the optimal design. On the other hand, the ATC optimal design process starts with an inconsistent design and the inconsistency, given by the discrepancy between the targets and responses exchanged between the different sub-systems of the ATC structure, is decreased along iterations, as can be seen from Figure 3.19b.

The evolution of the two types of variables of the ATC process, linking and local variables, is presented graphically in Figure 3.20 a and Figure 3.20b.



a) Linking variables evolution b) Local variables evolution

Figure 3.20 : Design variables evolution history using ATC with the AL-AD method

From the linking variables evolution presented in Figure 3.20a it can be remarked the exchanged variables converge to the values of the optimal design, with a consistency tolerance of 10^{-2}, obtained after 48 ATC iterations. The same convergence can be remarked from the evolution of the local design variables represented in Figure 3.20b.

Based on the obtained results, the evolution of the ATC optimization process can be followed using two distinct measures concerning:

a) the consistency of the design at a given iteration k of the ATC optimization process;

b) the convergence of the ATC design optimization process, followed with regard to the design inconsistency evolution between two successive iterations.

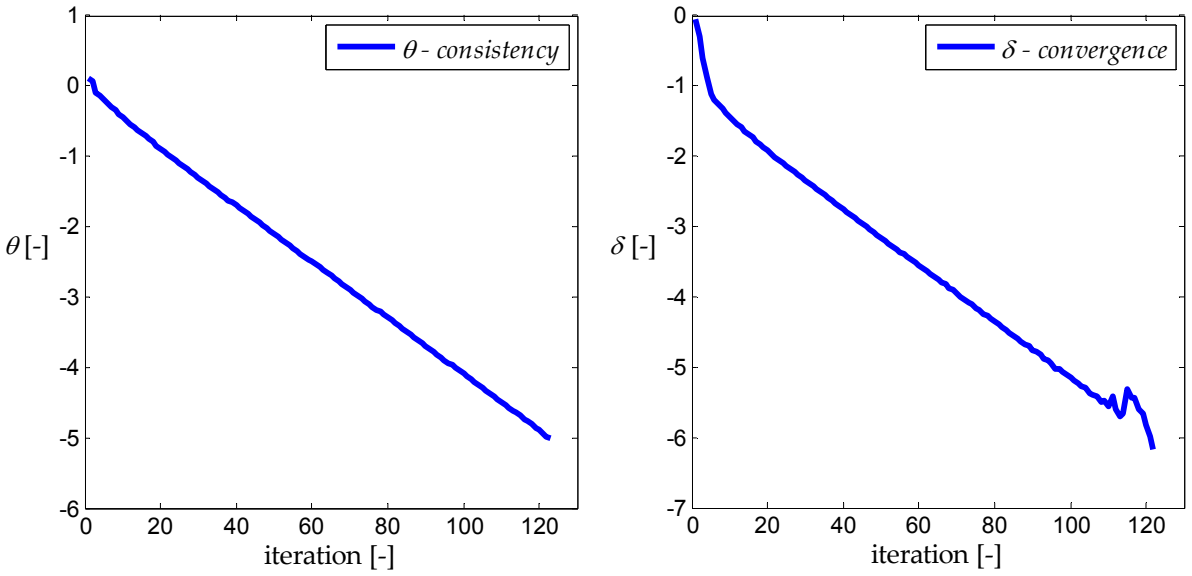
At a given iteration k of the ATC optimization process, the consistency of the design can be quantified by introducing the measure expressed in (3.35).

$$\theta^{(k)} = \log_{10}\left(\left\|\mathbf{c}^{(k)}\right\|_{\infty}\right) = \log_{10}\left(\left\|\mathbf{t}^{(k)} - \mathbf{r}^{(k)}\right\|_{\infty}\right) \tag{3.35}$$

This measure expresses the degree of (in-)consistency at a given iteration. The convergence of the ATC process, given by the evolution of the consistency between two consecutive ATC iterations can be followed using another measure, expressed in (3.36).

$$\delta^{(k)} = \log_{10}\left(\left\|\mathbf{c}^{(k)} - \mathbf{c}^{(k-1)}\right\|_{\infty}\right) = \log_{10}\left(\left\|\left(\mathbf{t}^{(k)} - \mathbf{r}^{(k)}\right) - \left(\mathbf{t}^{(k-1)} - \mathbf{r}^{(k-1)}\right)\right\|_{\infty}\right) \tag{3.36}$$

The two measures previously introduced have been calculated for the analytical test-problem optimization using the ATC with AL-AD method and their evolution during the optimization process is represented graphically in Figure 3.21 a and Figure 3.21b.



a) Design consistency measure θ b) ATC convergence measure δ

Figure 3.21 : Design consistency θ and ATC convergence δ measures representation

Looking at the consistency and convergence measures history in Figure 3.21, one can remark a linear trend of both these variables, which corresponds therefore to an exponential convergence of the design variables, due to the presence of the logarithm in the formulation of both θ and δ.

The design consistency evolution during the ATC optimization is resumed in Table 3.4.

Table 3.4 : Design consistency attainment for the analytical problem solved by ATC with AL-AD

Consistency, θ	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}
ATC iteration	23	48	73	98	123

In many cases, especially for practical applications, a design consistency of 10^{-2} can be considered sufficient and the algorithm can be stopped after a relatively reduced number of ATC iterations, hence saving considerable computation time.

The evolution of the second introduced measure, regarding the convergence of the ATC process, is resumed in Table 3.5.

Table 3.5 : ATC process convergence attainment for the analytical problem

Convergence, δ	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
ATC iteration	5	22	47	72	96	123

From the two summarizing tables, Table 3.4 and Table 3.5, it can be remarked a strong correlation between the values of the two measures, θ and δ. At a given iteration of the ATC process, a coefficient of 10 is present between the two measures.

Similar optimization test runs of the same analytical problem using different initial start points and other geometrical problems confirmed the remarks previously presented. This ATC implementation is shown to be very robust and does not need any parameter setting. The greatest

advantage represents however the fact that it is capable of finding the optimal solution for multi-level optimization formulations implying unattainable targets.

3.5.8 Existing efficiency enhancements of the ATC process

The computational burden given by the many iterations of the ATC process necessary for its convergence, combined with the complexity given by the additional linking variables introduced into all sub-problems seem to be an important issue and a lot of research has been concentrated on different ways of reducing this computational effort.

One factor which can significantly alter the convergence of different local optimization problems of elements from the ATC architecture, therefore indirectly the convergence of the ATC optimization process, is the presence of noise in the outputs of models at different places in the hierarchy of the ATC process. To overcome this impediment, Choudhary et al. proposed in [CHO 05] the use of a derivative-free optimization algorithm, from the class of MBDO strategies, Super-EGO [SAS 02a] to handle the non-smoothness of the simulation models from the lower level of the ATC hierarchy. They found that using ATC with different optimization solutions, adapted to the kind of model used at each element of the hierarchy resulted in a better solution than using an all-in-one optimization. The ATC formulation was used for solving a building design problem.

In order to ease the additional computational burden introduced by the linking variables, more precisely the shared variables between different sub-problems situated at the same level, Guarnieri et al. proposed in [GUA 11] the use of a modified version of SQP for solving the top level optimization problem. They exploit the specificity of the ATC information flux between sub-problems, particularly the fact that the top-level simulation model does not depend on variables shared between its children; copies of these variables are only used at top level to coordinate the convergence between lower-level sub-problems. Therefore, they exclude from the top-level fine model simulation the useless gradient calculations depending on the shared variables, which are null. This proposition is particularly interesting for the case where the top-level simulation model is the most demanding of all elements of the hierarchy, e.g. when a functioning cycle is simulated within the top-level simulation model. The computational reduction is directly related on the number of shared variables and the number of children sharing these variables. This idea was tested on two different test-cases, a structural and an automotive example problem, and showed an important reduction of the number of total simulation model evaluations at the top level of the ATC hierarchy.

Another recent study addressing the reduction of the computational burden of the demanding high fidelity model simulations when the coupling variables consist of large-dimensional vectors, such as functional data, was conducted by Alexander et al. in [ALE 11]. Their idea consists of using reduced representations of the highly discretized functional data representing the coupling variables exchanged between different sub-problems, from a few hundreds to just a few variables. For this, they propose and analyze two different techniques of reducing the representation of the functional data, a radial-basis function artificial neural network (RBF-ANN) and a proper orthogonal decomposition (POD). Both techniques performed well for solving a bi-level decomposed electric vehicle powertrain optimization problem, with a faster ATC convergence for the former method. However, some problems appeared when using the POD reduction method,

related to the poorer approximation of the functional data and conducing to repeated simulation model crashes. To overcome this problem, additional constraints were introduced into the expression of the sub-problem optimization problem. An improved management of these additional constraints for the reduced representation variables, using a penalty value-based heuristic scheme, was then proposed in [ALE 11a]. Unlike the functional data, the reduced representation variables are abstract, with no physical meaning. Hence, finding proper constraint values for such variables in order to reduce the search space to only the feasible domain and avoid simulation crashes is not obvious. The idea they proposed consists in penalizing the objective function when these constraints are not satisfied and using a heuristic scheme in finding proper penalty values for these constraints.

As can be seen from all the above, the different ways of reducing the complexity of the optimization problems in the ATC hierarchy are very problem-dependent. They exploit the particularities of the optimization problem to be solved and those of the given decomposition scheme used with ATC. This is in some way natural, since the goal of the ATC strategy is to coordinate the information exchanged between the different optimization sub-problems. In the following section, a more generally applicable computational burden reduction method is proposed. As generally the ATC formulation proposes to realize the coordinated optimization of several sub-systems or physical components usually represented using high fidelity simulation codes (e.g. FEM, CFD, etc.) of an engineering product, the metamodel-based design optimization strategies proposed in the previous chapter seem adapted for solving the local optimization problems of the hierarchical ATC structure. The ways of integrating these MBDO strategies within the ATC architecture, as well as the potential gain of computational reduction are analyzed in the following paragraph.

3.5.9 Metamodel approximation integrated to the ATC formulation

For the integration of high fidelity simulation models within the ATC multi-level optimization process with a reasonable computational cost, the use of metamodels offers a good approach. The metamodel-based design optimization approaches (MBDO) presented in Chapter 2 are well-suited to be employed for solving the different optimization problems of each element of the ATC multi-level hierarchy.

The greatest advantage of metamodels consists in the fact that once created, they offer the estimation of the simulation model's outputs at a negligible computational cost. Thus, in this paragraph it is proposed the employment of the adaptive metamodel-based optimization technique presented in the previous chapter. The workflow describing the actions undertaken by the P_{ij} element of the hierarchy, at a given iteration of the multi-level ATC process using the adaptive metamodel approximation technique, is presented in Figure 3.22.

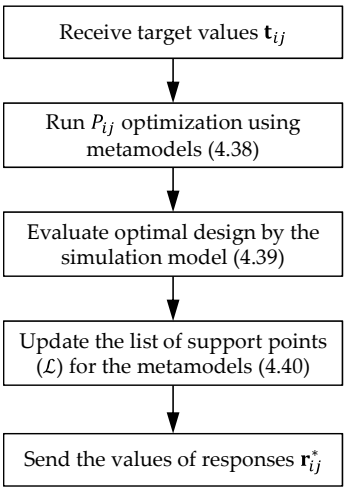


Figure 3.22 : Workflow of the P_{ij} element actions at a given iteration of the ATC process using the adaptive metamodel approximation technique

Prior to the ATC optimization process launch, for each simulation model employed by the hierarchical structure metamodels are created.

The new formulation of an optimization problem P_{ij} of the hierarchy is expressed in (3.37).

$$\begin{aligned}
&\underset{\bar{\mathbf{x}}_{ij}}{\text{Minimize}} && \hat{f}_{ij}\left(\bar{\mathbf{x}}_{ij}\right) + \pi\left(\hat{\mathbf{c}}_{ij}\left(\bar{\mathbf{x}}_{ij}\right)\right) \\
&\text{subject to} && \hat{\mathbf{g}}_{ij}\left(\bar{\mathbf{x}}_{ij}\right) \leq \mathbf{0} \\
& && \hat{\mathbf{h}}_{ij}\left(\bar{\mathbf{x}}_{ij}\right) = \mathbf{0} \\
&\text{with} && \bar{\mathbf{x}}_{ij} = \left[\mathbf{x}_{ij}, \mathbf{t}_{(i+1)k_1}, \cdots, \mathbf{t}_{(i+1)k_{c_{ij}}}\right] \\
& && \pi\left(\hat{\mathbf{c}}_{ij}\left(\bar{\mathbf{x}}_{ij}\right)\right) = \pi\left(\mathbf{t}_{ij}\left(\bar{\mathbf{x}}_{ij}\right) - \hat{\mathbf{r}}_{ij}\left(\bar{\mathbf{x}}_{ij}\right)\right) \\
& && \hat{\mathbf{r}}_{ij} = \hat{\mathbf{a}}_{ij}\left(\bar{\mathbf{x}}_{ij}\right)
\end{aligned} \tag{3.37}$$

where $\hat{\mathbf{a}}_{ij}$ represents the metamodel approximation of the analysis model, $\hat{\mathbf{r}}_{ij}$ are thus the metamodel-estimated outputs of the analysis, $\hat{f}_{ij}$ is the metamodel estimation of the objective function, $\hat{\mathbf{c}}_{ij}$ are the inconsistencies between the targets received and the metamodel estimations of the simulation model responses, $\hat{\mathbf{g}}_{ij}$ and $\hat{\mathbf{h}}_{ij}$ are the inequality, respectively equality constraint functions specific to the P_{ij} problem of the hierarchy.

At the end of the local optimization process, an optimal design $\bar{\mathbf{x}}^*{}_{ij}$ matching at its best the received targets is obtained. The design is evaluated then using the simulation model in order to obtain the true values of the P_{ij} problem's responses, as expressed by (3.38).

$$\mathbf{r}^*{}_{ij} = \mathbf{a}_{ij}\left(\bar{\mathbf{x}}^*{}_{ij}\right) \tag{3.38}$$

The optimal design $\bar{\mathbf{x}}^*{}_{ij}$ once evaluated using the simulation model it is added to the list of support points for the metamodels, according to (3.39), and new, improved, metamodels are created using the augmented list of support points.

$$\mathcal{L}_{ij}^{(k+1)} = \mathcal{L}_{ij}^{(k)} \cup \left\{ \left[\bar{\mathbf{x}}^*{}_{ij}, \mathbf{r}^*{}_{ij} \right] \right\} \tag{3.39}$$

This way, the metamodels prediction is improved from one ATC iteration to another. The ATC formulation implementing the adaptive metamodel approximation technique was employed for solving the bi-level optimization problem of an Alstom railway application consisting of the optimal design of an ultra-capacitor based energy storage system onboard a tramway (UC-ESS) presented later on in paragraph 3.8.

3.6 Multi-level optimization framework

Since multi-level optimization formulations imply the coordinated solving of several independent optimization problems, all these optimization processes, with the associated models of the different elements of the hierarchy and the information communication between the different optimizers must be coordinated in an efficient manner. This element is essential both for the comprehension of the coordinated optimization process and the success of the optimization process, since the amount of information exchanged, the number of files describing the models and the number of variables and outputs of the models is important. A high level of flexibility is required for these approaches, since several different decompositions of a design problem are to be implemented and tested. For this purpose, during the past decade, a number of generic decomposition-based optimization frameworks, both MDO and multi-level, have been developed by research scientists working in the field, such as Michelena et al. [MIC 99b], Huang et al. [HUA 06], and Tosserams et al. [TOS 10]. Others, such as Etman et al. [ETM 05], adapted and employed existing tools, initially conceived for other purposes, but which were suitable for this approach. The formalism used by the latter was initially developed for modeling purposes for simulating discrete events in manufacturing systems [ETM 05]. Different language specification programs or programming languages have been used for addressing the decomposition-based optimization, such as Cobra, Fortran, Python, and web-based platforms, such as Java and XML. These frameworks were developed independently by each team of scientists and are more or less generic, being developed and used for addressing particular optimization problems. At the present time however, no commercial tool dedicated to the decomposition-based optimization has been developed.

In this work, a Matlab® decomposition-based optimization framework has been developed and used for implementing and testing the mathematical test problems and the applications presented hereby. This choice was made mainly based on the author's programming skills and knowledge of the Matlab® software and the availability of most of the models for the electrical devices addressed and algorithms already implemented in Matlab®. The proposed platform is intended to be as generic as possible, and capable of handling multi-level optimization problems with different decompositions over several hierarchical architectures. The structure of the developed multi-level optimization framework is presented schematically in Figure 3.23.

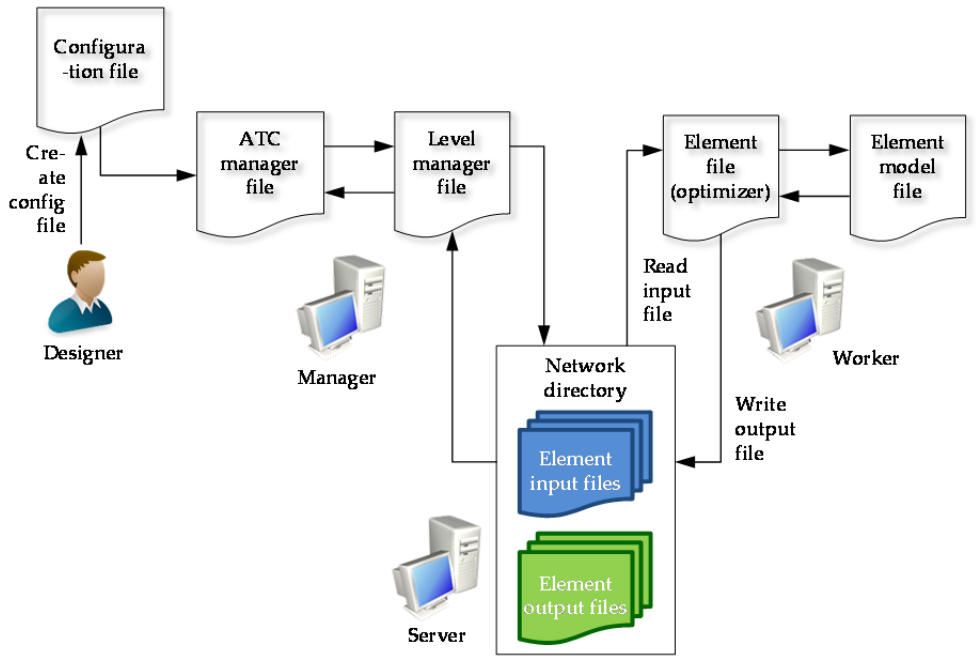


Figure 3.23 : The structure of the ATC multi-level optimization framework

The multi-level optimization platform has been conceived modular and in such a way to allow the evaluation of different models situated on different computers connected together over a network. Each element of the multi-level hierarchy can reside on a different computer. There is one manager computer that lodges the main application and controls the exchange between the different elements of the hierarchy. The functioning of the multi-level optimization platform can be described as follows:

The designer creates the configuration file of the multi-level decomposition structure of the problem to be solved. The main file of the multi-level optimization platform is the ATC manager file. It executes the main coordination loop (ATC coordination loop) by launching the execution of each level manager from the upper-most down to the lower-most level and back up. Once started the ATC manager file sends a start signal to all configured workers.

The level manager file is in charge with generating an input data file for each element of the multi-level hierarchy situated at that level. Once generated, the input data file is transferred onto a shared emplacement over the network, accessible to the other computers (workers) which are used for solving the local optimization problem and to evaluate the different element models. Then, the level manager file waits for the elements of that level to generate their output data files, by regularly scanning the shared network emplacement for the given output files.

On the other side, the defined workers, which foster the models of the different elements of the hierarchy, having received the start signal from the manager, wait for the appearance of the corresponding input data file in the network shared directory. Once appeared, the input data file is read by the element file and the local optimizer is launched using the model resident on that computer. At the end of the local optimization process, the results along with the information concerning the status of the local optimization process are written into an output data file. The new output file is then transferred onto the network shared directory, therefore being made available for

the level manager file. The local worker element file remains active, waiting for new input data of the next ATC iteration.

At the end of the multi-level optimization process, the ATC manager file send a stop signal to all workers implicated into the optimization process.

3.7 Application: Optimal design of a low-voltage single-phase safety-isolation transformer

The optimal design of a low-voltage single-phase safety-isolation transformer is addressed in this paragraph. The motivation for this optimal design study comes from the large quantity of such devices that are produced each year worldwide. An equally important decisional factor resides in the high penetration rate of similar optimal design applications in the field of electromagnetic converters. Firstly, the transformer optimization benchmark problem is presented and the models of the transformer which are used in the different optimization processes are briefly presented. Both the single-objective and the bi-objective optimization of the isolation transformer are addressed using the OSM technique presented in paragraph 3.2.2.1 and the results are compared to the ones obtained using the MEGO optimization algorithm presented in the previous chapter. The multi-disciplinary multi-level CO strategy presented in paragraph 3.2.2.2 is then addressed to account for the different coupled disciplines involved in the transformer modeling.

3.7.1 Transformer optimization benchmark

The optimal design problem of the safety-isolation transformer has been proposed as an optimization benchmark by Tran et al. in [TRA 07]. The optimal design problem involves 7 design variables: four variables defining the transformer's iron core geometry (a, b, c, d), one variable for the number of primary turns (n_1), and two variables for the sections of the enameled wires (S_1 and S_2) forming the primary respectively the secondary winding of the transformer. Six constraints, among which two regarding the thermal discipline, the copper and iron temperatures, T_{copper} respectively T_{iron}, three electromagnetic discipline constraints, regarding the ratio between the magnetizing current I_{10} and the primary current I_1, the ratio between the secondary voltage drop ΔV_2 and the secondary voltage V_2 and two geometrical constraints involving the filling factors of both windings, f_{f1} and f_{f2}, must be accounted for within the optimization processes. Two optimization problem formulations, a single-objective problem (SOP) and a bi-objective problem (MOP) formulation are expressed for the transformer design problem.

The mathematical formulations of both the single-objective problem (SOP) and the multi-objective problem (MOP) optimal design problems of the transformer are expressed in (3.40).

$$\begin{aligned} &\underset{\mathbf{x}}{\text{Minimize}} \quad F(\mathbf{x}) \\ &\text{where} \quad F(\mathbf{x}) = f(\mathbf{x}) = mass(\mathbf{x}) \qquad \text{for the single-objective (SOP) formulation} \\ &\qquad\quad F(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} mass(\mathbf{x}) \\ -\eta(\mathbf{x}) \end{bmatrix} \qquad \text{for the multi-objective (MOP) formulation} \end{aligned} \tag{3.40}$$

such as $T_{copper} \leq 120°C$ $T_{iron} \leq 100°C$ $f_{f1} \leq 0.5$ $\eta \geq 0.8$ (SOP only)

$\frac{I_{10}}{I_1} \leq 0.1$ $\frac{\Delta V_2}{V_2} \leq 0.1$ $f_{f2} \leq 0.5$

with $\mathbf{x} = [a, b, c, d, n_1, S_1, S_2]$

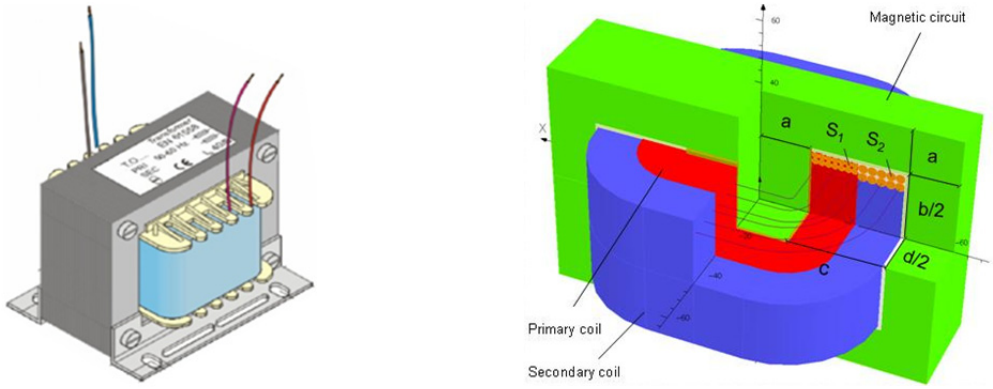
$a \in [3,30]mm$ $b \in [14,95]mm$ $c \in [6,40]mm$ $d \in [10,80]mm$

$S_1 \in [0.15,19]mm^2$ $S_2 \in [0.15,19]mm^2$ $n_1 \in [200,1200]$

The goal of the SOP formulation is to minimize the mass of the windings and the magnetic circuit of the transformer, with regard to the six physical and geometric constraints plus an additional constraint ensuring a minimal efficiency of the transformer.

The MOP formulation seeks to provide the designer not just one optimal design, but a set of compromise designs between the mass and the efficiency of the transformer. Hence, the result of the MOP optimization problem is a set of non-dominated designs forming the Pareto front.

The safety-isolation transformer to be optimized is presented in Figure 3.24.



a) Physical representation b) Geometrical variables representation

Figure 3.24 : Safety-isolation transformer to be optimized

3.7.2 *Transformer representation*

The safety-isolation transformer is represented using two models of different accuracy: a fine–numerical 3D FE model, respectively a coarse–analytical model. The high accuracy 3D FE model developed by Tran et al. using the commercial software Opera3D® is available from [TRA 07a]. This is a multidisciplinary FE model addressing the electromagnetic and the thermal aspects and it consists of a coupling model between two FE electromagnetic models (at full-load and no-load) and a thermal FE model, all constructed and evaluated using the Opera3D® software. The management of the coupling between the different models is handled using a MDF formulation and the iterative loop is solved using a fixed point iteration (FPI) method. The coupling between the two disciplines involved in the FE representation of the isolation transformer is presented in Figure 3.25.

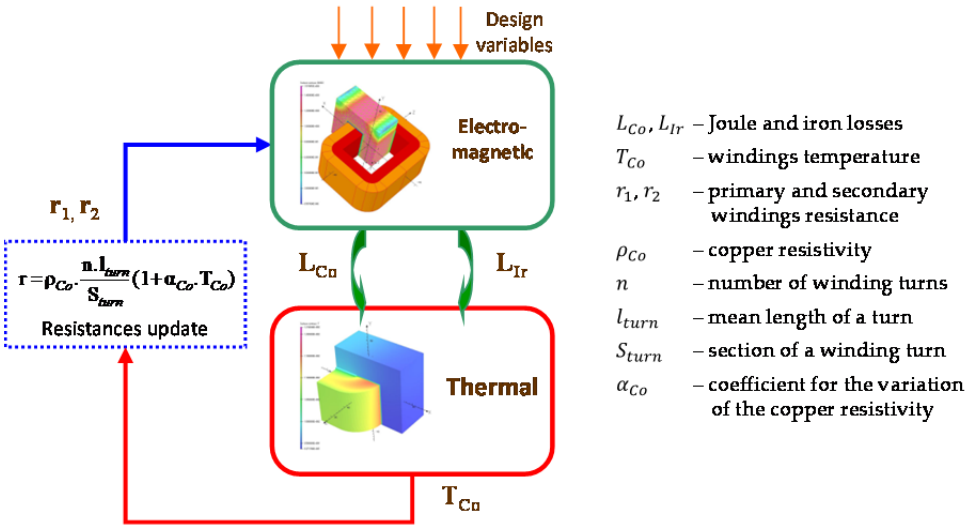


Figure 3.25 : Coupling between the electromagnetic and the thermal FE models of the transformer

Equally, an analytical model of the transformer is available under Matlab [TRA 07a]. This model is less accurate than the 3D finite element one, but it is able to capture the trends of the different outputs of the transformer. In exchange for its reduced accuracy, the analytical model benefits from a fast evaluation which represents an important desired feature for any classical optimization algorithm.

Both the FE and the analytical model of the transformer, as well as the full description of the transformer and the different optimization formulations are available in [TRA 09].

3.7.3 OSM – MEGO isolation transformer optimization comparison

The OSM multi-model optimization technique presented in paragraph 3.3 is employed here for the optimal dimensioning of the safety isolation transformer. Since a number of similarities between OSM and the MEGO algorithm developed in the previous chapter have been previously identified, a comparison of the two techniques on both single and multi-objective basis seems judicious and therefore will be addressed here next. The single-objective optimization of the transformer was addressed in [BEN 11]. The promising results of this preliminary comparison motivated a more general multi-objective comparison of the two optimization approaches, which was then addressed in [BER 12].

Prior to addressing the optimization of the isolation transformer, the influence of the different nonlinear constraint functions on the optimization problem needs to be analyzed. For this, the analytical model of the transformer has been considered, giving its satisfying accuracy. A very large experimental design was considered using a Latin Hypercube strategy (LHS of 10 000 designs) over the entire decision space. The analytical model of the transformer was then used to compute the values of the nonlinear constraint functions for the designs of the previously generated LHS. As a result, among the 10 000 designs of the LHS, a number of only 8 designs were found to satisfy all the 7 constraints of the optimization problem (0.08% of the total number of generated designs). A number of 99 designs satisfy the geometrical constraints (~1%), while 6283 designs satisfy the

different physical constraints of the optimization problem. The results show a large infeasible sub-space of the whole design space, and this is mainly due to the geometrical constraints. In consequence, a special care should be given to the handling of the constraint functions.

3.7.3.1 Single-objective transformer optimization

The OSM technique was shown to have a very good performance when applied to handling highly constraint optimization problems [BEN 11]. Therefore, its application here to the optimal design of the safety isolation transformer is well suited, since the transformer benchmark problem had been shown to possess a very narrow feasible decision sub-domain. Moreover, the analytical model of the transformer captures roughly the trends and tendencies of the fine model's outputs. The OSM technique consists in progressively aligning the analytical model to the FE model of the transformer in the vicinity of the optimal design predicted by the analytical model. Among the 8 outputs of the transformer's models, only 5 of them ($\eta, T_{copper}, T_{iron}, \frac{I_{10}}{I_1}, \frac{\Delta V_2}{V_2}$) will be accounted for within the space mapping process. The other 3 outputs ($mass, f_{f1}, f_{f2}$) are calculated analytically by both models of the transformer, producing the same values for both models and thus no correction is needed for these outputs.

The second considered optimization approach, the EGO algorithm is much more sensitive to the presence of the constraint functions in the expression of the optimization problem. The different EGO constraint handling techniques reviewed in Chapter 2 have been shown to produce acceptable optimal solutions when dealing with a reduced number of constraints [SAS 02]. Being mainly an intelligent sampling technique, EGO requires a large-enough feasible sub-domain of the design space in order to be able to correctly construct the *EI* infill criterion prediction. The EGO algorithm has been tested for the optimization of the transformer using different infill criteria (*PI*, *EI*, *GEI*) and constraint handling techniques (penalty, *PF*, *EV*) presented in the previous chapter. However, due to the strongly constraint character presented by the transformer optimization benchmark problem, no exploitable results have been produced with an acceptable number of FE model evaluations by the classical EGO technique. Hence, a different approach has been considered, consisting in applying a transformation technique to the initial single-objective transformer optimization problem formulation. A weighted formulation [BRI 07], [NEI 96], which transforms the single-objective constrained problem into a bi-objective unconstrained one, has been employed here. The 7 constraint functions of the initial optimization problem are aggregated into a penalty function ($f_{penalty}$) using the formulation given in (3.41).

$$f_{penalty} = \sum_{k=1}^{n_c} \max_{n_c}\left(0, g_k(\mathbf{x})\right) \tag{3.41}$$

where $g_k(\mathbf{x})$ represents the k-th constraint function of the initial optimization problem and n_c represents the total number of constraint function, equal to 7.

Instead of minimizing the total mass of the device, the newly formulated optimization problem consists of minimizing both the mass and the penalty function. The optimization problem takes the form expressed in (3.42).

$$\begin{aligned} &\underset{\mathbf{x}}{\text{Minimize}} \quad F(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} mass(\mathbf{x}) \\ f_{penalty}(\mathbf{x}) \end{bmatrix} \\ &\text{with} \quad \mathbf{x} = [a, b, c, d, n_1, S_1, S_2] \\ &\qquad f_{penalty}(\mathbf{x}) = \sum_{k=1}^{n_c} \max_{n_c}\left(0, g_k(\mathbf{x})\right) \end{aligned} \tag{3.42}$$

The newly formulated bi-objective optimization problem is addressed now by the multi-objective MEGO algorithm developed in the previous chapter. Instead of obtaining a single optimal design, as with the OSM technique, the MEGO algorithm supplies the designer with a set of non-dominated trade-off designs forming a Pareto front. Ideally, the penalty function is null for all designs respecting the constraint functions of the initial optimization problem formulation. A slight constraint overpassing might be accepted however. The resulting optimal $mass$-$f_{penalty}$ Pareto front is presented in Figure 3.26.

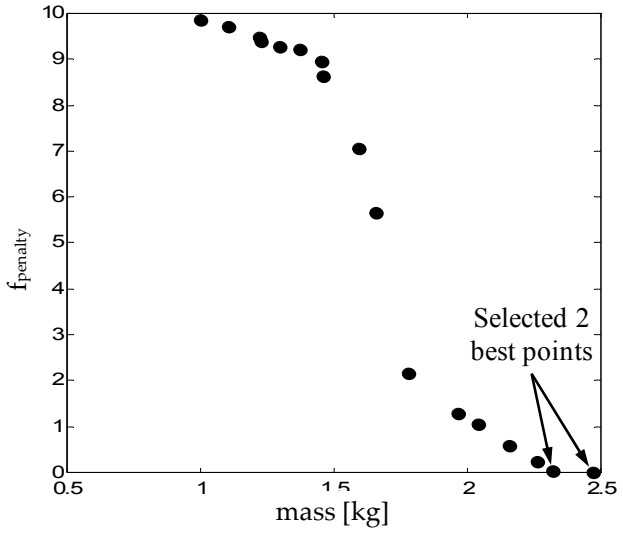


Figure 3.26 : Transformer compromise solutions $mass - f_{penalty}$ obtained by the MEGO algorithm

The final choice of the optimal design belongs in this case to the designer, which will seek the design or those designs which present a low value of total mass and at a same time present no or an acceptable value of constraint overpassing ($f_{penalty}$). Here, the attention has been focused on two designs belonging to the Pareto front, both presenting a low value of total mass and satisfying the constraint functions of the initial optimization problem within an acceptable low tolerance value. The first considered design, the right-most one in Figure 3.26, corresponds to a low value of mass and entirely satisfies the constraints ($f_{penalty} = 0$). The second interesting design, found right next to the first one, offers a lower value of mass, but with a slight constraint overpassing in exchange ($T_{iron} = 104.24°C > 100°C$). The values of these two interesting designs issued from the MEGO's Pareto front are presented for comparison with the optimal design obtained by the OSM technique and another design, obtained through optimization using a classical trust-region optimization algorithm, SQP, with the FE model in Table 3.6. Due to the max function in the expression of the penalty function (3.41), all the designs which satisfy the constraints are awarded a zero penalty function value, with no regard to the degree of constraint satisfaction. This formulation causes

therefore the algorithm to face some difficulties in determining the global optimum design with precision.

Table 3.6 : Comparison of OSM, MEGO and SQP optimal transformer results

Variables	SQP	OSM		MEGO		Constraint Bounds
$a(mm)$	13.2	12.98		12.02	9.21	
$b(mm)$	49.9	50.7		55.46	51.96	
$c(mm)$	17.25	16.72		19.39	21.13	
$d(mm)$	43.260	43.41		40.84	57.56	
$n_1(-)$	630.94	640.01		725.7	675.86	
$S_1(mm^2)$	0.335	0.331		0.35	0.34	
$S_2(mm^2)$	2.994	2.964		2.782	3.493	
		AM	3D FEM	1st best point	2nd best point	
$T_{copper}(°C)$	<u>120.04</u>	108.41	119.42	113.44	107.07	120
$T_{iron}(°C)$	99.8	100	100	<u>104.24</u>	98.79	100
$\Delta V_2/V_2$	0.0754	0.064	0.0751	0.0819	0.0789	0.1
I_{10}/I_1	0.100	0.1	0.0999	0.0974	0.0929	0.1
η	0.8938	0.891	0.8934	0.8845	0.8906	0.8
mass (kg)	2.3755	2.355	2.355	2.3206	2.4764	min
3D FEM evaluations	1793	15		150		
Max. constraint violation	5.6e-4	0		4.2e-2	0	
Time (seconds)	218755	1823		18150		

The single-objective optimization problem of the transformer was successfully handled by both considered optimization techniques. Since the single-objective EGO approach could not provide the optimal solution within a reasonable number of FE model evaluations due to the highly constraint character of the problem, the multi-objective MEGO algorithm developed in the previous chapter was employed to solve an unconstrained bi-objective optimization problem. The obtained results are thus close to the global optimum of the transformer problem, obtained by the OSM technique and confirmed by the classical SQP approach, as can be seen from Table 3.6.

Considering the number of FE model evaluations required by the analyzed approaches, it can be seen that the OSM technique has outperformed the MEGO approach by a factor of 10, and the classical SQP approach by a much more penalizing factor of 120, as can be observed from Table 3.6.

The performance of the OSM technique is in direct relation to the quality of the coarse model it employs to perform the optimization. The analytical model of the transformer shows a decent accuracy, resulting in an advantage of the OSM technique over MEGO in the optimization process. Hence, the global optimum of the transformer problem is obtained by OSM with an extremely

reduced number of only 15 FE model evaluations. Moreover, the MEGO algorithm employed for solving the transformer problem makes uses of a transformation technique in order to account for the constraints. The handled problem is bi-objective, therefore a higher number of FE model evaluations were required to determine the Pareto frontier. The MEGO approach provided a set of $mass - f_{penalty}$ compromise solutions among which two designs draw the attention and were selected for presentation in Table 3.6, one of them respecting entirely the constraints but with a superior mass value and the other one, presenting an inferior mass, but with a slight constraint violation.

3.7.3.2 Bi-objective transformer optimization

The same transformer optimization benchmark previously analyzed is used here within a bi-criteria optimization framework. The goal of the bi-objective optimization is to find those transformer designs which represent an optimal trade-off between the total mass and the efficiency of the transformer. The result of the bi-objective optimization this time is no longer a unique optimal design, but a set of Pareto non-dominated designs representing the optimal $mass - \eta$ compromise.

The OSM technique realizes a punctual correction of the coarse analytical model using the fine FE model. In order for OSM to account for both objectives of the optimization problem, a different transformation technique is set up, which is the epsilon-constraint method presented in the first chapter of the manuscript. The bi-objective optimization problem is transformed thus into a single-objective problem, by keeping the efficiency η as a unique objective, and considering the $mass$ as an additional constraint. The rephrased optimization problem is expressed in (3.43).

$$\begin{aligned} \underset{\mathbf{x}}{\text{Minimize}} \quad & f(\mathbf{x}) = -\eta(\mathbf{x}) \\ \text{with} \quad & \mathbf{x} = [a, b, c, d, n_1, S_1, S_2] \\ \text{subject to} \quad & g_k(\mathbf{x}) \leq 0 \qquad k = 1 \cdots n_c \qquad n_c = 6 \\ & g_{add}(\mathbf{x}) = mass(\mathbf{x}) - \varepsilon \leq 0 \end{aligned} \tag{3.43}$$

where $g_k(\mathbf{x})$ represents the k-th constraint function of the original bi-objective optimization problem, $g_{add}(\mathbf{x})$ represents the additional $mass$ constraint introduced after problem rephrasing, and ε represents different limit levels of $mass$, being used in order to spread the optimal trade-off solutions along the Pareto front.

For each imposed value of mass limit ε, a new optimization is launched using the OSM algorithm and a new compromise solution belonging to the Pareto front is obtained. A total of 20 optimal solutions forming the Pareto non-dominated front were obtained by the successive launching of the OSM algorithm with different mass constraint levels ε. The obtained Pareto front is presented in Figure 3.28 for comparison with the solutions obtained by the MEGO approach.

The second analyzed optimization technique, MEGO was equally employed for solving the multi-objective transformer optimization problem. As for the single-objective case presented in paragraph 3.7.3.1, the original constrained optimization problem is handled by MEGO as an unconstrained optimization problem, allowing thus a certain amount of constraint violation. Expressing the optimization problem as an unconstrained one allows indirectly eliminating any

constraint handling difficulties of MEGO. The different constraint functions of the original problem are aggregated under a normalized penalty function, which is accounted for as an additional objective function to be minimized by the MEGO algorithm. The rephrased optimization problem involves this time 3 antagonistic objective functions. The expression of the rephrased optimization problem to be handled by MEGO is given in (3.44).

$$\begin{aligned} &\underset{\mathbf{x}}{\text{Minimize}} \quad F(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ f_3(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} mass(\mathbf{x}) \\ -\eta(\mathbf{x}) \\ f_{penalty}(\mathbf{x}) \end{bmatrix} \\ &\text{with} \quad \mathbf{x} = [a, b, c, d, n_1, S_1, S_2] \\ &\qquad \mathbf{y} = \left[T_{copper}, T_{iron}, \frac{I_{10}}{I_1}, \frac{\Delta V_2}{V_2}, f_{f1}, f_{f2}\right] \\ &f_{penalty}(\mathbf{x}) = \sum_{k=1}^{n_c} g_k(\mathbf{x}) = \frac{y_1(\mathbf{x})}{120} + \frac{y_2(\mathbf{x})}{100} + \frac{y_3(\mathbf{x})}{0.1} + \frac{y_4(\mathbf{x})}{0.1} + \frac{y_5(\mathbf{x})}{0.5} + \frac{y_6(\mathbf{x})}{0.5} - 6 \end{aligned} \tag{3.44}$$

The result of the MEGO optimization process is this time a 3D Pareto front opposing the two objectives of the initial bi-objective problem and the additional objective representing the amount of constraint violation. The obtained 3D Pareto front is presented in Figure 3.27.

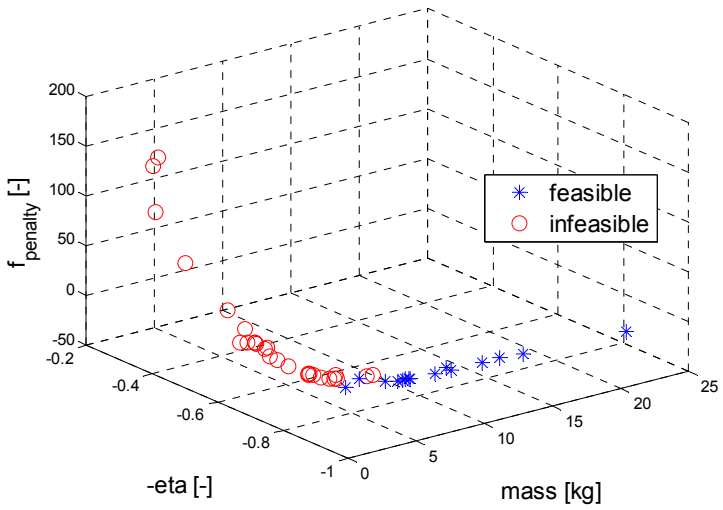


Figure 3.27 : 3D Pareto front of the transformer benchmark obtained by the MEGO algorithm

The blue stars in Figure 3.27 represent the feasible designs of the Pareto front, while the rest of designs, represented by the red circles, present a certain amount of constraint violation. The non-dominated solutions of the initial bi-objective constrained optimization problem are obtained by extraction of the feasible solutions out of the obtained 3D Pareto front. The optimal results are represented in Figure 3.28.

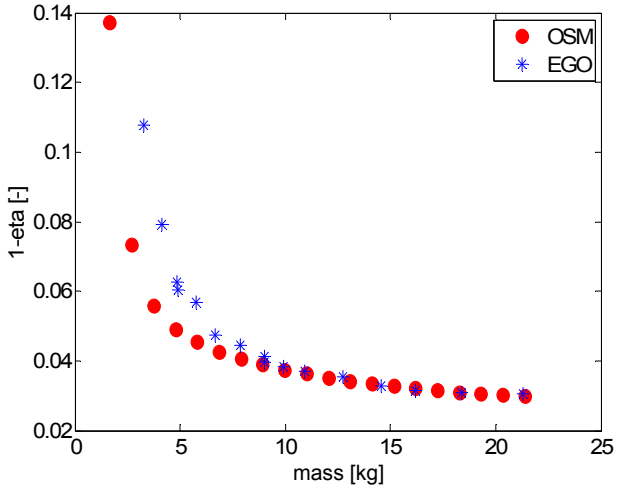


Figure 3.28 : Transformer Pareto front $(mass\ vs.\ (1-\eta))$ comparison of OSM and MEGO

From the optimal results presented in Figure 3.28 it can be seen that both Pareto fronts obtained by OSM and MEGO, have approximately the same number of solutions, 20 respectively 19 designs, for ~2.5 times more FE model evaluations for MEGO (124, respectively 300). Visually, it can be remarked that the Pareto front obtained using the OSM approach is slightly better than the one obtained using the MEGO algorithm. A lighter distribution of points is observed on the Pareto front of the OSM algorithm in the area corresponding to lighter mass designs. This is due to the epsilon-constraint transformation technique employed by OSM to account for the two objectives of the problem.

3.7.4 Multi-disciplinary optimization of the transformer by CO

Classical single-level multi-disciplinary (MDO) formulations have been already employed with success for solving the isolation transformer optimal design problem by Ben-Ayed et al. in [BEN 12]. The attention is focused here on the multi-level optimal design of the same transformer application, this time using the CO multi-disciplinary method presented previously in paragraph 3.4. The analytical model of the safety isolation transformer has been used within the CO optimization process. The multi-disciplinary representation of the isolation transformer with the interactions between the different disciplinary models is presented in Figure 3.29.

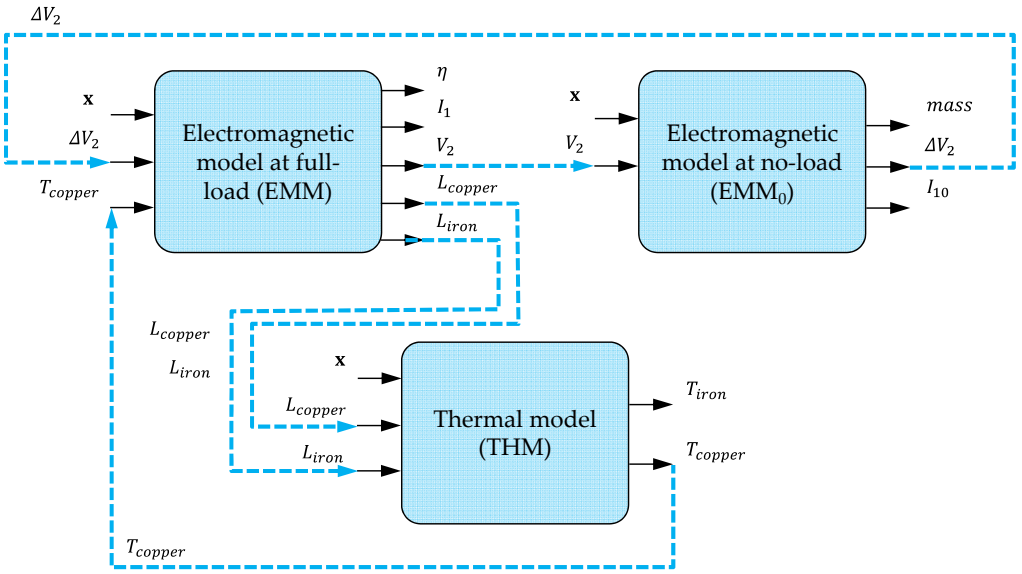


Figure 3.29 : Multi-disciplinary isolation transformer representation

The isolation transformer is represented by three distinct disciplinary models, a full-load electromagnetic model (EMM), a no-load electromagnetic model (EMM$_0$) and a thermal model (THM), as can be seen from Figure 3.29. The EMM$_0$ requires the value of V_2, calculated by EMM, in order to compute its outputs. The THM calculates its temperature outputs based on L_{copper} and L_{iron} computed by the EMM. The EMM requires information both from EMM$_0$ and THM, ΔV_2 respectively T_{copper} in order to compute its outputs. The decision vector $\mathbf{x}$ contains the seven geometrical design variables and it is common input to all three disciplinary models.

Since the same variable is the output of a disciplinary model and input for another discipline, a local copy of each exchanged variable is used by each disciplinary model. In order to have a consistent design, the values of all exchanged variables must be consistent. Under the multi-level CO strategy, the different disciplinary variables exchanged between the three models are handled through the expression of interdisciplinary consistency constraints. The system level optimization of the CO formulation has the role of coordinating the exchanges between the different disciplinary optimizations and driving the design towards consistency, by iteratively matching the copies of all exchanged variables. The expression of the system-level optimization problem is given in (3.45).

$$
\begin{aligned}
P_{sys}: \quad & \underset{\mathbf{y},\mathbf{z}}{\text{Minimize}} \quad f(\mathbf{y},\mathbf{z}) = mass(\mathbf{y},\mathbf{z}) \\
& \text{with} \quad \mathbf{z} = [a,b,c,d,n_1,S_1,S_2] \\
& \qquad\quad \mathbf{y} = [\mathbf{y}_1,\mathbf{y}_2,\mathbf{y}_3] = \left[[\Delta V_2, I_{10}],[V_2, L_{co}, L_{ir}],[T_{co}]\right] \\
& \text{subject to} \quad \boldsymbol{J}^* = [J_1^*(\mathbf{z}^*,\mathbf{z},\mathbf{y}_1,\mathbf{c}_1), J_2^*(\mathbf{z}^*,\mathbf{z},\mathbf{y}_2,\mathbf{c}_2), J_3^*(\mathbf{z}^*,\mathbf{z},\mathbf{y}_3,\mathbf{c}_3)] \leq \boldsymbol{\varepsilon}
\end{aligned}
\tag{3.45}
$$

where $\boldsymbol{\varepsilon}$ represents the predefined tolerance for the consistency constraints satisfaction, with its value fixed at 10^{-3}, $\mathbf{z}$ is the design variables vector, $\mathbf{y}$ is the vector of disciplinary coupling variables, $\mathbf{z}^*$ is the vector of optimal design variables from the discipline level, $\mathbf{c}_i$ is the vector of

optimal coupling variables from the *i*-th discipline optimization and J_i^* is the optimal objective function value of the *i*-th discipline optimization.

The formulation of the no-load EM discipline optimization is expressed in (3.46).

$$
\begin{aligned}
P_{EMM_0}\text{:}\quad & \underset{\mathbf{z}_1^*}{\text{Minimize}} && J_1(\mathbf{z}_1^*, \mathbf{z}_1, \mathbf{y}_1, \mathbf{c}_1) = \|\mathbf{z}_1 - \mathbf{z}_1^*\|_2^2 + \|\mathbf{y}_1 - \mathbf{c}_1\|_2^2 \\
& \text{with} && \mathbf{z}_1^* = [a^*, b^*, c^*, d^*, n_1^*, S_1^*, S_2^*] \\
& && \mathbf{c}_1 = [\Delta V_2, I_{10}] \\
& \text{subject to} && \mathbf{g}_{1,1}(\mathbf{z}_1^*) = \Delta V_2 / V_2 - 0.1 \leq 0 \\
& && \mathbf{g}_{1,2}(\mathbf{z}_1^*) = I_{10} / I_1 - 0.1 \leq 0
\end{aligned}
\tag{3.46}
$$

The no-load EM optimization problem seeks to match the values of the design variables and coupling variables received from the system-level optimizer, while respecting its specific constraints related to the secondary voltage drop $\Delta V_2$and magnetizing current I_{10}.

The formulation of the full-load EM discipline optimization is expressed in (3.47).

$$
\begin{aligned}
P_{EMM}\text{:}\quad & \underset{\mathbf{z}_2^*}{\text{Minimize}} && J_2(\mathbf{z}_2^*, \mathbf{z}_2, \mathbf{y}_2, \mathbf{c}_2) = \|\mathbf{z}_2 - \mathbf{z}_2^*\|_2^2 + \|\mathbf{y}_2 - \mathbf{c}_2\|_2^2 \\
& \text{with} && \mathbf{z}_2^* = [a^*, b^*, c^*, d^*, n_1^*, S_1^*, S_2^*] \\
& && \mathbf{c}_2 = [V_2, L_{co}, L_{ir}] \\
& \text{subject to} && \mathbf{g}_{2,1}(\mathbf{z}_2^*) = 0.8 - \eta \leq 0 \\
& && \mathbf{g}_{2,2}(\mathbf{z}_2^*) = f_{f1} - 0.5 \leq 0 \\
& && \mathbf{g}_{2,3}(\mathbf{z}_2^*) = f_{f2} - 0.5 \leq 0
\end{aligned}
\tag{3.47}
$$

Apart from its discipline-specific feasibility constraints, the full-load EM optimization problem was also allocated the two geometric constraint functions of the transformer optimization problem, relative to the primary and secondary winding filling factors, f_{f1} respectively f_{f2}.

The formulation of the TH discipline optimization is expressed in (3.48).

$$
\begin{aligned}
P_{THM}\text{:}\quad & \underset{\mathbf{z}_3^*}{\text{Minimize}} && J_3(\mathbf{z}_3^*, \mathbf{z}_3, \mathbf{y}_3, \mathbf{c}_3) = \|\mathbf{z}_3 - \mathbf{z}_3^*\|_2^2 + \|\mathbf{y}_3 - \mathbf{c}_3\|_2^2 \\
& \text{with} && \mathbf{z}_3^* = [a^*, b^*, c^*, d^*, n_1^*, S_1^*, S_2^*] \\
& && \mathbf{c}_3 = [T_{co}] \\
& \text{subject to} && \mathbf{g}_{3,1}(\mathbf{z}_3^*) = T_{co} - 120 \leq 0 \\
& && \mathbf{g}_{3,2}(\mathbf{z}_3^*) = T_{ir} - 100 \leq 0
\end{aligned}
\tag{3.48}
$$

The thermal discipline (TH) optimization is in charge with respecting the maximum winding and magnetic core temperatures, T_{co} respectively T_{ir}.

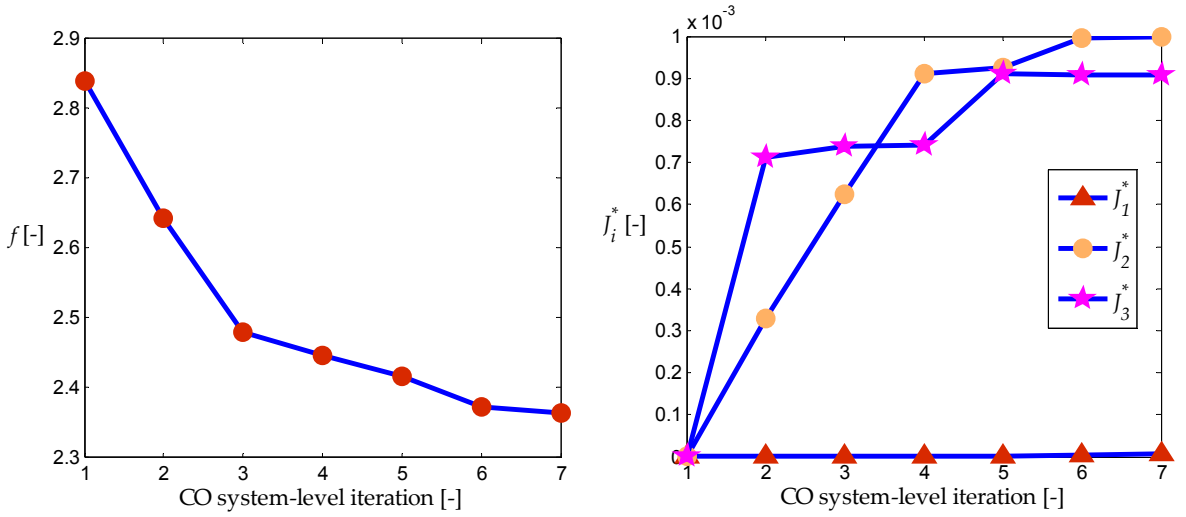
The optimization process using the CO formulation is started from an initial feasible design, and using the different disciplinary models calculated analytically. The results of the optimization process are presented in Table 3.7 along with the results of another optimization run using the SQP algorithm and considered as reference.

Table 3.7 : Optimal results comparison of the CO multi-level optimization process and SQP

Algo-rithm	**Design variables**							**Obj.**	**Constraints**				
	a	b	c	d	n_1	S_1	S_2	$mass$	T_{copper}	T_{iron}	$\Delta V_2/V_2$	I_{10}/I_1	η
	$[mm]$	$[mm]$	$[mm]$	$[mm]$	$[-]$	$[mm^2]$	$[mm^2]$	$[kg]$	$[°C]$	$[°C]$	$[-]$	$[-]$	$[-]$
SQP	12.9	50.1	16.6	43.2	640	0.32	2.91	2.31	108.8	99.9	0.07	0.1	0.89
CO	16.5	53.3	18.1	31.0	728	0.31	2.51	2.36	104.9	95.9	0.08	0.08	0.89

The convergence of the CO process using the analytical disciplinary models of the transformer was achieved in only 7 system-level iterations. An optimal mass value of 2.36kg was obtained with a predefined tolerance of 10^{-3} for the interdisciplinary consistency. This value is slightly superior to the global optimum obtained using the SQP algorithm, the difference being related to the predefined tolerance for the satisfaction of the interdisciplinary consistency constraints. For the convergence of the CO process, a total number of 128 system-level objective and constraint function evaluations were required. Each of these function evaluations corresponds to a complete disciplinary optimization of all disciplines.

The evolution of the system-level objective function value (the transformer mass) during the CO process is presented in Figure 3.30a. The evolution of the optimal disciplinary objective function values for the three disciplines implied is shown in Figure 3.30b. One can remark that the CO process starts with very low values for the disciplinary objective functions (corresponding to the feasible initial design) and these values increase along iterations, until attaining the predefined value for the consistency constraints satisfaction of 10^{-3}.



a) System-level objective function evolution b) Discipline-level objective function evolution

Figure 3.30 : System and discipline-level objective function evolution during the isolation transformer optimization process using CO

The results show a fast convergence of the CO process using the analytical models of the transformer, in terms of system-level optimization iterations. In order to benefit from the accuracy offered by the FE models of the isolation transformer, these accurate models must be introduced into the optimization process. However, their direct integration into the CO process, by simply replacing the call to the analytical models with a call to the analogue FE models, cannot be done without a consequent computational burden increase. Each of the 128 objective and constraint function evaluations spread over the 7 system-level optimization iterations implies one optimization run of each disciplinary model, thus the CO process required a total of 128 disciplinary optimizations. Considering an evaluation time of a few (1 to 5) minutes per each 3D disciplinary FE model evaluation, depending on the geometrical dimensions of the transformer and the mesh quality, and also considering ~200 disciplinary model evaluations required by a classical optimization algorithm, such as SQP, one can realize that a total of 128*200 3D FE model evaluations would be prohibitive.

In perspective, the integration of the accuracy provided by the 3D FE models of the transformer into the multi-disciplinary optimization process, at a reasonable computational cost is considered. A potential way of achieving this is represented by the implementation within the disciplinary level of the CO formulation of a space-mapping technique, the OSM version, idea submitted for presentation in conference, in [BER 12a].

3.8 Application: Optimal design of ultra-capacitor energy storage system (UC-ESS) on-board a tramway

The goal of the optimal design problem of an energy storage system (ESS) based on ultra-capacitors (UC) on-board a tramway (UC-ESS) is to conceive a transportation system with a high energy performance, hence more environment-friendly. Energy storage systems on-board tramways present two main functionalities. First, they allow passing the zones of the trajectory where the overhead contact line is unavailable, e.g. historical centers of cities, low-clearance passages under bridges and highways etc. Second, the ESSs allow improving the tramways energy performance by recovering a part of the energy resulting from braking and reusing it during tramways acceleration phases of the trajectory.

3.8.1 The ultra-capacitor energy storage system (UC-ESS)

A prototype of an UC-ESS, named STEEM (Maximized Energy Efficiency Tramway System) [MOS 10], has been developed by Alstom Transport and implemented by RATP on one tramway vehicle which is in circulation on a suburb line (T3) in the Paris region. The main goal of the STEEM prototype was to ensure the tramway's autonomy for the zones where the overhead contact line is unavailable due to the specificity of the zone i.e. historical city center or passage under low-clearance bridge. When the tramway arrives in this zone, the pantograph is lowered and the tramways traction power is supplied by the STEEM pack for the entire zone of autonomy, until the tramway arrives again under the catenary and the pantograph is raised again.

However, the STEEM prototype was dimensioned only to pass through the autonomy zone, regardless of its impact on the tramways energy consumption. Taking the STEEM prototype as start point, we will analyze here the possibility of reducing the tramways energy consumption from the catenary, while ensuring the tramways autonomy when the overhead contact line is unavailable.

The tramway on which the STEEM prototype is installed is presented schematically in Figure 3.31. The STEEM ultra-capacitor energy storage system (UC-ESS), consisting of a bank of 48 ultra-capacitor modules is installed on the roof of the tramway, outside, and is represented by the green area in Figure 3.31.

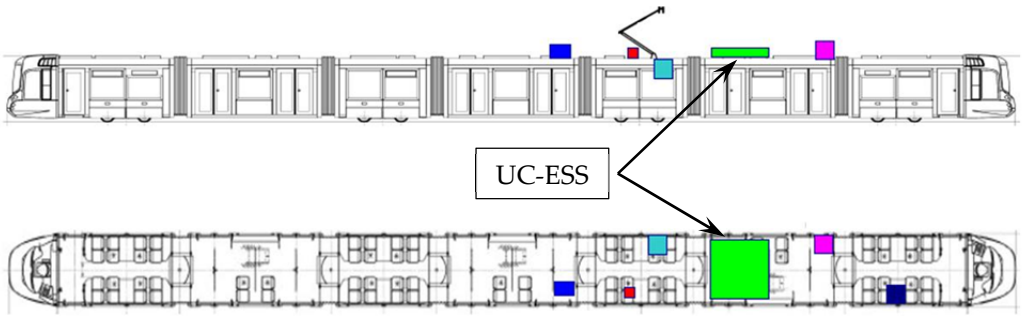


Figure 3.31 : Tramway with on-board ultra-capacitor energy storage system (UC-ESS)

The main goal of the STEEM prototype is to supply the tramway with the necessary energy to safely pass through the zone where the overhead contact line is unavailable. However, UC-ESSs can also be used to reduce the energy consumed by the tramway from the catenary, by recovering as much as possible from the breaking energy and re-using it when the tramway accelerates. The optimal dimensioning regarding the profitability of the integration of an UC-ESS on-board a tramway has been discussed in [CAN 10].

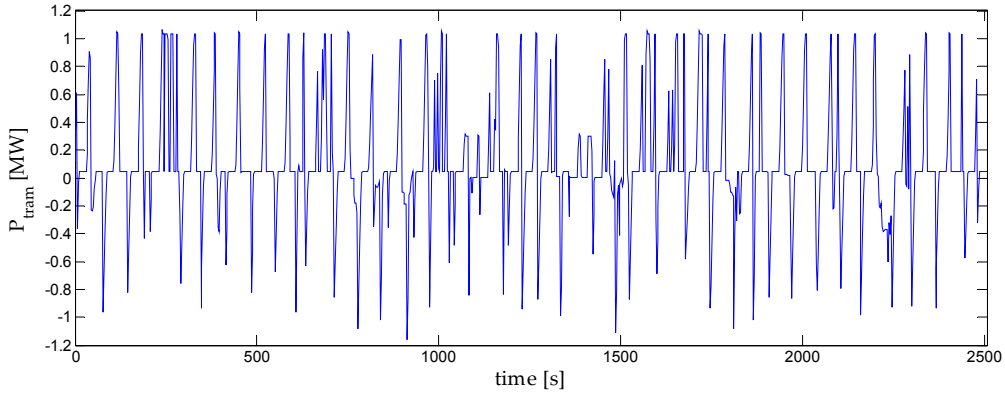


Figure 3.32 : Mission profile for the considered tramway

On the mission profile in Figure 3.32, the tramway's traction power is represented for a round-trip. The positive power peaks correspond to the acceleration phase of the tramway following its stop at a station. The negative power peaks correspond to the tramway's breaking phase when reaching a station. The zero power zones of the mission profile correspond to the tramway stopping at a station. In fact, in this case the power is close to zero but not null, due to the power consumed by the auxiliary equipment on-board the tramway, such as lighting, ventilation, heating, signaling

etc. In order to integrate an UC-ESS to the traction system of a tramway, an adapted control strategy of the pack is imposed. A control strategy of the UC-ESS on-board a tramway, adaptable to different mission profiles, has been proposed by Cantegrel et al. in [CAN 10a].

3.8.2 Tramway traction system description

The simplified energy diagram of a tramway using an on-board ultra-capacitor (UC) energy storage system (UC-ESS) is presented in Figure 3.33. The modeling of the different components of the tramways traction system does not make the object of this work and thus it is not addressed here. Details concerning the modeling of these components can be found in [CAN 11]. Here, the considered components of the tramway's traction system are briefly overviewed, the focus being on the optimization of the complex system represented by the tramway with on-board UC-ESS.

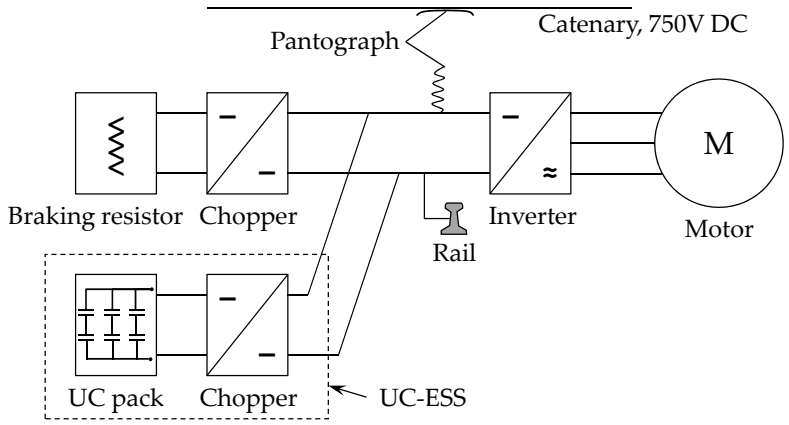


Figure 3.33 : Tramway with on-board ultra-capacitor energy storage system (UC-ESS)

On the diagram in Figure 3.33 several elements can be identified: the catenary, at 750V DC which supplies the energy for the tramways traction; the pantograph, which connects the tramways traction system to the catenary; the traction motors, ensuring the tramways traction; the braking resistor, which has the role of dissipating the braking energy developed by the motors when the tramway brakes and the UC-ESS, whose goal is to ensure the passage of the tramway through the zone where the overhead contact line is unavailable, by supplying the necessary traction power, and at the same time reduce the energy consumption of the tramway from the catenary, by supplying a part of the traction power of the motors and recovering as much as possible from the braking energy. The energy is supplied to the tramway's motors from the catenary at 750V DC, through a pantograph and through an inverter directly to the motor. When the tramway brakes, the energy flux is reversed, i.e. the motors become generators, sending back the braking energy. Due to restrictions concerning the voltage distortion level at the overhead contact line, the catenary does not recover any of the braking energy. Thus, in the absence of an UC-ESS, this energy is completely lost being burned entirely in the braking resistors. When an UC-ESS pack is present, a part of the braking energy is recovered by the UC pack by the intermediate of a chopper. To minimize the tramways energy consumption, it is preferable that the UC-ESS pack recovers as much energy as possible, in order to reduce the energy drawn from the overhead contact line, which underlines the interest of minimizing this energy within the optimal design problem formulation.

The considered UC-ESS consists of an ultra-capacitor (UC) pack and the power electronics module associated to it. It is composed of the following modules: an ultra-capacitor (UC) pack, an electronic module whose chopper coil will be dimensioned and the heat sink for cooling the IGBTs of the electronic module. A simplified energetic schema of the tramway embarking the UC-ESS is presented in Figure 3.34.

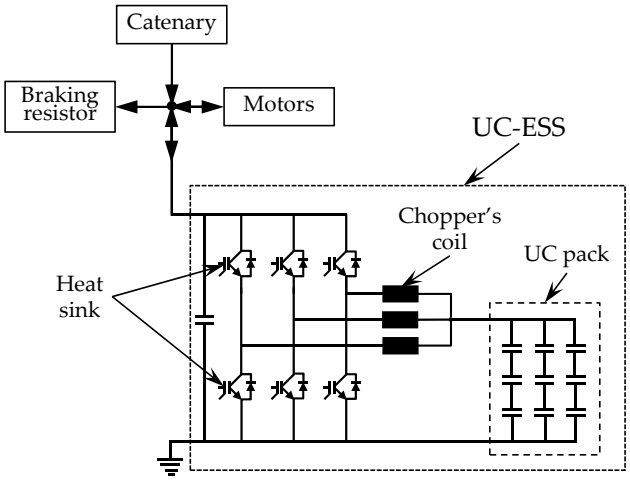


Figure 3.34 : Simplified energetic representation of the tramway embarking the UC-ESS

Ultra-capacitor (UC) pack

The ultra-capacitor (UC) pack of the UC-ESS to be optimally dimensioned is represented in Figure 3.35. The pack is composed of a number of elementary UC cells which are connected in series and in parallel, which number is to be determined by the optimization process.

a) UC pack b) Series-parallel connection of the UC cells

Figure 3.35 : Ultra-capacitor (UC) pack representation

The model of the UC pack, with the design variables and the model outputs are presented schematically in Figure 3.36.

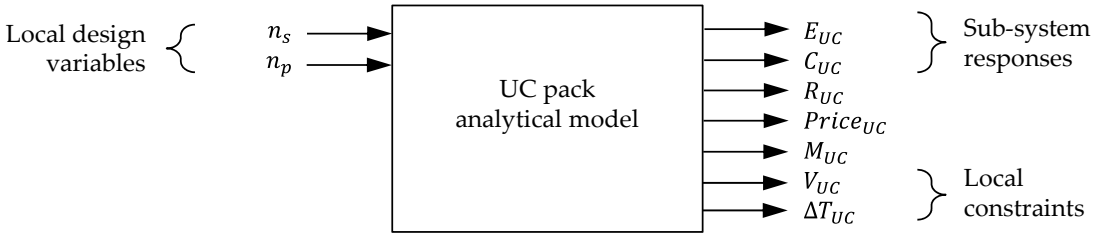
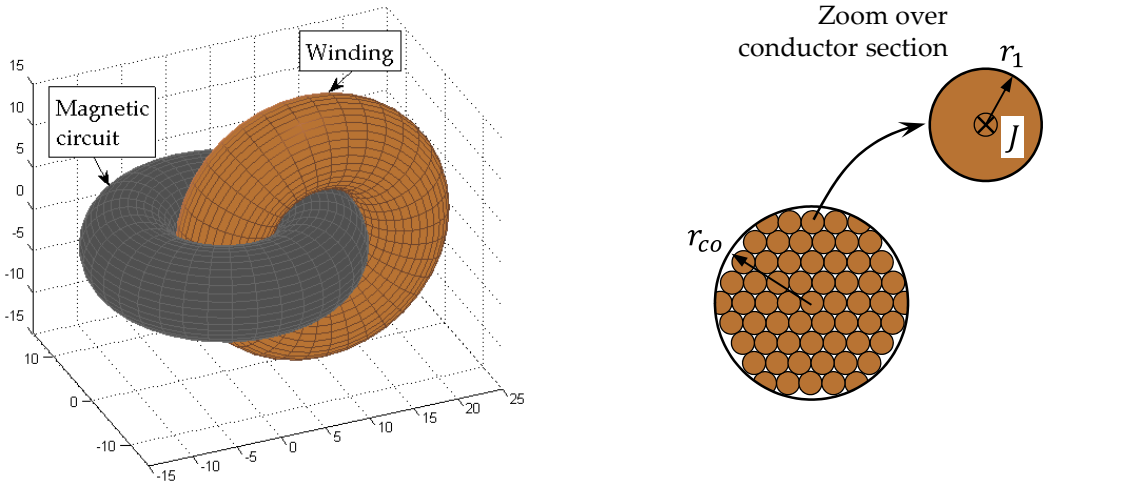


Figure 3.36 : UC pack model description

The UC pack optimization problem has two design variables, n_s, n_p and two coupling variables with the system optimization problem, E_{UC} and C_{UC}.

Chopper's coil

For the representation of the chopper's coil, a bi-toroid model was considered. Figure 3.37 presents the geometry of the bi-toroid model considered for representing the chopper's coil. However, even though this bi-toroid representation of the chopper's coil is different from the actual coil of the chopper, it provides a good estimation of the coil's parameters. A detailed description of the bi-toroid model of the coil can be found in [ROS 09].



a) Bi-toroid coil representation b) Winding slice cut view and conductors

Figure 3.37 : Chopper's coil representation

The design variables and the outputs of the chopper's coil model are presented in Figure 3.38.

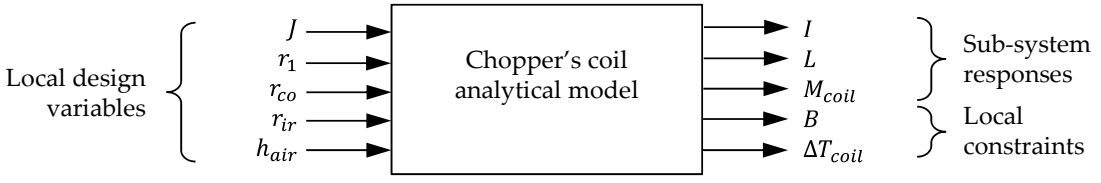


Figure 3.38 : Chopper's coil model description

The chopper's coil optimization problem implies a number of five design variables and three coupling variables connecting to the system optimization problem.

Heat Sink

The heat sink used for cooling the IGBTs of the power electronics module is represented in Figure 3.39. The description of the heat sink model considered here can be found in [KRE 08].

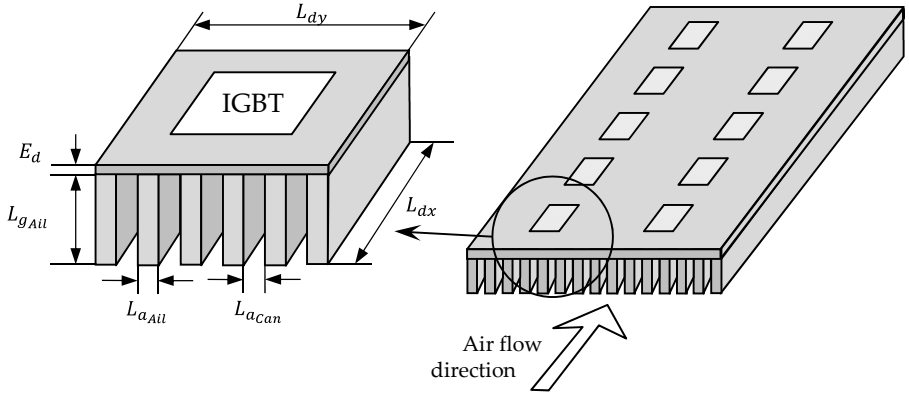


Figure 3.39 : Heat sink representation

The design variables and the outputs of the heat sink are presented in Figure 3.40.

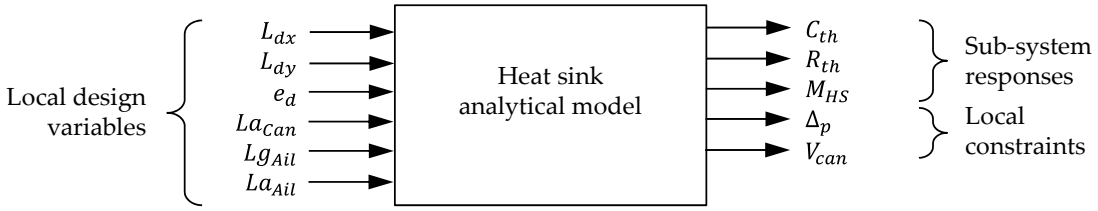


Figure 3.40 : Heat sink model description

A similar application of multi-level optimization, making use of the presented heat sink model has been addressed by Moussouni et al. in [MOU 08] and by Kreuawan et al. in [KRE 09].

Tramway system

The system model of the tramway system integrates the mission profile simulation, making the model the most computationally expensive of all models of the multi-level hierarchy. The system model has been developed by Cantegrel et al. [CAN 10a]. A linear programming problem is fostered by the system model, having as goal to find the traction power called from the catenary by the tramway and the power to be supplied by the UC pack for maximizing its participation over the considered mission profile. The design variables and the outputs of the tramway system model are presented in Figure 3.41.

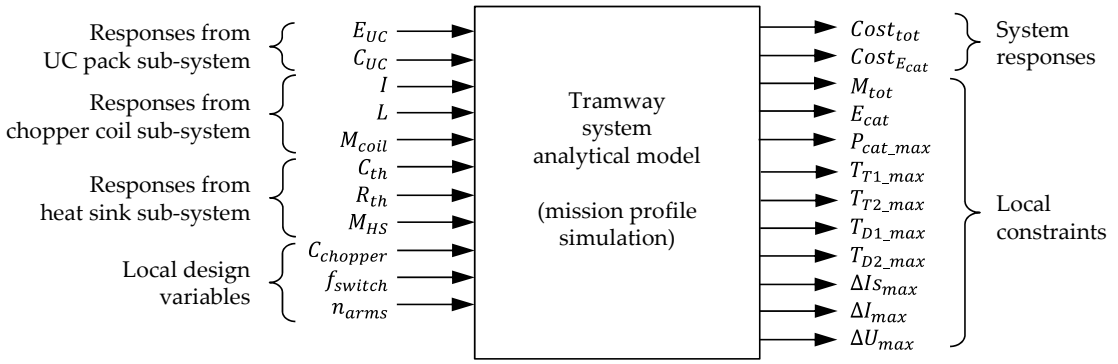


Figure 3.41 : Tramway system model description

The system model implies a number of 11 design variables among which 3 local variables and 8 coupling variables with the lower-level optimization problems. The goals of the optimization problem consist of minimizing the total cost of the UC-ESS system, $Cost_{tot}$ and simultaneously reducing the energy called from the catenary, $Cost_{E_{cat}}$, while respecting the local constraints.

3.8.3 UC-ESS multi-level optimal design problem formulation

The goal of the optimization problem consists in finding the optimal dimensions of the components of the UC-ESS pack, while taking into account the interaction of the pack with the tramway system. The optimal dimensioning of the UC-ESS with regard to its interactions with the tramway system is done for a given mission profile, presented in Figure 3.32.

The optimal dimensioning of an UC-ESS on-board a tramway using the ATC multi-level optimization method can be found in [CAN 11]. Compared to the classical hierarchically-decomposed problems handled using the ATC strategy, that are found in the literature, where the fidelity and complexity of the different elements of the hierarchy increases progressively down the hierarchy [CHO 05], [ALL 06a], this structural decomposition of the UC-ESS presents a complex high-fidelity model at the top of the hierarchy and analytical models of the sub-systems at the lower level. Two similar applications, a vehicle and a structural optimization problem where the computational-demanding model is situated at the top level of the ATC hierarchy, have been presented recently by Guarneri et al. in [GUA 11]. In the UC-ESS optimization case, the complexity of the system element at the top level is mainly given by the presence of a linear programming simulation using the entire mission profile, in order to find the optimal set point power for the UC pack.

Using the un-relaxed ATC multi-level formulation, the optimization problem corresponding to the system element from the top level of the UC-ESS hierarchy can be expressed as in (3.49).

$$\begin{aligned} P_{sys}: \quad & \underset{\bar{\mathbf{x}}_{sys}}{\text{Minimize}} \quad f_{sys}(\bar{\mathbf{x}}_{sys}) = \left\|\mathbf{T}_{sys} - \mathbf{R}_{sys}\right\|_2^2 \\ & \text{with respect to} \quad \bar{\mathbf{x}}_{sys} = [\mathbf{x}_{sys}, \mathbf{t}_{ss}] \\ & \text{where} \quad \mathbf{x}_{sys} = [C_{chopper}, f_{switch}, n_{arms}] \end{aligned} \tag{3.49}$$

$$\mathbf{t}_{ss} = \left[[E_{UC}, C_{UC}], [I, L, M_{coil}], [C_{th}, R_{th}, M_{HS}]\right]$$

$$\mathbf{T}_{sys} = [0,0] \qquad \mathbf{R}_{sys} = \left[Cost_{tot}, Cost_{E_{cat}}\right]$$

$$\text{subject to} \quad \mathbf{c}_{sys} = \bar{\mathbf{t}}_{ss} - \bar{\mathbf{r}}_{ss} = [\bar{\mathbf{t}}_{UC}, \bar{\mathbf{t}}_{coil}, \bar{\mathbf{t}}_{HS}] - [\bar{\mathbf{r}}_{UC}, \bar{\mathbf{r}}_{coil}, \bar{\mathbf{r}}_{HS}] = \mathbf{0}$$

$$\bar{\mathbf{t}}_{UC} = [E_{UC}, C_{UC}] \qquad \bar{\mathbf{r}}_{UC} = [E_{UC}^{L}, C_{UC}^{L}]$$

$$\bar{\mathbf{t}}_{coil} = [I, L, M_{coil}] \qquad \bar{\mathbf{r}}_{coil} = [I^{L}, L^{L}, M_{coil}^{L}]$$

$$\bar{\mathbf{t}}_{HS} = [C_{th}, R_{th}, M_{HS}] \qquad \bar{\mathbf{r}}_{HS} = [C_{th}^{L}, R_{th}^{L}, M_{HS}^{L}]$$

$$\mathbf{g}_{sys}(\bar{\mathbf{x}}_{sys}) \leq \mathbf{0}$$

To minimize the objective function f_{sys}, the global target $\mathbf{T}_{sys}$ is imposed null. The optimization problem is thus formulated using unattainable targets.

The system optimization problem has the goal of minimizing the global targets $\mathbf{T}_{sys}$, thus the total cost of the UC-ESS system $Cost_{tot}$ simultaneously with the cost of the energy taken from the catenary $Cost_{E_{cat}}$, while respecting the local system constraints $\mathbf{g}_{sys}$. The optimization problem both on the local design variables $\mathbf{x}_{sys}$ and the coupling variables with the lower-level elements $\mathbf{t}_{ss}$.

The ATC optimization process starts without counting for the consistency constraints, since the lower level elements were not yet launched to obtain response values for the coupling variables, $\bar{\mathbf{r}}_{ss} = [\bar{\mathbf{r}}_{UC}, \bar{\mathbf{r}}_{coil}, \bar{\mathbf{r}}_{HS}]$. The L superscript of the coupling variables denotes the responses received from the **L**ower level. Once the values for the coupling variables are determined at the system level, these are next "cascaded" as targets to the three children of the system element, P_{UC}, P_{coil} and P_{HS}. The un-relaxed expressions of the optimization problems for the sub-system level elements P_{UC}, P_{coil} and P_{HS} are given in (3.50), (3.51) and (3.52), respectively.

$$P_{UC}: \quad \underset{\bar{\mathbf{x}}_{UC}}{\text{Minimize}} \quad f_{UC}(\bar{\mathbf{x}}_{UC}) = 0$$

$$\text{with respect to} \quad \bar{\mathbf{x}}_{UC} = \left[n_s, n_p\right]$$

$$\text{where} \quad \bar{\mathbf{r}}_{UC} = [E_{UC}, C_{UC}] = a_{UC}(\bar{\mathbf{x}}_{UC}) \tag{3.50}$$

$$\text{subject to} \quad \mathbf{c}_{UC} = \bar{\mathbf{t}}_{UC} - \bar{\mathbf{r}}_{UC} = [E_{UC}^{U}, C_{UC}^{U}] - [E_{UC}, C_{UC}] = \mathbf{0}$$

$$\mathbf{g}_{UC}(\bar{\mathbf{x}}_{UC}) \leq \mathbf{0}$$

The superscript U of the coupling variables E_{UC}^{U} and C_{UC}^{U} denotes the targets for these variables, received from the system element at the **U**pper level.

The optimization problem of the chopper's coil is formulated in (3.51).

$$P_{coil}: \quad \underset{\bar{\mathbf{x}}_{coil}}{\text{Minimize}} \quad f_{coil}(\bar{\mathbf{x}}_{coil}) = 0$$

$$\text{with respect to} \quad \bar{\mathbf{x}}_{coil} = [J, r_1, r_{co}, r_{ir}, h_{air}] \tag{3.51}$$

$$\text{where} \quad \bar{\mathbf{r}}_{coil} = [I, L, M_{coil}] = a_{coil}(\bar{\mathbf{x}}_{coil})$$

$$\text{subject to} \quad \mathbf{c}_{coil} = \bar{\mathbf{t}}_{coil} - \bar{\mathbf{r}}_{coil} = [I^U, L^U, M^U_{coil}] - [I, L, M_{coil}] = \mathbf{0}$$

$$\mathbf{g}_{coil}(\bar{\mathbf{x}}_{coil}) \leq \mathbf{0}$$

The multi-level partitioning of the UC-ESS problem implies only two levels, at top and bottom. The formulation of the heat sink optimization problem is expressed in (3.52).

$$\begin{aligned} P_{HS}: \quad & \underset{\bar{\mathbf{x}}_{HS}}{\text{Minimize}} \quad f_{HS}(\bar{\mathbf{x}}_{HS}) = 0 \\ & \text{with respect to} \quad \bar{\mathbf{x}}_{HS} = \left[L_{dx}, L_{dy}, e_d, La_{Can}, Lg_{ail}, La_{ail}\right] \\ & \text{where} \quad \bar{\mathbf{r}}_{HS} = [C_{th}, R_{th}, M_{HS}] = a_{HS}(\bar{\mathbf{x}}_{HS}) \\ & \text{subject to} \quad \mathbf{c}_{HS} = \bar{\mathbf{t}}_{HS} - \bar{\mathbf{r}}_{HS} = [C^U_{th}, R^U_{th}, M^U_{HS}] - [C_{th}, R_{th}, M_{HS}] = \mathbf{0} \\ & \qquad\qquad \mathbf{g}_{HS}(\bar{\mathbf{x}}_{HS}) \leq \mathbf{0} \end{aligned} \tag{3.52}$$

For all three lower-level optimization problems, the coupling variables with the system optimization problem at the top-level of the multi-level hierarchy are in fact outputs of the local analysis models a_{UC}, a_{coil} and a_{HS}.

3.8.4 ATC formulation with adaptive-metamodel approximation

The evaluation of the upper-level simulation model takes between 1 and 2 minutes, depending on the design configuration. In this condition, the direct integration of the upper-level simulation model within the ATC multi-level optimization process would be prohibitive, due to the important number of model evaluations during the optimization process. Thus, the Kriging metamodel methodology has been employed to create response surfaces of the simulation model outputs. For this, an initial experimental design of 500 samples has been considered using a Latin Hypercube strategy (LHS) for the 11 input variables of the system-level simulation model. For each of the two objective functions and 10 constraint functions, a different Kriging metamodel has been created, based on the initial 500 sample designs.

The ATC optimization process has been launched using the Kriging metamodels as substitutes for the upper-level simulation model. The adaptive metamodeling strategy is employed with the ATC formulation. Hence, at each iteration of the ATC process, at the upper level an optimal result is obtained using the Kriging metamodels. This design is next validated using the true simulation model and the support points list for the Kriging metamodels is updated with the optimal result. The Kriging models are re-built, using the updated list of support points, thus increasing the precision of Kriging metamodels prediction.

The convergence of the ATC optimization process has been attained after 155 ATC iterations, with a pre-imposed consistency value of 10^{-3} in normalized values.

The optimal values of the local design variables for each optimization sub-problem of the hierarchy are presented in Table 3.8.

Table 3.8 : Local design variables of the different optimization problems of the hierarchy

Problem	Variable	Description	Units	Bounds	Optimal results
System	$C_{chopper}$	Chopper's capacity	F	$[1e^{-4}, 1e^{-2}]$	$7.83e^{-3}$
	f_{switch}	Switching frequency	Hz	$[1e^{3}, 1e^{4}]$	$3.59e^{3}$
	n_{arms}	Number of inverter arms	-	[1,20]	14.98
UC pack	n_s	Number of series-connected cells	-	[1,500]	155.33
	n_p	Number of parallel-connected cells	-	[1,100]	5.38
Coil	J	Current density	A/m²	$[5e^{5}, 8e^{6}]$	$1.69e^{6}$
	r_1	Copper wire radius	m	$[1e^{-3}, 1e^{-2}]$	$4.52e^{-3}$
	r_{co}	Winding's radius	m	$[1e^{-3}, 2e^{-1}]$	$2.13e^{-2}$
	r_{ir}	Magnetic core's radius	m	$[1e^{-3}, 2e^{-1}]$	$4.67e^{-2}$
	h_{air}	Air-gap thickness	m	$[1e^{-3}, 1e^{-2}]$	$1.58e^{-3}$
Heat sink	L_{dx}	Width of heat sink	m	$[1e^{-1}, 2]$	0.81
	L_{dy}	Length of heat sink	m	$[1e^{-1}, 2]$	0.39
	e_d	Diffuser thickness	m	$[1e^{-4}, 1e^{-2}]$	$6.15e^{-3}$
	La_{Can}	Heat sink fin gap	m	$[1e^{-4}, 1e^{-2}]$	$1e^{-2}$
	Lg_{ail}	Fin height	m	$[1e^{-3}, 1e^{-1}]$	$1e^{-1}$
	La_{ail}	Fin thickness	m	$[1e^{-4}, 1e^{-2}]$	$1e^{-4}$

The optimal values obtained using the Kriging metamodel for the upper-level model have been validated using the system-level simulation model. The optimal values of the system-level targets are presented in Table 3.9.

Table 3.9 : Target variables optimal values obtained by ATC

Target variable	Description	Units	Metamodel optimum	Simulation validation
$Cost_{tot}$	Total cost of the UC-ESS	[€]	$1.22e^{5}$	$1.27e^{5}$
$Cost_{E_{cat}}$	Cost of the energy called at the catenary	[€]	9.35	9.36

From the results presented in Table 3.9, it can be seen that the value predicted by the Kriging metamodel for the total cost of the UC-ESS is slightly different from the simulation model output, while the prediction of the catenary energy cost is very close to the simulation value.

The constraint limits for the different outputs of the optimization problems in the multi-level hierarchy are expressed in Table 3.10.

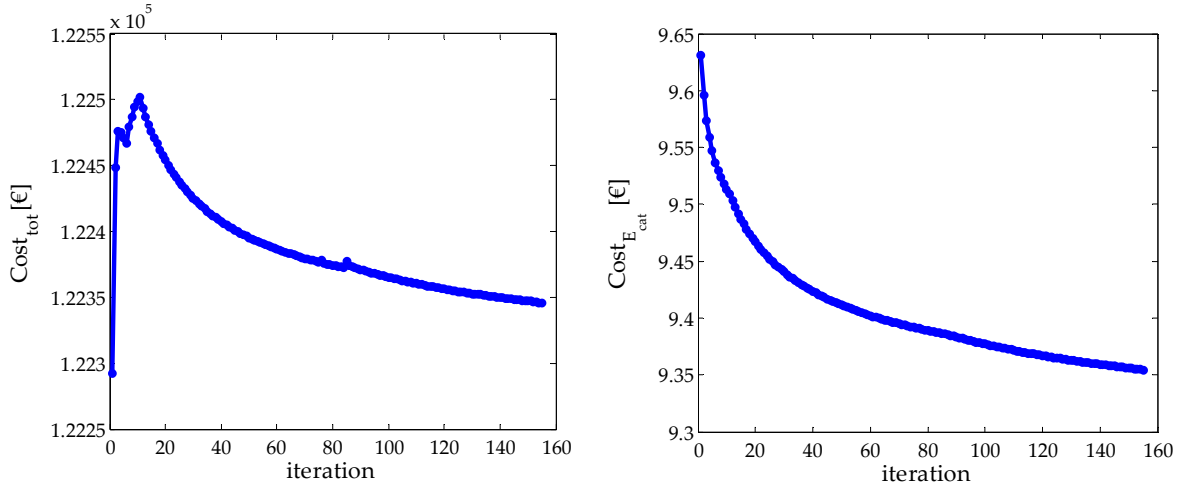
Table 3.10 : Local constraints of the different optimization problems of the hierarchy

Problem	Variable	Description	Units	Limit	Metamodel results	Simula-tion
System	M_{tot}	Total UC-ESS system mass	kg	≤ 600	593.97	589.6
	E_{cat}	Energy received from catenary	J	$\leq 4e^{8}$	$3.06e^{8}$	$3.06e^{8}$
	P_{cat_max}	Max. power called from catenary	W	$\leq 1e^{6}$	$9.46e^{5}$	$9.45e^{5}$

	T_{T1_max}	Max. temperature of transistor T1	°C	≤ 150	103.4	105.7
	T_{T2_max}	Max. temperature of transistor T2	°C	≤ 150	104.8	103.7
	T_{D1_max}	Max. temperature of diode D1	°C	≤ 150	91.4	90.5
	T_{D2_max}	Max. temperature of diode D2	°C	≤ 150	108.5	110.4
	ΔIs_{max}	Current rippling at converter output	A	≤ 20	1.28	1.29
	ΔI_{max}	Converter phase current rippling	A	≤ 200	19.49	19.7
	ΔU_{max}	Voltage rippling at converter entry	V	≤ 2	0.06	0.058
UC pack	V_{UC}	Volume of the UCs in the pack	m^3	≤ 400	-	250.7
	ΔT_{UC}	UC pack overheating	K	≤ 60	-	48
Coil	B	Induction density	T	≤ 2	-	1.92
	ΔT_{coil}	Coil overheating	K	≤ 150	-	150.13
Heat sink	Δ_p	Cooling agent pressure loss	Pa	≤ 1500	-	91.22
	V_{can}	Cooling agent speed	m/s	≤ 50	-	11.25

From the results presented in Table 3.9 and Table 3.10, it can be seen that the Kriging metamodels considered predict well-enough the simulation model outputs at the end of the optimization process. Employing the adaptive metamodeling strategy with the ATC formulation progressively improves the prediction accuracy of the Kriging metamodels along the iterations of the ATC process. This explains the low discrepancy between the Kriging metamodels prediction and the simulation model output values.

It is worth mentioning here that this test-case does not present variables shared between the different lower-level elements. Thus, the lower-level elements only seek to meet as well as they can the targets imposed by the system-level. The evolution of the system-level targets is presented graphically in Figure 3.42a and Figure 3.42b.



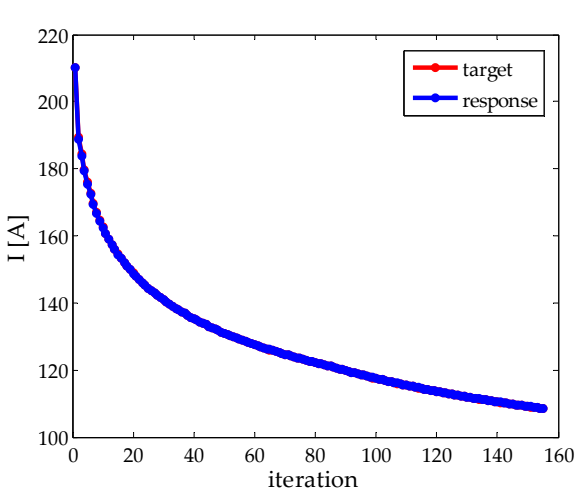
a) Total cost of the UC-ESS b) Catenary energy cost

Figure 3.42 : System-level targets/responses evolution

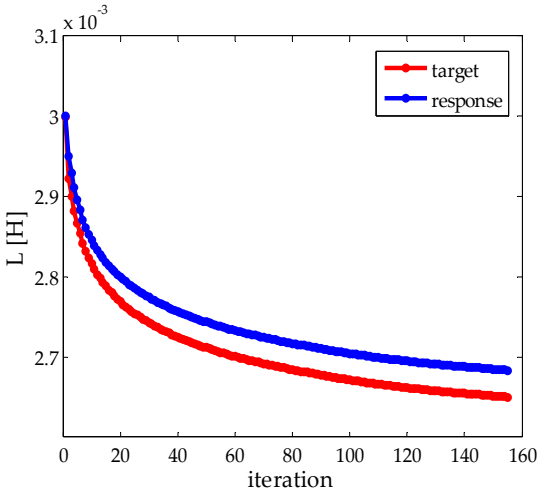
From the evolution of the first system-level target variable $Cost_{tot}$ presented in Figure 3.42a it can be seen that the value of $Cost_{tot}$ increases during the first 15-20 iterations of the ATC process, to start decreasing afterwards, until convergence is attained. Looking at the second system-level target

variable $Cost_{E_{cat}}$, it can be seen a strong reduction of its value at the beginning of the ATC process (first 20 iterations), continuing to decrease until convergence is attained.

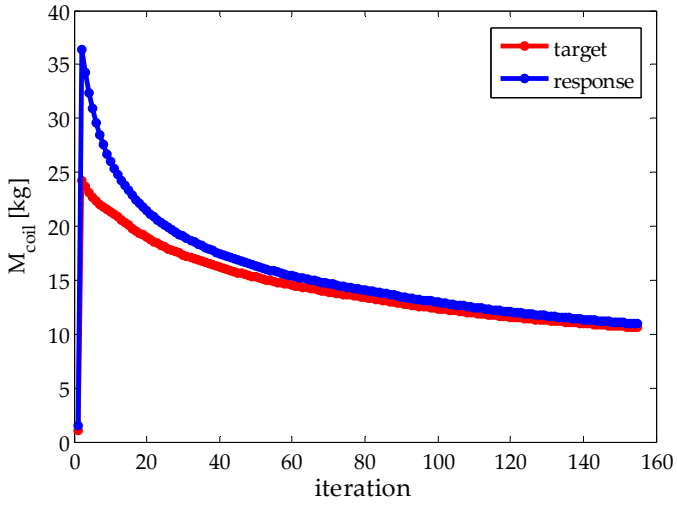
The evolution of the targets/responses for the chopper's coil element of the lower-level of the ATC hierarchy is presented graphically in Figure 3.43.



a) Evolution of I target/response



b) Evolution of L target/response



c) Evolution of M_{coil} target/response

Figure 3.43 : Chopper's coil targets/responses evolution

From the results presented in Figure 3.43, it can be seen that the different coupling variables converged towards their optimal values along the iterations of the ATC process. Similar evolution trends have been obtained for the other two lower-level elements of the ATC hierarchy, the UC pack, respectively the heat sink.

One important remark which should be stated at this point is represented by the important scale-difference between the different coupling variables exchanged between the system-level element and the lower-level elements. While the analytical test-problem addressed by the ATC strategy in paragraph 3.5.7 consisted of local, coupling and shared variables with values at the same scale, all within [0,5], for real engineering problems, such as the case of the present UC-ESS

problem, an important scale difference exists between the values of the different variables of its elements. For engineering problems, such a scale-difference exists naturally due to the different nature of the design variables and model output variables. For example, the three coupling variables of the chopper's coil, I, L, M_{coil} have all three different scales, 200, $3e^{-3}$ and 20, respectively. The influence of this scale difference upon the results of the ATC optimal results can be seen from the results presented in Figure 3.43a, Figure 3.43b and Figure 3.43c. The coupling variable with the greater value, I, had the greatest impact in the targets/responses matching process. Hence, the algorithm matched very well the values of its targets with the values of its responses, as can be seen from the results presented in Figure 3.43a. For the M_{coil} coupling variable, with a smaller value, it can be seen that the ATC process started with a discrepancy between the targets and the responses of this variable, the two matching progressively towards the convergence of the ATC process, as can be remarked from Figure 3.43c. The third coupling variable, with the smaller value among the three, L, had a much less influence within the targets/responses matching process, fact which can be seen from increasing discrepancy between the targets and the responses of this variable along the iterations of the ATC process, in Figure 3.43b.

For this kind of engineering problems, a normalization of all design variables, coupling and shared variables, as well as the system-level targets is strongly required, in order to give equal importance to all variables in the targets/responses matching process. Nevertheless, this task is not at all obvious, finding the most appropriate normalization technique and adequate reference values for all variables including the system-level targets, to comply with the expression of the augmented Lagrangian relaxation technique employed for matching the targets with the responses of the different variables requires further investigation.

3.9 Conclusion

The optimization of complex industrial devices or systems most often implies the breaking-down of the system into several sub-systems or components, which are smaller and more manageable. The optimal design techniques based on the decomposition of complex systems have been presented here and the focus has been set on the multi-level hierarchically-decomposed optimization techniques. The multi-level optimal design techniques imply thus the partitioning of the whole optimization problem into several smaller, but interconnected optimization problems, each associated to an element of the decomposed structure, disposed hierarchically and employing specific techniques for the coordination of the information exchanged between them. The decomposition of a system can be done on several different bases, among which the most popular are model, discipline and object-based decompositions. The multi-level optimization techniques are strictly correlated to the way the system has been partitioned. Different model, discipline and object-based multi-level optimization techniques have been addressed in this chapter.

The multi-level optimization techniques analysis started with the optimal design of systems or devices represented through several models of different fidelity of the same device. An optimization method belonging to this category, the Output Space Mapping (OSM) technique has been employed for the optimal design of a low-voltage single-phase safety isolation transformer, represented through both analytical and finite element modeling. A parallel has been made between the OSM method and a metamodel-based optimization technique, MEGO presented in the

previous chapter, both on a single and multi-objective basis. The computational complexity of the optimization problem using the OSM method is proven strongly reduced in comparison with classical optimization techniques and the MEGO algorithm.

The second subject of interest addressed in this chapter is represented by the optimization of devices or systems benefiting from a multi-disciplinary representation. One of the most representative multi-disciplinary optimization techniques, the Collaborative Optimization (CO) has been analyzed for the optimal design of complex systems decomposed following the different disciplines involved in the representation of the device or system to be designed. The transformer optimization benchmark has been used as application-case for the bi-level CO technique. The multi-level optimization of the different disciplinary analytical models which form the transformer model has been addressed by the presented CO technique. In exchange for the clarity and understanding of the functional relationships between the different elements of the system and the distribution of the computation obtained through the breaking-down of the complex system, provided by the multi-level technique, however the overall computation expense of the optimization process is increased, due to the nested optimization loops required by the CO formulation and the additional variables which link the different disciplines. This fact makes the use of accurate simulation models for representing the different components of the multi-level hierarchy unpractical, prohibitive even. In order to allow the integration of the accuracy provided by the FE models of the transformer into the optimal design process, at a reasonable computational cost, a new implementation of the CO technique, integrating the OSM technique for solving the different disciplinary optimizations, has been proposed in this chapter.

For addressing the optimal design of object-based decomposed systems, the ATC multi-level optimization strategy has been previously identified as an appropriate optimization approach, complying very well with the Alstom Company hierarchical organization. The research work presented in this chapter continues the ideas introduced within previous PhD thesis at the L2EP laboratory at Ecole Centrale de Lille. The analysis of the ATC optimization technique started in this chapter with the basic implementation. In the practical application of the ATC optimization strategy, two figure-cases appear, depending on the way the general system design specification targets are formulated. In the first case, the multi-level optimization problem was formulated as a target-attaining problem, where the goal is to find the design which best meets a given set of requirements, also known as "attainable targets". Previous research works at the L2EP laboratory addressed applications using this optimization formulation. Nevertheless, this formulation represents only a particular case of the general case where no attainable targets are "*a priori*" known and imposed to the ATC optimal design process. This second case is known under the name of "unattainable targets" optimal design and it is formulated in the spirit of the general expression of an optimization problem. The main difficulty of the ATC optimization process consists in the coordination of the information exchanges between the different elements of the hierarchically decomposed structure to be optimally designed. Several strategies for the coordination of the multi-level optimization process are presented along with three different consistency constraint relaxation techniques and their implementation has been discussed over both analytical and physical test-cases.

A multi-level optimization platform, based on the ATC formalism, has been implemented under Matlab®, which allows the resolution of object-based decomposed complex systems optimal design problems. The generic character of the developed platform allows the implementation with ease of

any analytical test-problem or physical application of optimal design, after a prior decomposition following the ATC-specific structure. The distributed implementation of the platform allows for integration within the optimal design process of multiple model elements distributed across an enterprise network, based on an input/output file-exchange mechanism. This characteristic is in perfect correlation with the hierarchically-distributed organization of the Alstom Company. The flexibility of the platform permits the designer to make use of the ATC formalism with a selection of different coordination strategies and consistency constraint relaxation techniques which have been presented in this chapter and implemented. The multi-level optimization platform is intended for an automatic optimization process. Nevertheless, an interactive utilization of the platform is possible, allowing the designer or different domain specialists to intervene within the optimization process. The validation of the multi-level optimization platform has been done using different analytical test-cases and through application to the resolution of an industrial test-case of the Alstom Company, concerning the optimal design of an ultra-capacity energy storage system (UC-ESS) on-board a tramway.

Conclusion

The research work presented in this manuscript addresses the methodologies specific to the optimal design of complex systems in general, and the electromagnetic domain with application in railway systems engineering in particular. The increased dimension of the problems to be handled, as well as the use of domain specific simulation tools for analyzing the behavior of the diverse devices or system's components, requires the use of optimization methodologies which are well adapted to this purpose.

The presentation of the research work in this thesis starts with the introduction of the complex systems optimal design concept in Chapter 1 of the manuscript. The optimal design process is a regarded as a three step process, where all steps are equally important and require being appropriately addressed. A preliminary phase of the optimal design process consists in the analysis of the models to be employed by the optimization and the appropriate formulation of the problem to be solved by the optimization process. The optimization process itself represents the core of the optimal design process and the optimization techniques of complex electromagnetic systems makes the subject of this research, being addressed in chapters 2 and 3. The third phase of the optimal design process consists in the results analysis and interpretation, finalized by the decision making concerning the design to be produced in practice.

Among the numerous statistical analysis methods existing in the literature, the correlation coefficients computation which allows gaining insights into the relationships governing the models is being presented in this first chapter. In electromagnetics, and in engineering in general, the majority of optimization problems are nonlinear, presenting many constraints which ensure the feasibility of designs and several, often conflicting, objectives which require to be optimized. The multi-objective character of the problem being possible to be handled either prior to the optimization process, through single-objective optimization techniques after applying some transformation technique, or "*a posteriori*", in this case a multi-objective optimization technique being employed to determine the Pareto front of the problem representing the optimal trade-off between the different objectives expressed. Several transformation techniques for handling a multi-objective problem using single-objective methods have been analyzed in this chapter with regard to their appropriation for determining the optimal solutions of problems presenting both convex and non-convex Pareto fronts. These basic techniques find their utility within the optimization techniques proposed later on in the manuscript. A few techniques and tools for multi-data representation, having the goal to assist the designer with the decision making are equally introduced towards the end of this chapter.

Chapter 2 of the manuscript addresses the optimization methodology based on metamodels of a device or system, which is very well suited for handling complex engineering design problems. The

problematic related to the generation of different types of metamodels (polynomial response surfaces, radial basis functions, Kriging models, neural networks, etc.) has been widely addressed in the literature and recent research studies cover this subject, therefore it has not been addressed here. In this chapter, the focus is set on analyzing the integration of metamodels within the optimization process.

The presentation of the optimization techniques employing metamodels starts with the most basic approach, in which the complex simulation model employed to represent the device or system to be designed is simply replaced with a metamodel during the optimization process. However, the fast evaluation of metamodels, incomparable to the time required by accurate simulation models, does not make up for the lack of precision of these metamodels. This fact can be remarked by application to a simple analytical problem. This brings to a second approach, an adaptive metamodel-based optimization technique, where the metamodels of the simulation model are simultaneously used to predict the outputs of the real model and are being progressively improved in order to increase their global accuracy. Several infill design selection techniques are proposed for addressing multi-objective problems using this technique. An electromagnetic optimal design problem of a linear induction motor (LIM) represented through 3D FE modeling is used as application for the adaptive metamodel-based design optimization technique developed. The 3-objective Pareto front of the problem has been determined by this technique using a reduced number of calls to the expensive 3D FE model of the LIM.

The third approach, which also represents the most complex technique and the core of the second chapter, proposes a different approach, where an infill selection criterion combining both the search for optimal designs and the improvement of the metamodels in the relevant areas of the design space, where the optimum is sought, is used to guide the algorithm towards the global optimal design. The Efficient Global Optimization (EGO) algorithm proposed by Jones in [JON 98] employs Kriging models as metamodels and uses a particularity of this type of models which offer, along with the prediction of the real model's outputs, also an estimation of the prediction error. These two quantities are used within the expression of a statistical infill criterion. Several single-objective infill criteria existing in the literature have been reviewed and the multi-objective infill criterion first proposed by Kreuawan et al. in [KRE 08] has been implemented under the multi-objective EGO algorithm (MEGO) developed. The infill mechanism of the original EGO algorithm is sequential, hence not allowing the use of distributed computation. To overcome this drawback, two strategies have been proposed and implemented within the MEGO algorithm, which allow for the generation of not just one design, but a set of infill designs to be evaluated by the accurate simulation model at each iteration of the optimization algorithm. This allows the distribution of the computational burden of the simulation model over several cores of the same computer, or several computers over a network, reducing thus the overall time required by the optimization process. The validation of the MEGO algorithm on a classical electromagnetic optimization benchmark problem TEAM22, consisting in the optimal dimensioning of a superconducting magnetic energy storage device (SMES), was done both on a single-core and on a multi-core server, benefitting thus of the computation distribution. The development of the MEGO algorithm has been finalized with the development of a graphical user interface in Matlab® for assisting the designer with the definition of the optimization process and the MEGO parameter setting, which has been packaged under the form of a toolbox.

Chapter 3 of the manuscript is dedicated to the optimization techniques of complex systems through means of decomposition of the problem following different criteria. Different decompositions of the complex system are possible, following the level of fidelity of the models employed for representing the system, the different disciplines involved in the simulation of its behavior, or following the different physical sub-systems and components which form the system. The discussion started with the Output Space Mapping (OSM) technique, a version of the Space Mapping method which deals with two or more models of different fidelity of the same device or system. A single-phase low-voltage safety isolation transformer has been used as benchmark for assessing the performance of this technique. A parallel has been made between OSM and the MEGO technique developed in Chapter 2 of the manuscript, which was made possible by the sum of features common between the two techniques. For this particular benchmark, the OSM technique has been shown to outrun the MEGO algorithm both in a single- and a multi-objective optimization context. The main reason of this outstanding performance is related to the quality of the analytical model of the transformer, the optimum/optima being determined by OSM with a much reduced number of high fidelity model evaluations.

Another decomposition-based optimization technique addresses systems decomposable following the different disciplines involved in their representation. The Collaborative Optimization (CO) strategy is the most representative technique from this category. This strategy implies the bi-level decomposition of the system, where the disciplinary models of the system or device are displaced on the inferior level of the hierarchy, while a system-level coordination strategy situated at the top level is in charge with ensuring the interdisciplinary consistency of the design. A different optimization problem is formulated for each discipline and the interactions between disciplines are managed by a system-level optimizer in charge with optimizing the objective function of the problem and seeking for attaining the interdisciplinary consistency. Hence, during the CO process, inconsistencies between the different disciplinary models involved are allowed; the design consistency is guaranteed only at the convergence of the CO process. The previously mentioned transformer benchmark problem with the analytical model representation has been used as application case for the CO strategy. The optimization process was proved to converge in a reduced number of CO iterations. However, the number of evaluations of each disciplinary analytical model of the transformer is large enough to impede the replacement of the analytical disciplinary models with the corresponding 3D FE disciplinary models of the transformer. In order to benefit from the increased accuracy offered by the FE modeling, the integration of the previously presented OSM technique, employed this time for each discipline at a time rather than to the coupled model of the transformer, has been proposed. The proof of the adequacy of the presented hybrid CO-OSM technique for solving the transformer benchmark problem, as well as other electromagnetic problems, represents the subject of future studies.

The physical decomposition of a system into its sub-systems and components is the subject of the third multi-level optimization approach addressed in this chapter. The ATC method is the most representative optimization technique which handles systems subjected to an object-based decomposition. According to ATC, the system is decomposed into its constituent components, which are hierarchically displaced onto two or more levels, starting with the general representation of the system at the top of the hierarchy, continuing down the hierarchy with its sub-systems and having its basic components at the bottom level of the hierarchy. For each element of the decomposed structure, a different optimization problem is formulated and a general coordination

strategy is employed for managing the exchanges between the different optimization processes, driving the design towards convergence. This optimization technique was found to comply very well with the organization of the Alstom Company in previous research studies of Kreuawan [KRE 08] and Moussouni [MOU 09a] at the L2EP laboratory of Ecole Centrale de Lille. Their approach was however limited to addressing only a particular case of the multi-level optimization problems, when attainable values are imposed for the general specification targets are known and imposed to the ATC optimization algorithm. This case is however marginal in practice, such attainable targets being unavailable prior to the optimization process. The study of the ATC technique was continued in the present work in order to extend the application of ATC to the general case of optimization using unattainable targets, thus without having to know and specify attainable values for the upper level targets. A special coordination process and consistency constraint relaxation technique, allowing the resolution of ATC problems with unattainable targets, has been identified in the literature and implemented within a distributed multi-level optimization platform in Matlab®. The multi-level ATC formulation has been employed for solving an optimization test-case of the Alstom Company consisting in the optimal dimensioning of an ultra-capacitor energy storage system onboard a tramway (UC-ESS) with the goal of reducing the energy called from the overhead power line and reducing the cost of the UC-ESS. To reduce the computational burden induced by the expensive simulation model at the system-level of the ATC hierarchy, the implementation of an adaptive metamodel-based optimization strategy similar to the one presented in Chapter 2 of the manuscript has been proposed. This way, the accuracy of the simulation model is combined with the fast evaluation of metamodels, allowing for the completion of the ATC optimization process of the UC-ESS within a reasonable time.

Within the Optimization team of the L2EP laboratory at Ecole Centrale de Lille, we are committed to seek methods and methodologies to pose and solve complex problems of optimal design. The work presented in this manuscript is meant to bring some new perspective look over the optimal design process of complex systems. The optimization methodologies based on the use of metamodels of the devices to be conceived show a great potential for real engineering optimal design processes. Several enhancements might be addressed in future related works. An interesting improvement line might consist in the integration of discrete variables into the metamodel-based optimization process. The optimization approaches based on the decomposition of a system into its components offer the important possibility of distributing the computational complexity over the system. A bi-level railway application has been addressed using the Analytical Target Cascading object-based decomposition method. The integration of multi-disciplinary methods within the multi-level formulations might be an interesting future development line, meaning to address more complex system optimal design problems.

The optimization tool is shown to be incontrovertible, when the capacity of understanding of engineers is overpassed by the ever-growing complexity of the optimal design task. The traditional design approach of engineers consists in a step-by-step problem solving, adopting diverse simplifying hypothesis along the way, mainly based on experience and intuition. More effort needs to be directed to the translating the Alstom Company's optimal design tasks into the appropriate optimization tasks, higher in the design cycle, so that a larger number of degrees of freedom would be available and therefore included in the optimization process.

Appendix A

Kriging metamodeling

The Kriging model has been first introduced in the field of numerical simulations by Sacks et al. [SAC 89] under the name of Design and Analysis of Computer Experiments (DACE). The description of the Kriging model along with practical implementation issues can be found in [JON 98], [JON 01], [KRE 08]. A widely used Matlab® implementation of the Kriging model is represented by the DACE toolbox developed by the Technical University of Denmark [LOP 02].

Although numerical simulations are deterministic, within the Kriging approach the functions are considered as the realization of a stochastic process. Using the Kriging modeling, an unknown true function can be expressed as in (A.1).

$$y(\mathbf{x}) = B(\mathbf{x}) + Z(\mathbf{x}) \tag{A.1}$$

The first term of the expression (A.1), $B(\mathbf{x})$ represents a regression or a polynomial model, which gives the global trend of the function. The second term $Z(\mathbf{x})$ is a model of a random process with normal distribution, zero mean and variance of σ^2. This term gives the local deviations of the function from the global trend. The expression of the covariance takes the form presented in (A.2).

$$Cov\left[Z\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right)\right] = \sigma^2 \mathbf{R}\left[R\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right)\right] \tag{A.2}$$

where R represents the correlation function, $\mathbf{R}$ is the correlation matrix, the superscripts (i) and (j) indicate the sample point from the list of n_s evaluated points which serve to construct the Kriging model. The correlation function R controls the way the model fits the data. Several correlation functions were proposed in [SAC 89], among which the Gaussian correlation function is the most commonly used. The Gaussian correlation is used here and its expression is given in (A.3).

$$R\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right) = \exp\left[-\sum_{k=1}^{n_v} \theta_k \left|x_k^{(i)} - x_k^{(j)}\right|^{p_k}\right] \tag{A.3}$$

where n_v is the number of design variables, θ_k represents the unknown correlation function parameter ($\theta_k \geq 0$), which determines how fast the correlation among points drops off with the change in $\mathbf{x}$ along the k-th dimension (larger values imply a fast decrease in correlation), and p_k represents the smoothness parameter ($p_k \in (0,2]$). Higher values of p_k gives smoother functions,

while values close to 0 give rougher functions. A smooth representation of the function ($p_k = 2$) is often considered and it is also adopted here. The θ_k parameter remains to be determined.

From the expression (A.3) it can be observed that for already sampled points, the correlation is $R = 1$, because of $\left|x_k^{(i)} - x_k^{(j)}\right| = 0$. This suggests a full correlation for the already sampled points, meaning that the Kriging model is an interpolation model, "passing through the points". However, a regression model can also be considered, especially when the level of noise in the modeled function is high (e.g. FEA), but this aspect is not considered here.

The Kriging model produces an estimate of the true response value y, which is the Kriging predictor $\hat{y}$. The expected difference between the estimated value $\hat{y}$ and the true value y of the function is represented by the Mean Squared Error (MSE). The expression of MSE is given in (A.4).

$$MSE(\mathbf{x}) = E\left(\left(y(x) - \hat{y}(x)\right)^2\right) \tag{A.4}$$

For the sampled points, the value of MSE is null, since the Kriging model interpolates the data. For other points, the MSE should be minimized in order to accurately predict the data.

The Kriging predictor takes the form expressed in (A.5).

$$\hat{y} = \mathbf{1}\hat{\beta} + \mathbf{r}^T\mathbf{R}^{-1}\left(y - \mathbf{1}\hat{\beta}\right) \tag{A.5}$$

and the MSE can be stated using the expression (A.6).

$$MSE = \sigma^2\left[1 - \mathbf{r}^T\mathbf{R}^{-1}\mathbf{r} + \frac{(1 - \mathbf{1}^T\mathbf{R}^{-1}\mathbf{r})^2}{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}}\right] \tag{A.6}$$

where $\hat{\beta}$ represents the estimator of the regression model, $\mathbf{r}$ is the correlation vector between a new point $\mathbf{x}$ and the already sampled points, and $\mathbf{1}$ is a unit vector with the length of n_s. The expression of $\mathbf{r}$ is given in (A.7).

$$\mathbf{r}(\mathbf{x}) = \left[R\left(\mathbf{x}, \mathbf{x}^{(1)}\right), R\left(\mathbf{x}, \mathbf{x}^{(2)}\right), \cdots, R\left(\mathbf{x}, \mathbf{x}^{(n_s)}\right)\right] \tag{A.7}$$

The form of the correlation matrix $\mathbf{R}$ is given in (A.8).

$$\mathbf{R} = \left[\begin{pmatrix} R\left(\mathbf{x}^{(1)}, \mathbf{x}^{(1)}\right) & R\left(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}\right) & \cdots & R\left(\mathbf{x}^{(1)}, \mathbf{x}^{(n_s)}\right) \\ R\left(\mathbf{x}^{(2)}, \mathbf{x}^{(1)}\right) & R\left(\mathbf{x}^{(2)}, \mathbf{x}^{(2)}\right) & \cdots & R\left(\mathbf{x}^{(2)}, \mathbf{x}^{(n_s)}\right) \\ \vdots & \vdots & \ddots & \vdots \\ R\left(\mathbf{x}^{(n_s)}, \mathbf{x}^{(1)}\right) & R\left(\mathbf{x}^{(n_s)}, \mathbf{x}^{(2)}\right) & \cdots & R\left(\mathbf{x}^{(n_s)}, \mathbf{x}^{(n_s)}\right) \end{pmatrix}\right] \tag{A.8}$$

The correlation matrix $\mathbf{R}$ is symmetric ($R\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right) = R\left(\mathbf{x}^{(j)}, \mathbf{x}^{(i)}\right)$), with ones on the main diagonal, since $R\left(\mathbf{x}^{(i)}, \mathbf{x}^{(i)}\right) = 1$.

The values of $\mathbf{R}$ and $\mathbf{r}$ depend on the θ_k parameter, which can be found using the Maximum Likelihood Estimation (MLE). The likelihood function (L) is defined as in (A.9).

$$L = \frac{1}{\sqrt{(2\pi\hat{\sigma}^2)^{n_s}|\mathbf{R}|}}\exp\left(\frac{-\left(y - \mathbf{1}\hat{\beta}\right)^T\mathbf{R}^{-1}\left(y - \mathbf{1}\hat{\beta}\right)}{2\hat{\sigma}^2}\right) \tag{A.9}$$

To simplify the expression of L in (A.9), the log-likelihood is used instead ($logL$). The expression of the log-likelihood is given in (A.10).

$$logL = \frac{-\left(y - \mathbf{1}\hat{\beta}\right)^T \mathbf{R}^{-1}\left(y - \mathbf{1}\hat{\beta}\right)}{2\hat{\sigma}^2} - \frac{n_s \ln(2\pi\hat{\sigma}^2) + \ln|\mathbf{R}|}{2} \tag{A.10}$$

where $\hat{\beta}$ and $\hat{\sigma}^2$ can be estimated using the expression (A.11), respectively (A.12).

$$\hat{\beta} = (\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1})^{-1}\mathbf{1}^T\mathbf{R}^{-1}y \tag{A.11}$$

$$\hat{\sigma}^2 = \frac{1}{n_s}\left(\left(y - \mathbf{1}\hat{\beta}\right)^T\right)\mathbf{R}^{-1}\left(y - \mathbf{1}\hat{\beta}\right) \tag{A.12}$$

Substituting the expressions of $\hat{\beta}$ and $\hat{\sigma}^2$ from (A.10) with (A.11) and (A.12), the log-likelihood function takes the form expressed in (A.13).

$$logL = -\frac{n_s \ln(\hat{\sigma}^2) + \ln|\mathbf{R}|}{2} \tag{A.13}$$

The θ_k parameter is found by solving the rephrased MLE optimization problem (A.14). Here, a genetic algorithm is used to solve the MLE problem. Once the optimal value of θ_k is determined, it is then used in the expression (A.5) to predict the values of any other design point **x**.

$$\max_{\theta_k} \quad logL = -\frac{n_s \ln(\hat{\sigma}^2) + \ln|\mathbf{R}|}{2} \qquad \text{subject to} \quad 0 < \theta_k < \infty \tag{A.14}$$

The Kriging model requires an important amount of computation, due to the MLE optimization problem. The computation time of the MLE problem increases with the number of design variables and the size of the sampling plan. In order to obtain a reasonable computation time for the construction of the Kriging model, in [FOR 08] is suggested to be used for a maximum number of 20 design variables, and with a sampling plan inferior to 500 designs.

Bibliography

[AGT 10] J. Agte et al., "MDO: assessment and direction for advancement - an opinion of one international group," *Structural and Multidisciplinary Optimization*, vol. 40, no. 1-6, pp. 17-33, 2010.

[ALB 11] I. Alberto and P. Mateo, "A crossover operator that uses Pareto optimality in its definition," *TOP*, vol. 19, no. 1, pp. 67-92, 2011.

[ALE 00] N.M. Alexandrov and R.M. Lewis, "Analytical and Computational Aspects of Collaborative Optimization," Langley Research Center, Hampton, Virginia, Tech. rep., April 2000.

[ALE 11] M.J. Alexander, J.T. Allison, and P.Y. Papalambros, "Reduced representations of vector-valued coupling variables in decomposition-based design optimization," *Structural and Multidisciplinary Optimization*, vol. 44, no. 3, pp. 379-391, 2011.

[ALE 11a] M.J. Alexander, J.T. Allison, P.Y. Papalambros, and D.J. Gorsich, "Constraint management of reduced representation variables in decomposition-based design optimization," *Journal of Mechanical Design*, vol. 133, no. DETC2010-28788, p. 101014, October 2011.

[ALL 05] J. Allison, M. Kokkolaras, M. Zawislak, and P.Y. Papalambros, "On the use of analytical target cascading and collaborative optimization for complex system design," in *Proceedings of the 6th world congress on structural and multidisciplinary optimization, Rio de Janeiro, Brazil*, 2005.

[ALL 06] J. Allison, B. Roth, M. Kokkolaras, I. Kroo, and P.Y. Papalambros, "Aircraft family design using decomposition-based methods," in *Proceedings of the 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Portsmouth, Virginia, SUA*, September 2006.

[ALL 06a] J. Allison, D. Walsh, M. Kokkolaras, P.Y. Papalambros, and M. Cartmell, "Analytical Target Cascading in aircraft design," in *44th AIAA Aerospace Sciences Meeting and Exhibit, AIAA-2006-1325, Reno, Nevada, SUA*, 2006.

[ALL 08] J.T. Allison, "Optimal Partitioning and Coordination Decisions in Decomposition-based Design Optimization," University of Michigan, USA, Ph.D. dissertation, 2008.

[ALL 09] J.T. Allison, M. Kokkolaras, and P.Y. Papalambros, "Optimal Partitioning and Coordination Decisions in Decomposition-Based Design Optimization," *Journal of Mechanical Design*, vol. 131, no. 8, p. 081008, 2009.

[ALL 10] J.T. Allison and P.Y. Papalambros, "Consistency constraint allocation in Augmented

Lagrangian Coordination," *Journal of Mechanical Design*, vol. 132, p. 071007, 2010.

[ALO 96] P. Alotto et al., "Multiobjective optimization in magnetostatics: a proposal for benchmark problems," *IEEE Transactions on Magnetics*, vol. 32, no. 3, pp. 1238-1241, May 1996.

[ALO 08] P. Alotto et al., "SMES Optimization Benchmark Extended: Introducing Pareto Optimal Solutions Into TEAM22," *IEEE Transactions on Magnetics*, vol. 44, no. 6, pp. 1066-1069, June 2008.

[AUD 00] C. Audet, J.E. Jr. Dennis, D.W. Moore, A. Booker, and P.D. Frank, "A surrogate-model-based method for constrained optimization," in *Proceedings of the 8th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 2000.

[BAL 94] R.J. Balling and J. Sobieszczanski-Sobieski, "Optimization of Coupled Systems: A Critical Overview of Approaches," Institute for Computer Applications in Science and Engineering, DTIC Document, Tech. rep., 1994.

[BAL 11] M. Balesdent, N. Bérend, P. Dépincé, and A. Chriette, "A survey of multidisciplinary design optimization methods in launch vehicle design," *Structural and Multidisciplinary Optimization*, vol. 1, pp. 1-24, 2011.

[BAL 11a] M. Balesdent, "Optimisation multidisciplinaire de lanceurs," Ecole Centrale de Nantes, Nantes, France, in English, Mechanical Engineering, November 2011.

[BAN 94] J.W. Bandler, R.M. Biernacki, S.H. Chen, P.A. Grobelny, and R.H. Hemmers, "Space mapping technique for electromagnetic optimization," *IEEE Transactions on Microwave Theory and Techniques*, vol. 42, no. 12, pp. 2536-2544, December 1994.

[BAN 95] J.W. Bandler, R.M. Biernacki, S.H. Chen, R.H. Hemmers, and K. Madsen, "Electromagnetic Optimization Exploiting Aggressive Space Mapping," *IEEE Transactions on Microwave Theory and Techniques*, vol. 43, no. 12, pp. 2874-2882, December 1995.

[BAN 04] J.W. Bandler et al., "Space Mapping: The State of the Art," *IEEE Transactions on Microwave Theory and Techniques*, vol. 52, no. 1, pp. 337-361, 2004.

[BEN 11] R. Ben-Ayed, A. C. Berbecea, S. Brisset, F. Gillon, and P. Brochet, "Comparison between Efficient Global Optimization and Output Space Mapping technique," *International Journal of Applied Electromagnetics and Mechanics, IJAEM*, vol. 37, no. 2-3, pp. 109-120, 2011.

[BEN 11a] R. Ben-Ayed and S. Brisset, "n-Level Output Space Mapping Technique," in *15th International Symposium on Electromagnetic Fields in Mechatronics, Electrical and Electronic Engineering (ISEF 2011), Funchal, Madeira, Portugal*, 2011.

[BEN 12] R. Ben-Ayed and S. Brisset, "Multidisciplinary optimization formulations benefits on space mapping techniques," *International Journal for Computation and Mathematics in Electrical and Electronic Engineering, COMPEL*, vol. 31, no. 3, pp. 945-957, 2012.

[BEN 12a] R. Ben-Ayed, J. Gong, S. Brisset, F. Gillon, and P. Brochet, "Three-Level Output Space Mapping Strategy for Electromagnetic Design Optimization," *IEEE Transactions on Magnetics*, vol. 48, no. 2, pp. 671-674, February 2012.

[BEN 12b] R. Ben-Ayed, "Eco-conception d'une chaîne de traction ferroviaire," Université des Sciences et Technologie de Lille - Lille 1, Lille, France, Ph.D. dissertation, June 2012.

[BER 99] D.P. Bertsekas, *Nonlinear programming*.: Athena Scientific, 1999.

[BER 03] D.P. Bertsekas, "6.252J Nonlinear Programming, Spring 2003," 2003. [Online]. http://ocw.mit.edu

[BER 10] A. C. Berbecea, S. Kreuawan, F. Gillon, and P. Brochet, "A Parallel Multiobjective Efficient Global Optimization: The Finite Element Method in Optimal Design and Model Development," *IEEE Transactions on Magnetics*, vol. 46, no. 8, pp. 2868-2871, August 2010.

[BER 12] A. C. Berbecea, R. Ben-Ayed, F. Gillon, S. Brisset, and P. Brochet, "Comparison of Efficient Global Optimization and Output Space Mapping on the Biobjective Optimization of a Safety Isolating Transformer," *IEEE Transactions on Magnetics*, vol. 48, no. 2, pp. 791-794, February 2012.

[BER 12a] A. C. Berbecea, F. Gillon, and P. Brochet, "Multi-level design of an isolation transformer using Collaborative Optimization," in *Proceedings of XII-th International Workshop on Optimization and Inverse Problems in Electromagnetism, OIPE 2012, Ghent, Belgium*, 2012.

[BIN 97] T.T. Binh and U. Korn, "MOBES: A multiobjective evolution strategy for constrained optimization problems," in *The Third International Conference on Genetic Algorithms (Mendel 97)*, pp. 176-182, 1997.

[BLO 05] V.Y. Blouin, J.B. Lassiter, M.M. Wiecek, and G.M. Fadel, "Augmented Lagrangian Coordination for decomposed design problems," in *Proceedings of the 6th world congress on structural and multidisciplinary optimization, Rio de Janeiro, Brazil*, vol. 30, 2005.

[BRA 96] R.D. Braun and B. Kroo, "Development and Application of the Collaborative Optimization Architecture in a Multidisciplinary Design Environment," in *Multidisciplinary Design Optimization: State of the Art*, pp. 98-116, 1996.

[BRA 96a] R.D. Braun, P. Gage, I. Kroo, and I. Sobieski, "Implementation and performance issues in collaborative optimization," Langley Research Center, Tech. rep., 1996.

[BRI 07] S. Brisset, Démarches et Outils pour la Conception Optimale des Machines Electriques, Habilitation à diriger des recherches, Université des Sciences et Techniques de Lille, France, December 2007, In French.

[CAN 10] M. Cantegrel, S. Brisset, F. Gillon, and M. Diguet, "An investigation regarding the profitability of railways energy storage packs," in *11th International Workshop on Optimization and Inverse Problems in Electromagnetism, OIPE 2010, 14-18 September, Sofia, Bulgaria*, 2010.

[CAN 10a] M. Cantegrel, S. Brisset, and F. Gillon, "An investigation for improving the use of tramways energy storage pack," in *11th International Workshop on Optimization and Inverse Problems in Electromagnetism, OIPE 2010, 14-18 September, Sofia, Bulgaria*, 2010.

[CAN 11] M. Cantegrel, S. Brisset, F. Gillon, and A. C. Berbecea, "Optimisation systémique d'un système de stockage embarqué à bord d'un tramway," in *EF 2011 Electrotechnique du Futur, 14-15 December 2011, Belfort, France, in French*, 2011.

[CHI 07] A. Chinchuluun and P. Pardalos, "A survey of recent developments in multiobjective optimization," *Annals of Operations Research*, vol. 154, no. 1, pp. 29-50, 2007.

[CHO 05] R. Choudhary, A. Malkawi, and P.Y. Papalambros, "Analytic Target Cascading in simulation-based building design," *Automation in Construction*, vol. 14, no. 4, pp. 551-568, 2005, 20th International Symposium on Automation and Robotics in Construction.

[COB 12] Cobham, "Cobham Company's Opera 3D v15.0 FE commercial software: http://www.cobham.com/about-cobham/aerospace-and-security/about-us/antenna-systems/kidlington/products/opera-3d.aspx," 2012. [Online]. http://www.cobham.com/about-cobham/aerospace-and-security/about-us/antenna-systems/kidlington/products/opera-3d.aspx

[COE 07] C.A.C. Coello, G.B. Lamont, and D.A. Van Veldhuizen, *Evolutionary algorithms for solving multi-objective problems*.: Springer-Verlag New York Inc., vol. 5, 2007.

[COL 02] Y. Collette and P. Siarry, *Optimisation multiobjectif*.: Eyrolles, 2002.

[COU 10] I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, and L. Knockaert, "Surrogate-based infill optimization applied to electromagnetic problems," *International Journal of RF and Microwave Computer-Aided Engineering*, vol. 20, no. 5, pp. 492-501, July 2010.

[COX 97] D.D. Cox and S. John, "SDO: A statistical method for global optimization," *Multidisciplinary design optimization: State-of-the-Art, SIAM, Philadelphia, PA*, vol. 1, pp. 315-329, 1997.

[CRA 94] E.J. Cramer, J.E. Dennis, P.D. Frank, R.M. Lewis, and G.R. Shubin, "Problem Formulation for Multidisciplinary Optimization," *SIAM Journal on Optimization*, vol. 4, pp. 754-776, 1994.

[DEB 02] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, April 2002.

[DEP 07] P. Dépincé, B. Guédas, and J. Picard, "Multidisciplinary and multiobjective optimization: Comparison of several methods," in *7th World Congress on Structural and Multidisciplinary Optimization, Seoul, Korea*, 2007.

[DIB 10] P. DiBarba, *Multiobjective Shape Design in Electricity and Magnetism, 1st edition*.: Springer, vol. 47, 2010.

[ECH 08] D. Echeverría and P. Hemker, "Manifold mapping: a two-level optimization technique," *Computing and Visualization in Science*, vol. 11, no. 4-6, pp. 193-206, 2008.

[ENC 07] L. Encica et al., "Efficient optimal design of electromagnetic actuators using space mapping," *Structural and Multidisciplinary Optimization*, vol. 33, no. 6, pp. 481-491, 2007.

[ENC 08] L. Encica, J.J.H. Paulides, E.A. Lomonova, and A.J.A. Vandenput, "Aggressive Output Space-Mapping Optimization for Electromagnetic Actuators," *IEEE Transactions on Magnetics*, vol. 44, no. 6, pp. 1106-1109, June 2008.

[EST 12] Esteco, "Esteco Company's modeFRONTIER v4.4.2 optimization commercial software: http://www.esteco.com/home/mode_frontier/mode_frontier.html," 2012. [Online]. http://www.esteco.com/home/mode_frontier/mode_frontier.html

[ETM 05] L.F.P. Etman, M. Kokkolaras, A.T. Hofkamp, P.Y. Papalambros, and J.E. Rooda, "Coordination specification in distributed optimal design of multilevel systems using the X language," *Structural and Multidisciplinary Optimization*, vol. 29, no. 3, pp. 198-212, 2005.

[FER 09] S. Ferguson, E. Kasprzak, and K. Lewis, "Designing a family of reconfigurable vehicles using multilevel multidisciplinary design optimization," *Structural and Multidisciplinary Optimization*, vol. 39, no. 2, pp. 171-186, 2009.

[FOR 08] A.I.J. Forrester, A. Sóbester, and A.J. Keane, *Engineering Design via Surrogate Modelling: A Practical Guide*.: Wiley, July 2008.

[FOR 09] A.I.J. Forrester and A.J. Keane, "Recent advances in surrogate-based optimization," *Progress in Aerospace Sciences*, vol. 45, no. 1-3, pp. 50-79, January 2009.

[GIL 09] F. Gillon, Méthodologies de Conception Optimale des Composants Electromagnétiques, Habilitation à diriger des recherches, Université des Sciences et Techniques de Lille, France, June 2009, In French.

[GIN 10] D. Ginsbourger, R. Le Riche, and L. Carraro, "Kriging Is Well-Suited to Parallelize Optimization," in *Computational Intelligence in Expensive Optimization Problems*, Yoel Tenne et al., Eds.: Springer Berlin Heidelberg, vol. 2, pp. 131-162, 2010.

[GON 11] J. Gong, A. C. Berbecea, F. Gillon, X. Cimetière, and P. Brochet, "Optimal design of a Double-Sided Linear Induction Motor using Efficient Global Optimization," in *8th International Symposium on Linear Drives for Industrial Applications, LDIA 2011*, July 2011.

[GON 11a] J. Gong, "Modélisation et conception optimale d'un moteur linéaire à induction pour système de traction ferroviaire," Université des Sciences et Technologie de Lille - Lille 1, Lille, France, Ph.D. dissertation, October 2011.

[GOU 06] J. Goupy and L. Creighton, *Introduction aux plans d'expériences, 3e édition*.: Dunod, 2006.

[GUA 11] P. Guarneri, M. Gobbi, and P.Y. Papalambros, "Efficient multi-level design optimization using Analytical Target Cascading and Sequential Quadratic Programming," *Structural and Multidisciplinary Optimization*, vol. 44, no. 3, pp. 351-362, 2011.

[GUT 01] H.M. Gutmann, "A radial basis function method for global optimization," *Journal of Global Optimization*, vol. 19, no. 3, pp. 201-227, 2001.

[HAW 07] G. Hawe and J.K. Sykulski, "Considerations of Accuracy and Uncertainty with Kriging Surrogate Models in Single-Objective Electromagnetic Design Optimization," *IET Science, Measurement \& Technology*, vol. 1, no. 1, pp. 37-47, January 2007.

[HAW 07a] G.I. Hawe and J.K. Sykulski, "An Enhanced Probability of Improvement Utility Function for Locating Pareto Optimal Solutions," in *16th Conference on the Computation of Electromagnetic Fields, COMPUMAG, Aachen, Germany*, vol. 3, pp. 965-966, June 2007.

[HAW 08] G.I. Hawe and J.K. Sykulski, "A Scalarizing One-Stage Algorithm for Efficient Multi-Objective Optimization," *IEEE Transactions on Magnetics*, vol. 44, no. 6, pp. 1094-1097, June 2008.

[HEM 08] T. Hemker, "Derivative Free Surrogate Optimization for Mixed-Integer Nonlinear

Black Box Problems in Engineering," TU Darmstadt, Ph.D. dissertation, December 2008.

[HOL 08] K. Holmström, "An adaptive radial basis algorithm (ARBF) for expensive black-box global optimization," *Journal of Global Optimization*, vol. 41, no. 3, pp. 447-464, 2008.

[HUA 06] G.Q. Huang, T. Qu, D.W.L. Cheung, and L. Liang, "Extensible multi-agent system for optimal design of complex systems using Analytical Target Cascading," *The International Journal of Advanced Manufacturing Technology*, vol. 30, no. 9-10, pp. 917-926, 2006.

[JAN 05] B.S. Jang, Y.S. Yang, H.S. Jung, and Y.S. Yeun, "Managing approximation models in collaborative optimization," *Structural and Multidisciplinary Optimization*, vol. 30, no. 1, pp. 11-26, 2005.

[JEO 05] S. Jeong and S. Obayashi, "Efficient global optimization (EGO) for multi-objective problem and data mining," in *The 2005 IEEE Congress on Evolutionary Computation*, vol. 3, pp. 2138-2145, 2005.

[JEO 06] S. Jeong, Y. Minemura, and S. Obayashi, "Optimization of combustion chamber for Diesel engine using kriging model," *Journal of Fluid Science and Technology*, vol. 1, no. 2, pp. 138-146, 2006.

[JON 98] D.R. Jones, M. Schonlau, and W.J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global Optimization*, vol. 13, no. 4, pp. 455-492, 1998.

[JON 01] D.R. Jones, "A Taxonomy of Global Optimization Methods Based on Response Surfaces," *Journal of Global Optimization*, vol. 21, no. 4, pp. 345-383, 2001.

[KEA 06] A.J. Keane, "Statistical improvement criteria for use in multiobjective design optimization," *AIAA journal*, vol. 44, no. 4, pp. 879-891, April 2006.

[KIM 01] H.M. Kim, "Target Cascading in optimal system design," The University of Michigan, Michigan, USA, Mechanical Engineering, 2001.

[KIM 03] H.M. Kim, N.F. Michelena, P.Y. Papalambros, and T. Jiang, "Target cascading in optimal system design," *Journal of Mechanical Design*, vol. 125, no. 3, pp. 474-480, 2003.

[KIM 03a] H.M. Kim, D.G. Rideout, P.Y. Papalambros, and J.L. Stein, "Analytical Target Cascading in Automotive Vehicle Design," *Journal of Mechanical Design*, vol. 125, no. 3, pp. 481-489, 2003.

[KIM 06] H.M. Kim, W. Chen, and M.M. Wiecek, "Lagrangian coordination for enhancing the convergence of Analytical Target Cascading," *AIAA journal*, vol. 44, no. 10, p. 2197, 2006.

[KNO 05] J. Knowles and E.J. Hughes, "Multiobjective Optimization on a Budget of 250 Evaluations," in *Evolutionary Multi-Criterion Optimization*, Carlos Coello Coello, Arturo Hernandez Aguirre, and Eckart Zitzler, Eds.: Springer-Verlag Berlin Heidelberg, vol. 3410, pp. 176-190, 2005.

[KNO 05a] J. Knowles, "ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 50-66, February 2005.

[KNO 09] J. Knowles, D. Corne, and A. Reynolds, "Noisy Multiobjective Optimization on a Budget of 250 Evaluations," in *Evolutionary Multi-Criterion Optimization*, Matthias Ehrgott et al., Eds.: Springer-Verlag Berlin Heidelberg, vol. 5467, pp. 36-50, 2009.

[KOK 02] M. Kokkolaras, R. Fellini, H.M. Kim, N.F. Michelena, and P.Y. Papalambros, "Extension of the target cascading formulation to the design of product families," *Structural and Multidisciplinary Optimization*, vol. 24, no. 4, pp. 293-301, 2002.

[KRE 08] S. Kreuawan, "Modelling and optimal design in railway applications," Ecole Centrale de Lille, Lille, France, Ph.D. dissertation, November 2008.

[KRE 09] S. Kreuawan et al., "Optimal design of a traction system using Target Cascading," *International Journal of Applied Electromagnetics and Mechanics, IJAEM*, vol. 30, no. 3-4, pp. 163-178, 2009.

[KRO 00] I. Kroo and V. Manning, "Collaborative optimization: status and directions," in *The 8th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Long Beach, CA, USA*, vol. 6, p. 8, September 2000.

[LI 08] X. Li, W. Li, and C. Liu, "Geometric analysis of collaborative optimization," *Structural and Multidisciplinary Optimization*, vol. 35, no. 4, pp. 301-313, 2008.

[LI 08a] Y. Li, Z. Lu, and J.J. Michalek, "Diagonal Quadratic Approximation for parallelization of Analytical Target Cascading," *Journal of Mechanical Design*, vol. 130, p. 051402, 2008.

[LAS 05] J.B. Lassiter, M.M. Wiecek, and K.R. Andrighetti, "Lagrangian coordination and Analytical Target Cascading: Solving ATC-decomposed problems with Lagrangian Duality," *Optimization and Engineering*, vol. 6, no. 3, pp. 361-381, 2005.

[LOP 02] S.N. Lophaven, H.B. Nielsen, and J. Søndergaard, "DACE - A MATLAB Kriging Toolbox, version 2.0," Technical University of Denmark, DTU, DK-2800 Kgs. Lyngby – Denmark, software toolbox, August 2002. [Online]. http://www.imm.dtu.dk/~hbn/dace

[MAG 08] C. Magele, P. Alotto, M. Repetto, F. Freschi, and A. Köstinger, "TEAM22 benchmark problem of the International Compumag Society, available online at: http://www.igte.tugraz.at/team22/," International Compumag Society, Tech. rep., 2008. [Online]. http://www.igte.tugraz.at/team22/

[MAS 06] G. Mastinu, M. Gobbi, and C. Miano, "Multi-objective Optimisation," in *Optimal Design of Complex Mechanical Systems*.: Springer Berlin Heidelberg, vol. Part I, pp. 47-98, 2006.

[MCK 00] M.D. McKay, R.J. Beckman, and W.J. Conover, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 42, no. 1, pp. 55-61, February 2000.

[MES 07] V. Mester, "Conception Optimale Systémique des Composants des Chaînes de Traction Electrique," Ecole Centrale de Lille, Lille, France, Ph.D. dissertation, May 2007.

[MIC 99] N. Michelena, H.M. Kim, and P.Y. Papalambros, "A system partitioning and optimization approach to target cascading," in *12th International Conference on Engineering Design, Munich, Germany*, pp. 1109-1112, August 1999.

[MIC 99a] N. Michelena, H.A. Park, P.Y. Papalambros, and D. Kulkarni, "Hierarchical

Overlapping Coordination for Large-Scale Optimization by Decomposition," *AIAA Journal*, vol. 37, no. 7, pp. 890-896, 1999.

[MIC 99b] N. Michelena, C. Scheffer, R. Fellini, and P.Y. Papalambros, "CORBA-Based Object-Oriented Framework for Distributed System Design," *Journal of Structural Mechanics*, vol. 27, no. 4, pp. 365-392, 1999.

[MIC 03] N. Michelena, H. Park, and P.Y. Papalambros, "Convergence properties of Analytical Target Cascading," *AIAA Journal*, vol. 41, no. 5, pp. 897-905, 2003.

[MIC 05] J.J Michalek, "Preference Coordination in Engineering Design Decision-Making," Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI, USA, Mechanical Engineering, 2005.

[MIC 05a] J.J. Michalek and P.Y. Papalambros, "An Efficient Weighting Update Method to Achieve Acceptable Consistency Deviation in Analytical Target Cascading," *Journal of Mechanical Design*, vol. 127, no. 2, pp. 206-214, 2005.

[MIC 05b] J.J. Michalek and P.Y. Papalambros, "Weights, Norms, and Notation in Analytical Target Cascading," *Journal of Mechanical Design*, vol. 127, no. 3, pp. 499-501, 2005.

[MIC 06] J.J. Michalek and P.Y. Papalambros, "BB-ATC: Analytical Target Cascading Using Branch and Bound for Mixed-Integer Nonlinear Programming," in *ASME Conference proceedings, paper no. DETC2006-99040*, vol. 1, pp. 685-691, 2006.

[MIE 99] K. Miettinen, *Nonlinear multiobjective optimization*.: Springer, vol. 12, 1999.

[MIN 11] H. Minghong, "Application of an Improved Collaborative Optimization Method for Speed Reducer Optimization," in *Third International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, vol. 1, pp. 519-522, January 2011.

[MOO 96] B. Moore, R.D. Braun, and I.M. Kroo, "Use Of The Collaborative Optimization Architecture For Launch Vehicle Design," in *6th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, pp. 306-318, 1996.

[MOS 10] J.-P. Moskowitz and J.-L. Cohuau, "STEEM: ALSTOM and RATP experience of supercapacitors in tramway operation," in *Vehicle Power and Propulsion Conference (VPPC), 2010 IEEE*, pp. 1-5, September 2010.

[MOU 08] F. Moussouni et al., "Analytical Target Cascading for Optimal Design of Railway traction System," in *International Conference on Engineering Optimization (EngOpt 2008), Rio de Janeiro, Brazil*, June 2008.

[MOU 09] F. Moussouni et al., "Multi-level design optimization using target cascading, an improvement of convergence," *International Journal for Computation and Mathematics in Electrical and Electronic Engineering, COMPEL*, vol. 28, no. 5, pp. 1162-1178, 2009.

[MOU 09a] F. Moussouni-Messad, "Multi-level and multi-objective design optimization tools for handling complex systems," Ecole Centrale de Lille, Lille, France, Ph.D. dissertation, July 2009.

[NAK 09] H. Nakayama, Y. Yun, and M. Yoon, "Basic Concepts of Multi-objective Optimization," in *Sequential Approximate Multiobjective Optimization Using Computational Intelligence*.: Springer-Verlag Berlin Heidelberg, pp. 1-15, 2009.

[NEI 96] P. Neittaanmäki, M. Rudnicki, and A. Savini, *Inverse problems and optimal design in electricity and magnetism*.: Oxford University Press, USA, vol. 35, 1996.

[NOE 12] Noesis, "Noesis Solutions Company's Optimus v5.3 Rev9 optimization commercial software," 2012. [Online]. http://www.noesissolutions.com/Noesis/

[PAP 08] M. Pappalardo, "Multiobjective Optimization: A Brief Overview," in *Pareto Optimality, Game Theory And Equilibria*, Altannar Chinchuluun et al., Eds.: Springer New York, vol. 17, pp. 517-528, 2008.

[PAR 09] Y. Parte, "Quelques techniques de couplage de modèles et de données, in English," Université Toulouse III - Paul Sabatier, Toulouse, France, Applied Mathematics, November 2009.

[PAR 10] J.M. Parr, C.M.E. Holden, A.I.J. Forrester, and A.J. Keane, "Review of Efficient Surrogate Infill Sampling Criteria with Constraint Handling," in *2nd International Conference on Engineering Optimization, EngOpt 2010, Lisbon, Portugal*, September 2010.

[RAB 07] S. Rabeau, P. Dépincé, and F. Bennis, "Collaborative Optimization of complex systems: a multidisciplinary approach," *International Journal on Interactive Design and Manufacturing*, vol. 1, no. 4, pp. 209-218, 2007.

[RAH 99] Y. Rahmat-Samii and E. Michielssen, *Electromagnetic Optimization by Genetic Algorithms*.: John Wiley \& Sons Inc., 1999.

[ROS 09] M. Rossi, "Conception holistique de transformateurs pour une application ferroviaire," in *JCGE 2009, Compiegne, France*, 2009.

[ROT 08] B.D. Roth and I.M. Kroo, "Enhanced Collaborative Optimization: A Decomposition-Based Method for Multidisciplinary Design," in *ASME 2008 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2008), August 3–6, 2008 , Brooklyn, New York, USA*, pp. 927-936, 2008.

[ROT 08a] B.D. Roth and I.M. Kroo, "Enhanced Collaborative Optimization: Application to an Analytic Test Problem and Aircraft Design," in *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, September 10-12, 2008, Victoria, BC, USA*, 2008.

[SAC 89] J. Sacks, W.J. Welch, T.J. Mitchell, and H.P. Wynn, "Design and analysis of computer experiments," *Statistical science*, vol. 4, no. 4, pp. 409-423, November 1989.

[SAS 00] M.J. Sasena, P.Y. Papalambros, and P. Goovaerts, "Metamodeling sampling criteria in a global optimization framework," in *8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, pp. AIAA-2000-4921, September 2000.

[SAS 01] M.J. Sasena, P.Y. Papalambros, and P. Goovaerts, "The use of surrogate modeling algorithms to exploit disparities in function computation time within simulation-based optimization," in *The Fourth World Congress of Structural and Multidisciplinary Optimization*, vol. 2, pp. 1-6, 2001.

[SAS 02] M.J. Sasena, P.Y. Papalambros, and P. Goovaerts, "Exploration of metamodeling sampling criteria for constrained global optimization," *Engineering Optimization*, vol. 34, no. 3, pp. 263-278, 2002.

[SAS 02a] M.J. Sasena, P.Y. Papalambros, and P. Goovaerts, "Global optimization of problems

with disconnected feasible regions via surrogate modeling," in *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, pp. AIAA-2002-5573, Septembre 2002.

[SAS 02b] M.J. Sasena, "Flexibility and Efficiency Enhancements for Constrained Global Design Optimization with Kriging Approximations," University of Michigan, USA, Ph.D. dissertation, 2002.

[SCH 97] M. Schonlau, "Computer Experiments and Global Optimization," University of Waterloo, Waterloo, Ont., Canada, Ph.D. dissertation, 1997.

[SOB 88] J. Sobieszczanski-Sobieski, "Optimization by decomposition: A step from hierarchic to non-hierarchic systems," in *Proceedings of the 2nd NASA/USAF Symposium on Recent Advances in Multidisciplinary Analysis and Optimization, Hampton, Virginia, USA*, 1988.

[SOB 98] J. Sobieszczanski-Sobieski, J.S. Agte, R.R. Sandusky, and Langley Research Center, "Bi-level integrated system synthesis (BLISS)," NASA Center for AeroSpace Information (CASI), Tech. rep., August 1998.

[SOB 00] I. Sobieski and I. Kroo, "Collaborative optimization using response surface estimation," *AIAA Journal*, vol. 38, no. 10, pp. 1931-1938, 2000.

[SOB 00a] J. Sobieszczanski-Sobieski, M.S. Emiley, J.S. Agte, and R.R. Jr Sandusky, "Advancement of bi-level integrated system synthesis (BLISS)," in *AIAA journal*, 2000.

[SOB 04] A. Sóbester, S.J. Leary, and A.J. Keane, "A parallel updating scheme for approximating and optimizing high fidelity computer simulations," *Structural and multidisciplinary optimization*, vol. 27, no. 5, pp. 371-383, 2004.

[SOB 05] A. Sóbester, S.J. Leary, and A.J. Keane, "On the Design of Optimization Strategies Based on Global Response Surface Approximation Models," *Journal of Global Optimization*, vol. 33, pp. 31-59, 2005.

[TOS 06] S. Tosserams, L.F.P. Etman, P.Y. Papalambros, and J.E. Rooda, "An augmented Lagrangian Relaxation for Analytical Target Cascading using the alternating direction method of multipliers," *Structural and Multidisciplinary Optimization*, vol. 31, no. 3, pp. 176-189, 2006.

[TOS 07] S. Tosserams, L.F.P. Etman, and J.E. Rooda, "An augmented Lagrangian decomposition method for quasi-separable problems in MDO," *Structural and Multidisciplinary Optimization*, vol. 34, no. 3, pp. 211-227, 2007.

[TOS 08] S. Tosserams, M. Kokkolaras, L.F.P. Etman, and J.E. Rooda, "Extension of Analytical Target Cascading using Augmented Lagrangian Coordination for Multidisciplinary Design Optimization," in *Proceedings of the 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Victoria, British Columbia, Canada*, 2008.

[TOS 08a] S. Tosserams, L.F.P. Etman, and J.E. Rooda, "Augmented Lagrangian coordination for distributed optimal design in MDO," *International Journal for Numerical Methods in Engineering*, vol. 73, no. 13, pp. 1885-1910, 2008.

[TOS 09] S. Tosserams, L.F.P. Etman, and J.E. Rooda, "Block-separable linking constraints in augmented Lagrangian coordination," *Structural and Multidisciplinary Optimization*, vol. 37, no. 5, pp. 521-527, 2009.

[TOS 10] S. Tosserams, A.T. Hofkamp, L.F.P. Etman, and J.E. Rooda, "A specification language for problem partitioning in decomposition-based design optimization," *Structural and Multidisciplinary Optimization*, vol. 42, no. 5, pp. 707-723, 2010.

[TRA 07] T.V. Tran, S. Brisset, and P. Brochet, "A Benchmark for Combinatorial and Multi-level Optimization of a Safety Isolating Transformer," in *International Conference on the Computation of Magnetic Fields, COMPUMAG 2007, Aachen, Germany*, June 2007.

[TRA 07a] T.V. Tran, "Isolation transformer optimization benchmark, available on-line at: http://l2ep.univ-lille1.fr/come/benchmark-transformer.htm," Ecole Centrale de Lille, France, Tech. rep., 2007. [Online]. http://l2ep.univ-lille1.fr/come/benchmark-transformer.htm

[TRA 09] T.V. Tran, "Problèmes combinatoires et modèles multi-niveaux pour la conception optimale des machines électriques," Ecole Centrale de Lille, Lille, France, Ph.D. dissertation, June 2009.

[TZE 03] N. Tzevelekos et al., "An empirical local convergence study of alternative coordination schemes in Analytical Target Cascading," in *Proceedings of the 5th World Congress on Structural and Multidisciplinary Optimization*, pp. 19-23, 2003.

[VEL 99] D.A. van Veldhuizen, "Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations," Airforce Institute of Technology, Wright-Patterson AFB, Ohio, USA, Ph.D. dissertation, June 1999.

[VEL 00] D.A. van Veldhuizen and G.B. Lamont, "On measuring multiobjective evolutionary algorithm performance," in *Proceedings of the 2000 Congress on Evolutionary Computation*, vol. 1, pp. 204-211, 2000.

[VEN 01] P. Venkataraman, *Applied Optimization with MATLAB Programming*.: Wiley-Interscience, 2001. [Online]. http://www.wiley.com/legacy/products/subject/engineering/venkat/

[VIL 09] J. Villemonteix, E. Vazquez, M. Sidorkiewicz, and E. Walter, "Global optimization of expensive-to-evaluate functions: an empirical comparison of two sampling criteria," *Journal of Global Optimization*, vol. 43, no. 2-3, pp. 373-389, 2009.

[VIV 02] S. Vivier, "Stratégies d'optimisation par la méthode des Plans d'Expériences, et Application aux dispositifs électrotechniques modélisés par Eléments Finis," Université des Sciences et Technologie de Lille - Lille 1, Lille, France, Ph.D. dissertation, July 2002.

[WAN 07] G.G. Wang and S. Shan, "Review of Metamodeling Techniques in Support of Engineering Design Optimization," *Journal of Mechanical Design*, vol. 129, no. 4, pp. 370-380, 2007.

[WAN 09] N. Wang, L. Wang, Y. Bu, G. Zhang, and L. Shen, "Tactical Aircraft Pop-Up Attack Planning Using Collaborative Optimization," in *Emerging Intelligent Computing Technology and Applications. With Aspects of Artificial Intelligence*, De-Shuang Huang et al., Eds.: Springer-Verlag Berlin Heidelberg, vol. 5755, pp. 361-370, 2009.

[WAT 95] A.G. Watson and R.J. Barnes, "Infill sampling criteria to locate extremes," *Mathematical Geology*, vol. 27, no. 5, pp. 589-608, 1995.

[WEG 90] E.J. Wegman, "Hyperdimensional Data Analysis Using Parallel Coordinates," *Journal of the American Statistical Association*, vol. 85, no. 411, pp. 664-675, 1990.

[XIA 10] M. Xiao, L. Gao, H.B. Qiu, X.Y. Shao, and X.Z. Chu, "An Approach Based on Enhanced Collaborative Optimization and Kriging Approximation in Multidisciplinary Design Optimization," *Advanced Materials Research*, vol. 118, pp. 399-403, 2010.

[XIA 11] S. Xiao, M. Rotaru, and J.K. Sykulski, "Exploration versus Exploitation Using Kriging Surrogate Modelling in Electromagnetic Design," *ISEF 2011 - XV International Symposium on Electromagnetic Fields in Mechatronics, Electrical and Electronic Engineering*, vol. 1, p. OS.1.6, September 2011, Funchal, Madeira, September 1-3, 2011.

[ZAD 09] P.M. Zadeh, V.V. Toropov, and A.S. Wood, "Metamodel-based collaborative optimization framework," *Structural and Multidisciplinary Optimization*, vol. 38, no. 2, pp. 103-115, 2009.

[ZHA 07] M. Zhao and W.C. Cui, "Application of the optimal Latin hypercube design and radial basis function network to collaborative optimization," *Journal of Marine Science and Application*, vol. 6, no. 3, pp. 24-32, 2007.

[ZHA 11] M. Zhao and W. Cui, "On the development of Bi-Level Integrated System Collaborative Optimization," *Structural and Multidisciplinary Optimization*, vol. 43, no. 1, pp. 73-84, January 2011.

[ZIT 99] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257-271, November 1999.

Résumé étendu en français

Le processus classique de conception, traditionnellement utilisé par les ingénieurs et qui consiste en une approche par essais et erreurs pour la sélection et la validation, ne répond plus aux exigences de délais très courts, de limitation des ressources financières et de « respect de l'environnement ». Ces exigences sont directement dictées par le contexte actuel d'un marché globalisé. « Mieux, plus vite, moins cher! » est le slogan qui décrit le mieux les exigences qui guident le processus de conception industrielle d'aujourd'hui. Ce nouveau besoin est confirmé par de nombreuses études récentes concernant la recherche industrielle dans divers domaines de l'ingénierie, dont les plus notables sont l'aéronautique, l'automobile, l'électronique et l'industrie chimique.

Pour toutes les questions précédemment invoqués et la complexité sans cesse croissante de la tâche de conception réalisée par les ingénieurs, l'optimisation se présente comme une réponse naturelle en aidant le concepteur dans le processus de prise de décision. De nombreuses techniques d'optimisation ont été développées au fil du temps, surtout au cours des deux dernières décennies, alors que les ressources de calcul ont connu un développement exponentiel. Aujourd'hui, la conception assistée par ordinateur (CAO) et l'ingénierie assistée par ordinateur (IAO) représentent des outils d'analyse puissants, qui sont utilisés dans tous les domaines de l'ingénierie, tels que l'électricité, la mécanique, la thermique, l'acoustique, etc. pour simuler le comportement des dispositifs ou des systèmes à concevoir. Ces codes de simulation se sont imposés en tant que prototypes virtuels des dispositifs à concevoir. La tendance naturelle est d'**introduire** ces **outils de simulation dans le processus d'optimisation**, afin de bénéficier de la **précision** offerte par ces outils. Cependant, l'intégration de ces outils de simulation dans le processus d'optimisation soulève un certain nombre de questions qui doivent être abordées, principalement en raison du **temps de calcul rédhibitoire** due aux nombreuses évaluations successives de ces modèles, requises par les algorithmes classiques d'optimisation. Le **grand nombre de contraintes** de faisabilité **et des objectifs** contradictoires qui doivent être prises en compte au sein du processus de conception alourdit la tâche de l'optimisation. En outre, une approche de conception optimale plus systémique est souhaitée, car des interactions fortes sont exercées entre les composants d'un système. La nature multidisciplinaire des systèmes complexes nécessite l'intégration de toutes les disciplines impliquées dans le processus de conception. Pour résoudre tous ces problèmes, des techniques d'optimisation adaptées sont nécessaires et l'élaboration de telles approches d'optimisation fait l'objet de ce travail de recherche. L'accent dans ce manuscrit est mis sur les méthodologies de conception optimale de systèmes complexes, qui trouvent leur application dans de nombreux domaines de l'ingénierie en général, et dans les systèmes ferroviaires en particulier. La taille importante des problèmes à être traités, ainsi que l'utilisation des outils de simulation spécifiques à

chaque domaine pour l'analyse du comportement des divers dispositifs ou des composants du système, nécessite l'utilisation de **méthodes d'optimisation adaptées**.

La présentation des travaux de recherche dans ce manuscrit est repartie en trois chapitres. Le **premier chapitre** introduit le contexte du travail de recherche et présente les aspects généraux de la méthodologie de conception optimale des systèmes complexes. Le **deuxième chapitre** du manuscrit est dédié aux méthodologies d'optimisation de problèmes complexes par la construction de surfaces de réponse par la méthode du krigeage. Deux applications d'optimisation de dispositifs électromagnétiques, un moteur linéaire à induction et un dispositif de stockage d'énergie magnétique par anneaux supraconducteurs sont explorées dans ce chapitre. Dans le **troisième et dernier chapitre** du manuscrit sont explorées les formulations de problèmes d'optimisation de systèmes complexes par la décomposition du système selon différents critères. L'optimisation d'un transformateur d'isolation basse-tension monophasé est utilisée en tant qu'application d'une méthode d'optimisation par décomposition disciplinaire bi-niveaux. Un cas test de conception optimale, de la Société Alstom Transport, d'un système de stockage d'énergie à base de super-capacités embarqué à bord d'un tramway est résolu en utilisant une stratégie d'optimisation par décomposition physique du système.

Chapitre 1 Outils d'aide à la décision pour la conception de systèmes électromagnétiques complexes

Le premier chapitre représente une partie introductive à la problématique de l'optimisation mono-objectif et multi-objectif en général.

L'exposé débute avec la description des aspects généraux d'un processus de conception optimale. La conception optimale est vue comme un processus qui comporte trois étapes : une étape préliminaire de formulation du problème d'optimisation, l'étape la plus lourde du déroulement du processus d'optimisation et une étape finale de visualisation, analyse et interprétation des résultats, terminée par la prise de décision concernant la configuration optimale à retenir en pratique. Ces trois étapes sont interdépendantes et la qualité du résultat du processus de conception optimale dépend fortement de la façon dont chacune de ces trois étapes est abordée. Une description de ces étapes est fournie. Pendant chacune de ces étapes, des outils spécifiques sont employés par le concepteur.

L'analyse des modèles occupe une partie importante de la phase préliminaire de formulation du processus de conception optimale. Cela consiste à analyser le comportement des modèles, ainsi que l'influence des différentes variables et leur corrélation avec les sorties des modèles. Parmi les outils d'analyse qualitative des modèles, les coefficients de corrélation de Pearson et de Spearman représentent des outils puissants pour obtenir un aperçu des relations fonctionnelles d'un modèle. Ces deux coefficients de corrélation sont décrits, ainsi que leur représentation graphique sous la forme d'une matrice de corrélation.

L'exposé continue avec l'introduction de la formulation mathématique d'un problème d'optimisation mono-objectif. L'optimalité locale et globale d'une configuration est discuté. Pour la

résolution d'un problème d'optimisation mono-objectif, il existe deux catégories d'algorithmes : les algorithmes déterministes et les algorithmes stochastiques. Pour les algorithmes basés sur le calcul du gradient, des précautions sont à prendre lors de leur utilisation, telles que la mise à l'échelle des variables de conception et l'usage de plusieurs points de départ pour éviter de tomber sur un optimum local (en anglais : multi-start). Quelques algorithmes courants dans chaque catégorie sont cités.

L'aspect multicritère est ensuite abordé. La formulation générique d'un problème d'optimisation multi-objectif est introduite. Contrairement au cas mono-objectif, un problème d'optimisation multi-objectif n'a pas qu'un seul optimum, mais un ensemble de solutions optimales. Ainsi, l'optimalité au sens de Pareto représente la mesure de l'efficacité dans un contexte multi-objectif. La notion de dominance au sens de Pareto est introduite et discuté pour des problèmes ayant la forme du front de Pareto convexe, ainsi que concave. Même si le résultat d'un problème d'optimisation multi-objectif consiste en un ensemble de solutions de compromis entre les différents objectifs définis, en pratique un seul ou quelques configurations optimales sont retenues pour être conçues. Une prise de décision doit être faite à un moment du processus de conception optimale. En fonction du moment, il existe trois catégories de méthodes d'optimisation : méthodes « a priori », méthodes « a posteriori » et méthodes interactives. Pour les méthodes « a priori », le compromis entre les différents objectifs est fixé avant de lancer le processus d'optimisation. Les méthodes dites « a posteriori » sont les méthodes d'optimisation multi-objectif, qui fournissent un ensemble de solutions optimales parmi lesquelles le concepteur va faire son choix. Pour les méthodes d'optimisation interactives, la prise de décision est faite au fur et à mesure de l'avancement du processus d'optimisation. Ces méthodes sont moins courantes.

Pour les méthodes d'optimisation « a priori », une transformation de la formulation multi-objectif doit être faite en une formulation mono-objectif, afin de fixer le compromis avant de lancer le processus d'optimisation. Différentes techniques de transformation d'un problème d'optimisation, multi-objectif en mono-objectif sont recensées. Parmi ceux-ci, la technique de *pondération des objectives*, la méthode *epsilon-contrainte* et la méthode du *but à atteindre* sont présentées. Afin d'illustrer le comportement de chacune de ces techniques de transformation, un exemple simple de problème d'optimisation bi-objectif avec un front de Pareto concave est considéré.

Dans l'industrie, les dispositifs et systèmes à concevoir sont complexes, nécessitant des techniques d'optimisation bien adaptées aux particularités de chaque problème. La complexité de ces systèmes est donnée principalement par :

- le calcul informatique couteux en termes de ressources et de temps de calcul des modèles de simulation employés ;
- le grand nombre des composants du système, avec un impact sur la taille du problème à résoudre.

Pour répondre au premier aspect de la complexité des systèmes industriels évoqués, une solution efficace consiste en l'utilisation des méta-modèles dans le processus d'optimisation. Cet aspect fait l'objet du deuxième chapitre du manuscrit.

Les stratégies d'optimisation basées sur la décomposition de système complexe sont destinées à gérer la taille importante de tels systèmes industriels. Différentes types de décompositions peuvent

être considérés, comme la décomposition *orientée objet*, la décomposition *disciplinaire* ou la décomposition *basé sur les modèles*. En fonction du nombre d'optimiseurs impliqués dans le processus d'optimisation d'un système complexe, ils se distinguent deux stratégies : décomposition mono-niveau et multi-niveaux. Les stratégies multi-niveaux font l'objet d'étude du troisième chapitre de ce manuscrit.

Une partie également importante du processus de conception optimale consiste en l'analyse et l'interprétation des résultats issus de l'optimisation.

Afin d'évaluer la qualité d'un front de Pareto ou de comparer objectivement deux fronts de Pareto, un certain nombre de mesures ont été développés et existent dans la littérature. Trois mesures différentes et complémentaires pour évaluer la qualité d'un front de Pareto sont présentées : la distance générationnelle (en anglais : Generational Distance – GD), l'espacement (en anglais : Spacing – S) et le rapport d'erreur (en anglais : Error Ratio – ER).

Un certain nombre d'outils graphiques pour la représentation des données multidimensionnelles existent dans la littérature et sont destinés à aider le concepteur dans le processus de prise de décision. Parmi ceux-ci, on peut citer : les boîtes à moustaches (en anglais : box-plots), les diagrammes de dispersion (en anglais : scatter plots), les boîtes à bulles (en anglais : bubble plots) et les représentations par coordonnées parallèles (en anglais : parallel coordinates representation).

A la fin de ce chapitre, quelques outils d'optimisation commerciaux sont cités.

Chapitre 2 Optimisation basée sur méta-modèles

Dans ce chapitre, l'intégration de méta-modèles dans le processus de conception optimale est étudiée. Le processus de conception optimale basée sur l'utilisation de méta-modèles est appelé : optimisation basée sur les méta-modèles (en anglais : Metamodel-based Design Optimization – MBDO) ou optimisation assistée par modèles de substitution (en anglais : surrogate-assisted optimization). Le principe de l'optimisation basée sur méta-modèles consiste à réduire la charge de calcul des simulations lourdes, à l'aide de méta-modèles, qui bénéficient d'une évaluation rapide.

Afin de modéliser avec précision les nombreux phénomènes physiques qui gouvernent les dispositifs électromagnétiques, des codes numériques d'analyse lourds sont souvent utilisés pour simuler leur comportement. Malgré les progrès de la puissance de calcul, particulièrement au cours de la dernière décennie, la charge de calcul pour l'exécution de ces codes reste non-négligeable. Une seule évaluation d'un code éléments finis peut prendre par exemple de quelques minutes jusqu'à plusieurs heures, voire jours, en fonction du type de simulation désiré (analyse dynamique, transitoire, etc). Le but principal de l'utilisation de méta-modèles au sein d'un processus d'optimisation consiste à réduir de façon très significative le temps global du processus d'optimisation, en évitant les simulations lourdes en termes de temps de calcul.

Les modèles numériques sont confrontés au bruit numérique (par ex. pour la FEM, l'adaptation de maillage et la discrétisation FE) [NCA 96], [MES 07], ce qui affecte ou modifie la convergence de l'algorithme d'optimisation, spécialement pour les algorithmes à base de gradient. Cependant, les méta-modèles, la plupart d'interpolation, n'ont pas de bruit numérique.

Les modèles numériques complexes sont souvent non robustes ; l'analyse et même la génération du maillage de ces modèles peut échouer pour différentes configurations. Généralement, la mise au point d'un tel modèle numérique complexe est réalisée à partir d'un dispositif électromagnétique existant, afin de simuler son comportement. Le modèle de simulation peut cependant être adapté et amélioré en alignant les résultats de simulation à des mesures expérimentales du dispositif existant. Le modèle numérique va bien fournir des résultats précis pour des configurations proches de la configuration initiale du dispositif qui a servi à la construction du modèle numérique. Pour des modifications importantes de la configuration, le modèle est susceptible de fournir des résultats avec une précision réduite, ou encore pire, aucun résultat (à défaut de la simulation haute-fidélité). Lorsque cet événement se produit, l'ingénieur doit intervenir, corriger le problème en ajustant les divers paramètres et relancer la simulation. Les ressources (temps, argent, expérience, etc) requises pour le développement d'un modèle numérique entièrement paramétrable peuvent être prohibitives. Par contre, les méta-modèles ne nécessitent pas de réglage des paramètres et, une fois de plus, ils offrent la robustesse que les simulations numériques ne sont pas en mesure de fournir.

Dans la pratique industrielle, il arrive souvent que la personne qui développe un modèle d'un dispositif ou d'un système n'est pas la même qui exécute la tâche de conception optimale. En outre, les modèles de simulation peuvent être échangés entre les ingénieurs des différentes équipes au sein de la même entreprise. Cependant, tous les problèmes de compatibilité engendrés par l'utilisation de différentes plateformes, configurations matérielles différentes, différentes versions des logiciels, etc. peuvent représenter un réel problème et entraînent une perte de temps importante dans le développement de produits. En raison de leur taille réduite et de leur portabilité, les méta-modèles se présentent comme une solution pour ce genre de problèmes.

Un autre aspect est lié aux questions de protection de la propriété intellectuelle. Par exemple, dans le cas de partenariats, les modèles doivent être échangées entre les entreprises ou entre l'entreprise et diverses sociétés prestataires de services ou des bureaux d'études. En outre, les logiciels de simulation CAO / IAO commerciaux ou maison sont souvent protégés par des licences statiques ou réseau, clés ou autres moyens de prévention de l'usage illicite du logiciel. Une solution à ce problème de coopération sans divulguer d'informations confidentielles pourrait être l'utilisation de méta-modèles. De ce point de vue, le méta-modèle se comporte comme une « boîte noire », simulant les relations fonctionnelles entre les entrées et les sorties du modèle fin, sans fournir aucune indication sur la nature de ces relations.

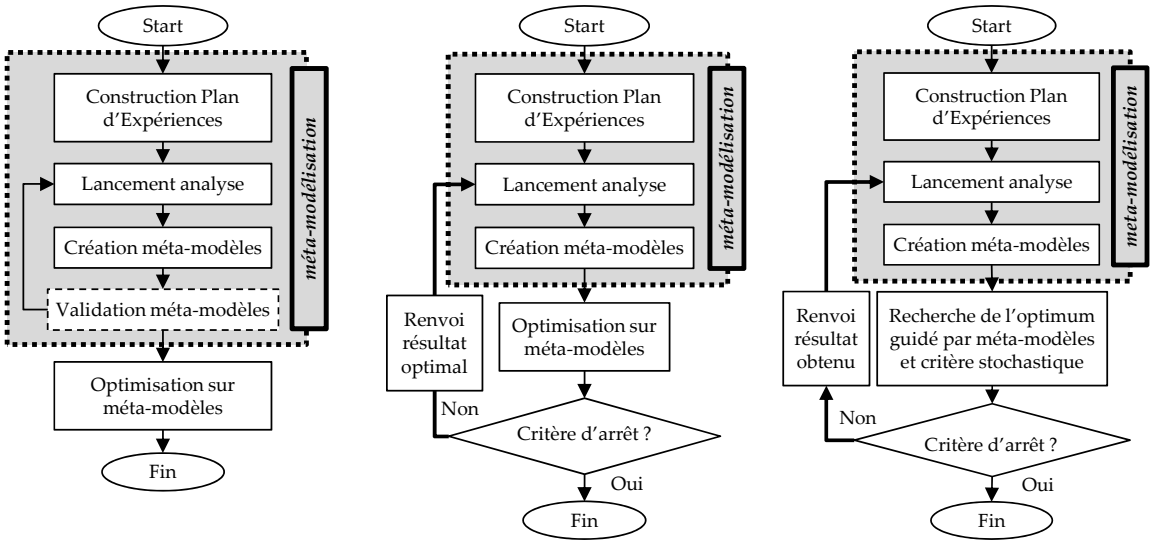
Les méta-modèles, en raison de leur évaluation rapide, permettent une exploration approfondie de l'espace de conception. Ils fournissent rapidement un aperçu des relations fonctionnelles entre les entrées et les sorties du modèle haute-fidélité, par différentes méthodes statistiques comme, l'analyse de l'influence des paramètres, l'analyse de la variance, la détection des facteurs non influents, l'exploration de l'espace de conception, etc.

Cependant, l'utilisation de méta-modèles dans un processus d'optimisation présente aussi quelques inconvénients qui doivent être mentionnés. Quel que soit le nombre de configurations qui ont servi à la construction d'un méta-modèle, la précision de ce méta-modèle sera toujours inférieure à la précision du modèle de référence ; le méta-modèle ne peut pas remplacer entièrement le modèle réel. Le méta-modèle est moins précis que le modèle réel, ce qui peut parfois

être trompeur du point de vue de l'optimisation et pourrait induire en erreur le processus d'optimisation.

Stratégies d'optimisation basées sur les méta-modèles

Dans la littérature, il y a un nombre important d'œuvres qui traitent l'intégration des méta-modèles dans le processus d'optimisation. Les travaux les plus remarquables qui traitent des stratégies d'intégration de méta-modèles dans le processus d'optimisation dans différents domaines de l'ingénierie sont recensés. Toutes ces stratégies peuvent être regroupées dans trois catégories, comme présenté dans la Figure 1.



a) approche séquentielle b) approche adaptative c) approche basée sur critère

Figure 1 : Différentes stratégies MBDO

La première approche présentée dans la Figure 1a) est une approche séquentielle, qui consiste à faire une optimisation directe sur un méta-modèle, sans reconstruction du méta-modèle au cours du processus d'optimisation.

Un méta-modèle global est d'abord construit en utilisant une stratégie de méta-modélisation, afin de représenter avec précision les simulations coûteuses. Ensuite, une étape de validation peut être ou ne pas être présent dans le processus de méta-modélisation. Une fois le méta-modèle créé, il est alors utilisé pour se substituer au code d'analyse coûteux dans une boucle d'optimisation mono-objectif ou multi-objectif, à l'aide d'un algorithme d'optimisation classique (optimiseur à base de gradient ou un algorithme évolutionnaire) pour trouver la configuration optimale.

L'avantage principal de cette approche réside dans la simplicité de son application. Le modèle fin est simplement remplacé par les fonctions objectif et contrainte du méta-modèle dans le processus d'optimisation. Toutefois, lorsque la précision des méta-modèles est médiocre, le processus d'optimisation peut échouer à trouver la solution optimale. C'est plus problématique dans le cas de problèmes contraints, où le manque de précision des fonctions contraintes du méta-modèle peut amener vers une configuration sous-optimale, ou pire, irréalisable.

La seconde approche, synthétisée dans la Figure 1b), consiste en un processus itératif d'adaptation du méta-modèle au cours du processus d'optimisation par méta-modèle.

A chaque itération de cette stratégie, les méta-modèles sont utilisés dans une boucle interne d'optimisation mono-objectif ou multi-objectif afin de rechercher une ou plusieurs conceptions optimales. Une fois ces configurations trouvées, le code de simulation couteux est exécuté et les valeurs obtenues sont utilisées pour reconstruire les méta-modèles. Les nouveaux méta-modèles intègrent bien plus d'information, ce qui se traduit par une augmentation de la précision. Afin d'assurer l'augmentation de la précision des méta-modèles tout au long des itérations, une phase supplémentaire d'amélioration de la précision globale peut être présente dans le processus d'optimisation général. La prédiction des méta-modèles devient de plus en plus précise avec les itérations, ce qui augmente la précision des solutions finales.

En dehors de sa complexité de mise en œuvre relativement modérée, cette approche a l'avantage de se plier naturellement au calcul distribué, ce qui pourrait réduire considérablement le temps global d'optimisation. Le nombre d'évaluations du modèle fin à chaque itération du processus MBDO est spécifié par le concepteur. Néanmoins, un inconvénient majeur de cette approche est la forte dépendance de son succès à la précision globale des méta-modèles. Si le problème implique un grand nombre de variables de conception, ou si le modèle fin est fortement non-linéaire, l'obtention de méta-modèles avec une précision globale élevée n'est pas possible. Le manque de précision des méta-modèles peut empêcher le processus d'optimisation de converger vers la solution ou les solutions optimales du problème.

Dans de nombreux cas pratiques, les relations fonctionnelles qui régissent le fonctionnement du dispositif à concevoir sont fortement non-linéaires. Dans ce cas, des méta-modèles globalement précis sont difficiles, voire impossibles à obtenir, exigeant un temps total de calcul rédhibitoire. **La troisième approche**, présentée dans la Figure 1c), propose d'aborder ce type de problèmes, en combinant la recherche de solutions optimales avec la phase d'exploration des méta-modèles, à travers l'expression d'un critère de sélection de nouveaux points (en anglais : infill point criterion).

Par rapport à l'approche précédente, cette stratégie ne cherche pas à atteindre la précision globale élevée des méta-modèles. En revanche, elle essaye d'obtenir des méta-modèles avec une précision élevée localement, dans les régions où se trouvent les solutions optimales. Cette approche est la plus complexe parmi les trois approches présentées, mais aussi la plus efficace. En utilisant un critère complexe, qui combine l'amélioration globale progressive de méta-modèles avec la recherche de solutions optimales, l'exploration de l'espace de conception est assurée tout en obtenant des solutions améliorées, et avec un nombre réduit d'évaluations du modèle fin.

Un inconvénient majeur de cette approche est dans le caractère séquentiel de son critère de sélection de points. C'est à dire qu'un seul point est sélectionné pour l'évaluation avec le modèle fin, à chaque itération de l'algorithme. Le processus d'optimisation nécessite des résultats valides de l'évaluation du modèle fin. C'est un inconvénient majeur parce que les modèles de simulation ne sont jamais complètement robustes. Quand l'analyse d'une configuration échoue, le processus automatique est suspendu et l'ingénieur d'études doit intervenir pour régler les paramètres du modèle de simulation et reprendre le processus d'optimisation. En outre, en raison de son caractère séquentiel, l'évaluation du modèle fin ne peut pas bénéficier des avantages du calcul distribué. Pour surmonter cet aspect, deux solutions sont proposées dans ce manuscrit afin d'adapter cette

approche au calcul distribué. L'approche est bien adaptée pour les problèmes d'optimisation avec un nombre modéré de variables de conception, inférieur à 10-15 variables.

Les algorithmes d'optimisation par méta-modèles présentent deux caractéristiques principales:

(i) Ils recherchent l'espace de conception pour les solutions optimales en utilisant les méta-modèles, phase appelée « exploitation des méta-modèles », ou simplement « exploitation » ;
(ii) Ils recherchent l'espace de conception pour des endroits prometteurs et améliorent la précision globale des méta-modèles, phase appelée « exploration de l'espace de conception », ou simplement « exploration ».

Ces deux caractéristiques se retrouvent à la base de tous les algorithmes d'optimisation à base de méta-modèles. Il peut s'agir de deux mécanismes indépendants, comme dans le cas de la seconde approche MBDO. Ou bien ces mécanismes peuvent être regroupés au sein d'un mécanisme unique, comme le critère de sélection de points de la troisième approche. Une attention spéciale est consacrée à cette dernière approche, qui contient de nombreuses contributions originales et pour laquelle un outil d'optimisation dédié a été développé.

Les approches d'optimisation à base de méta-modèles mentionnées sont présentées en détail, leurs avantages et inconvénients sont analysés en les appliquant d'une part à des problèmes analytiques mathématiques, et d'autre part à des applications physiques de dispositifs électromagnétiques.

L'approche MBDO séquentielle (pp. 61-64)

Cette approche représente la stratégie d'optimisation basée sur des méta-modèles la plus classique, et aussi la plus basique. Son principe est de créer une copie « peu couteuse » du modèle de simulation coûteux, par la construction d'un méta-modèle pour chaque objectif et contrainte de celui-là. Le méta-modèle a l'avantage d'une évaluation rapide, en échange d'une perte de précision, ce qui lui permet d'être intégré dans un processus d'optimisation.

Le processus d'optimisation employant la technique séquentielle d'optimisation à base de méta-modèles commence par la sélection d'un plan d'expériences initial. Les points de ce plan sont ensuite évalués par le modèle de simulation, afin d'obtenir les valeurs de ses sorties. Pour chaque fonction objectif et contrainte, un méta-modèle global (par ex. fonction radiale de base – RBF, méta-modèle de krigeage) est créé, basé sur les valeurs du plan d'expériences. Un algorithme d'optimisation classique (par exemple : algorithme génétique, SQP) est utilisé avec les méta-modèles créés pour trouver le ou les points optimaux du problème pour le cas mono-objectif et multi-objectif. Une fois ces points obtenus, ils sont ensuite évalués par le modèle de simulation pour obtenir les valeurs précises de ses sorties.

Afin d'illustrer les propos, un cas-test simple, consistant en une fonction mono-objectif à deux variables, est utilisé. Les résultats obtenus montrent que la fonction choisie est trompeuse. Bien que le méta-modèle a réussi à capturer la tendance globale de la fonction modélisée, sa précision n'est pas suffisante pour prédire les sorties de la fonction réelle avec précision. Ainsi, pour ce cas, le méta-modèle ne peut pas remplacer le modèle réel.

Le principal inconvénient de cette approche est bien représenté par l'imprécision dans la prédiction des méta-modèles de la vraie fonction. En outre, dans le cas de problèmes d'optimisation

avec contraintes, le manque de précision des méta-modèles des fonctions contraintes pourrait induire en erreur l'algorithme d'optimisation, soit en sélectionnant une configuration qui n'est pas optimale, ou pire, une conception qui ne respecte pas les contraintes du modèle vrai, donc irréalisable. Une solution qui pourrait remédier à cet inconvénient consiste à utiliser une stratégie de méta-modélisation adaptative. La qualité globale d'un méta-modèle est liée à la taille du plan d'expériences utilisé. Ainsi, l'ajout de plus de points à cette liste permettra d'améliorer la précision globale de sa prédiction. L'intégration d'une méta-modélisation adaptative au sein d'un processus d'optimisation est analysée dans le paragraphe suivant.

L'approche MBDO adaptative (pp. 64-78)

L'approche adaptative d'optimisation à base de méta-modèle consiste en un enchaînement de plusieurs processus d'optimisation, à l'aide d'un méta-modèle qui est systématiquement amélioré à chaque processus. Le point optimal résultant d'un processus d'optimisation est introduit dans la liste des points supports qui servent à la construction du nouveau méta-modèle pour le dispositif à optimiser. Ainsi, à travers ce processus répétitif, la prédiction des méta-modèles est progressivement améliorée, ce qui augmente les chances de trouver la configuration globalement optimale.

Le processus général de cette stratégie débute par la sélection d'un plan d'expériences initial. Le modèle de simulation est utilisé pour calculer les valeurs de sortie pour le plan d'expériences. Pour chaque fonction objectif et contrainte, un méta-modèle est construit sur la base des points évalués. Ensuite, un algorithme classique d'optimisation globale (par exemple : algorithme génétique, SQP avec multi-start) est utilisé pour trouver l'optimum des méta-modèles. Le modèle de simulation est ensuite utilisé pour valider le point optimal obtenu. Par rapport à l'approche précédente, qui se contente de cet optimum, l'approche adaptative continue en ajoutant le point optimal obtenu à la liste des points supports. De nouveaux méta-modèles sont ensuite construits en utilisant cette liste augmentée et le processus continue avec une nouvelle optimisation jusqu'à ce qu'un critère d'arrêt soit atteint (par exemple : nombre maximum d'évaluations du modèle de simulation).

Pour le cas multi-objectif, ce processus est plus complexe. Au lieu d'utiliser un algorithme d'optimisation mono-objectif pour chercher l'optimum global des méta-modèles, un algorithme multi-objectif est employé afin de trouver un front de Pareto à chaque itération du processus. Parmi la totalité de points du front de Pareto trouvé, il y a seulement quelques-uns qui sont sélectionnés pour être évalués par la suite avec le modèle de simulation. Comme pour l'approche mono-objectif, ces points sont ensuite ajoutés à la liste des points supports utilisée pour la construction de nouveaux méta-modèles. Le processus itératif est continué jusqu'à ce qu'un critère d'arrêt soit respecté.

Lorsque le problème d'optimisation est contraint, les fonctions contrainte sont méta-modélisées de la même façon que les fonctions objectif. Les contraintes sont ensuite prises en compte directement dans l'algorithme d'optimisation avec les méta-modèles, à travers leurs prédictions. Ainsi, au début du processus MBDO, lorsque seulement quelques points supports sont disponibles pour la construction des méta-modèles, et en particulier pour les fonctions de contrainte fortement non-linéaires, ce manque de précision pourrait avoir un impact important sur la forme du front de Pareto des méta-modèles. Il peut en résulter une réduction indésirable de l'espace de conception

faisable, ou une grande variété de solutions non-dominées, y compris des solutions qui satisfont les contraintes du méta-modèle mais pas ceux du modèle fin. Ceci est plus critique lorsque les points optimaux se situent à la limite entre les régions faisables et infaisables de l'espace de conception. Pour pallier cet inconvénient, une méthode de relaxation de contraintes peut être utilisée avec le processus MBDO.

Dans le cas multi-objectif, à chaque itération de la technique MBDO adaptative un ensemble de points optimaux doit être choisi au lieu d'un seul. Différentes façons de sélectionner un sous-ensemble de points de taille donnée, représentatif du front de Pareto obtenu par l'algorithme d'optimisation multi-objectif avec les méta-modèles, sont décrites.

Pour le cas bi-objectif, le choix d'un sous-ensemble de points à partir du front de Pareto des méta-modèles est intuitif et peut être fait également de manière graphique. Cela est illustré sur un problème simple, bi-objectif.

La sélection est bien plus compliquée si le front de Pareto obtenu en utilisant les méta-modèles n'est pas convexe ou concave, il présente des discontinuités, ou s'il n'a pas de forme particulière. Lorsque le problème d'optimisation comporte trois ou plusieurs fonctions objectif, la représentation graphique est moins évidente et la sélection des points devient une tâche complexe. Pour illustrer les propos, un exemple simple, à trois objectifs est considéré. Pour répondre à ce besoin, il est nécessaire de définir des critères de sélection. Trois critères de sélection d'un nombre donné de points à partir d'un ensemble de Pareto sont considérés :

- *stratégie de sélection aléatoire ;*
- *stratégie séquentielle de sélection du mieux placé ;*
- *stratégie de sélection basée sur une optimisation discrète.*

La *stratégie de sélection aléatoire* consiste à tirer au hasard un grand nombre de sous-ensembles de taille donnée à partir des points non-dominés du front de Pareto des méta-modèles. Ensuite, parmi tous ces sous-ensembles, celui qui a les points le mieux reparti, selon la distance Euclidienne entre tous ses points, est sélectionné.

La *stratégie séquentielle de sélection du mieux placé* commence par la sélection d'un point quelconque du front. Ensuite, tous les autres points souhaités sont sélectionnés un par un, de manière à être disposés de la façon la plus espacée par rapport aux points déjà sélectionnés.

La *stratégie de sélection basée sur une optimisation discrète* représente une automatisation de la stratégie de sélection aléatoire. Cette stratégie consiste à définir et résoudre un problème d'optimisation mono-objectif à variables discrètes, représentées par les index des points à choisir parmi les points du front de Pareto. La fonction objectif de ce problème est définie en fonction des distances Euclidiennes entre les points à sélectionner. Cette stratégie est la plus complexe des trois et aussi la plus difficile à mettre en œuvre.

Les avantages et les inconvénients de ces trois stratégies sont discutés et des recommandations par rapport à leur utilisation sont données.

La technique MBDO adaptative, en utilisant la stratégie de sélection de points basée sur une optimisation discrète, a été employée avec succès pour résoudre un problème de dimensionnement optimal d'un moteur linéaire à induction (en anglais : linear induction motor – LIM). Le dispositif à

concevoir de manière optimale représente un prototype à échelle réduite d'un dispositif de traction pour systèmes ferroviaires. Le LIM est constitué de deux primaires symétriques, placés face-à-face, et un secondaire consisté d'une plaque d'aluminium placée entre les deux primaires. La partie statique, représentée par la plaque d'aluminium est installé sur le sol, tandis que les parties mobiles, représentées par les deux primaires, sont installés sur le train. Les enroulements primaires sont alimentés à partir d'un système de tension triphasé symétrique, en produisant un champ magnétique mobile, ce qui induit des courants de Foucault dans la plaque d'aluminium. Le champ magnétique mobile et les courants de Foucault donnent naissance à une force de poussée qui déplace les deux primaires le long de la plaque d'aluminium.

Le LIM est représenté par une modélisation éléments finis (EF) 3D, développée au laboratoire L2EP, dans le cadre d'une thèse précédente [GON 11a]. Le modèle EF 3D du LIM est en fait un couplage entre trois modèles : deux modèles EF 3D électromagnétiques (en charge et à vide) et un modèle EF 3D thermique. Les deux modèles électromagnétiques et le modèle thermique ont été développés en utilisant un logiciel commercial éléments finis, Opera 3D, un produit de la Société Cobham [COB 12]. Le modèle du dispositif est paramétrable, c'est-à-dire que les dimensions des différents éléments du dispositif sont facilement modifiables.

La formulation du problème d'optimisation du LIM est présentée. Le but de ce problème tri-objectif est de minimiser deux fonctions objectifs, la masse du dispositif et les pertes puis de maximiser un troisième objectif, la force de poussée, en modifiant trois dimensions géométriques des primaires et la tension d'alimentation. Le processus d'optimisation MBDO est lancé à partir d'un plan d'expériences initial de 50 points et encore 100 points sont ajoutés par l'algorithme. Le mécanisme et les paramètres de cette technique sont présentés. Le front de Pareto tri-objectif est obtenu et une discussion concernant le temps de calcul pris par chaque étape de l'algorithme est présentée.

L'approche MBDO basée sur un critère statistique (pp. 79-106)

La stratégie MBDO précédente fonctionne bien lorsque les méta-modèles représentent une bonne approximation du modèle fin. Dans le cas où les fonctions contrainte et objectif présentent de fortes non-linéarités, avec beaucoup de pics et de vallées qui peuvent être très difficiles à modéliser, la stratégie MBDO précédente a de fortes chances d'échouer à trouver la ou les solutions optimales. En outre, afin de prédire avec précision les résultats du modèle fin, un grand nombre de points support bien repartis est nécessaire pour la construction des méta-modèles. La taille de l'espace de conception est directement lié au nombre de variables de conception et à leur domaine de variation. Lorsque le nombre de variables est important et leur domaine de variation est large, la taille de l'espace de conception devient énorme. Par conséquent, il est impossible d'échantillonner uniformément l'espace de conception et de créer des méta-modèles avec une bonne précision, avec un budget limité d'évaluations du modèle fin (des dizaines jusqu'à quelques centaines).

Atteindre une précision globale des méta-modèles très élevée n'est pas forcément nécessaire, ni souhaité dans cette stratégie MBDO. Au lieu de gaspiller l'effort de calcul pour échantillonner les sous-domaines non-optimaux de l'espace de conception, l'accent est mis sur les sous-domaines prometteurs, avec une forte chance de détenir les points optimaux. Par conséquent, l'idée de la stratégie présentée dans ce paragraphe est de rechercher simultanément la ou les solutions

optimales (exploiter les méta-modèles), tout en améliorant la précision de la prédiction globale des méta-modèles (explorer l'espace de conception). Ces deux fonctions peuvent être regroupées au sein de l'expression du critère de sélection de nouveaux points (en anglais : infill point selection criterion – IC). Le processus d'optimisation sera donc guidé par un critère unique pour l'ajout de nouveaux points, qui cherche à la fois à trouver les points optimaux et augmenter la précision des méta-modèles.

Cette stratégie d'optimisation est la plus complexe parmi toutes les stratégies d'optimisation à base de méta-modèles présentées dans ce chapitre et ce paragraphe contient la contribution la plus originale de l'auteur au domaine.

Critères mono-objectif de sélection de points

Une taxonomie détaillée des critères de sélection de nouveaux points pour l'optimisation mono-objectif basée sur méta-modèles est présentée par Jones et al. dans [JON 01].

Le critère le plus simple représente la prédiction des méta-modèles. Cette idée consiste à optimiser directement en utilisant la prédiction fournie par les méta-modèles. Cette stratégie de minimisation de la prédiction (« MinPred ») est équivalente à celle présentée dans le paragraphe précédent. L'objectif dans ce cas est complètement orienté sur l'exploitation des méta-modèles, en supposant un bon niveau de précision globale de ceux-ci.

Une autre idée possible consiste à trouver le point qui présente le maximum d'incertitude dans la prédiction des méta-modèles (le point avec la prédiction de l'erreur maximum), critère appelé « MaxVar ». En échantillonnant séquentiellement les points qui présentent le maximum d'erreur (variance maximale) on obtient une amélioration progressive de la précision globale des méta-modèles. Néanmoins, dans ce cas il n'y a aucun mécanisme pour diriger la recherche vers des zones susceptibles de contenir la solution optimale du problème. Ce critère se concentre exclusivement sur l'exploration de l'espace de conception, tout en négligeant la recherche de points optimaux.

Afin de montrer l'action de ces deux critères de base, MinPred et MaxVar, une fonction simple est considérée. Un méta-modèle de krigeage est utilisé pour l'optimisation avec les deux critères à partir d'un plan d'expériences initial réduit. Favorisant exclusivement l'exploitation du méta-modèle, le critère MinPred recherche localement et finit par retrouver l'un des optimums locaux de la fonction. Au contraire, le critère MaxVar effectue une exploration de l'espace de conception, améliorant globalement la qualité du méta-modèle de krigeage.

Les deux critères, MinPred et MaxVar, présentent leurs avantages et leurs inconvénients, et représentent le deux cas extrêmes de l'équilibre entre l'exploitation et l'exploration du méta-modèle. Basé sur la combinaison de ces deux critères, un certain nombre de critères plus complexes, avec des mécanismes intégrés pour équilibrer l'exploration / l'exploitation ont été proposés par Sasena [SAS 02b]. Parmi ces critères, les plus populaires sont recensés ici :

- *la limite inférieure de confiance (en anglais : lower confidence bound – LCB) ;*
- *la probabilité d'amélioration (en anglais : probability of improvement – PI) ;*
- *l'amélioration attendue (en anglais : expected improvement – EI) ;*
- *l'amélioration attendue généralisée (en anglais : generalized expected improvement – GEI).*

Le critère de *la limite inférieure de confiance* (*LCB*) est le plus simple parmi ces critères et il consiste à minimiser une limite statistique. Son expression est donnée dans (1).

$$LCB = LCB(\mathbf{x}) = \hat{y}(\mathbf{x}) - \alpha\hat{s}(\mathbf{x}) \tag{1}$$

où $\mathbf{x}$ représente le vecteur de variables de conception, $\hat{y}$ représente la prédiction du méta-modèle, $\hat{s}$ représente l'estimation de l'erreur de prédiction du méta-modèle et α est une constante qui contrôle la balance entre l'exploration et l'exploitation du méta-modèle.

Du point de vue pratique, l'utilisation de ce critère reste marginale, due au choix du paramètre α qui nécessite un certain degré de connaissance du comportement de la fonction objectif.

Le critère de *la probabilité d'amélioration (PI)* représente la probabilité que la meilleure valeur obtenue pour la fonction objectif soit améliorée par l'échantillonnage en un certain point. Son expression est donnée dans (2).

$$PI = P[I(\mathbf{x})] = \Phi\left(\frac{I-\hat{y}}{\hat{s}}\right) = \frac{1}{\hat{s}\sqrt{2\pi}}\int_{-\infty}^{0} e^{-[I-\hat{y}]^2/(2\hat{s}^2)}dI \tag{2}$$

avec $I = I(\mathbf{x}) = \max\{f_{min} - \hat{y}(\mathbf{x}), 0\}$

où I représente l'amélioration, f_{min} est la meilleure valeur de la fonction objectif à un moment donné et Φ représente la fonction de distribution cumulative normale.

La probabilité de trouver une meilleure solution est plus forte dans la région contenant la meilleure solution actuelle et ayant moins d'incertitude. Toutefois, le critère *PI* quantifie la probabilité d'améliorer la meilleure valeur connue de la fonction objectif d'une certaine valeur, mais il ne donne aucune indication sur le montant de cette amélioration.

Le critère de *l'amélioration attendue (EI)* quantifie l'amélioration susceptible d'être atteinte en échantillonnant en un certain point. Ce critère est une extension de la probabilité d'amélioration, en proposant une quantification de l'amélioration estimée par *PI*. L'expression d'*EI* est donnée en (3).

$$EI = E[I(\mathbf{x})] = \begin{cases} (f_{min} - \hat{y})\Phi\left(\frac{f_{min}-\hat{y}}{\hat{s}}\right) + \hat{s}\phi\left(\frac{f_{min}-\hat{y}}{\hat{s}}\right) & \text{if } \hat{s} > 0 \\ 0 & \text{if } \hat{s} = 0 \end{cases} \tag{3}$$

où ϕ représente la fonction normale de densité de probabilité.

Dans l'expression d'*EI*, se distinguent les deux termes correspondant respectivement à l'exploitation des méta-modèles (premier terme) et à l'exploration de l'espace de conception (deuxième terme).

Lorsque la valeur de l'erreur prédite, $\hat{s}$ est nulle (point déjà échantillonné), l'*IE* devient nul, ce qui signifie que pour ce point il ne s'attend pas à une amélioration. Si l'erreur prédite est positive, mais faible, et la valeur prédite de la fonction $\hat{y}$ est très faible en comparaison avec la meilleure valeur connue pour la fonction f_{min}, le premier terme de l'expression (3) devient prédominant. Ainsi, la recherche est effectuée localement, en exploitant la bonne précision de la prédiction des méta-modèles. Sinon, si l'erreur prédite est importante, le deuxième terme de (3) prend le contrôle et cherche à explorer les endroits de l'espace de conception où le méta-modèle est fortement imprécis.

Le critère de l'amélioration attendue a été utilisé pour la première fois par Schonlau [SCH 97] en 1997. Un algorithme d'optimisation basé sur le critère de l'*EI*, appelé Optimisation Globale Efficace (en anglais : Efficient Global Optimization – EGO), a été proposé par Jones et al. [JON 98] en 1998. L'algorithme EGO de Jones se présente comme référence dans le domaine de la conception optimale basée sur méta-modèles et a servi de base de recherche pour de nombreux chercheurs.

Le critère de *l'amélioration attendue généralisée (GEI)* représente une extension du critère EI au cas général, intégrant un mécanisme de contrôle de la balance exploitation/exploration. Son expression est donnée dans (4).

$$GEI = E[I^g(\mathbf{x})] = \hat{s}^g \sum_{k=0}^{g} (-1)^k \left(\frac{g!}{k!\,(g-k)!}\right) u^{g-k} T_k$$

avec $u = u(\mathbf{x}) = \dfrac{f_{min} - \hat{y}(\mathbf{x})}{\hat{s}(\mathbf{x})}$

$$T_k = -\phi(u)u^{k-1} + (k-1)T_{k-2}$$

$$T_0 = \Phi(u)$$

$$T_1 = -\phi(u)$$

$$\hat{s} = \hat{s}(\mathbf{x}) = \sqrt{\widehat{MSE}(\mathbf{x})}$$

$$g = 0,1,2,\cdots,n \tag{4}$$

L'équilibre entre l'exploration et l'exploitation est contrôlé par un seul paramètre, g. Avec l'augmentation de la valeur de g, l'erreur de prédiction, $\hat{s}$ devient plus importante, l'accent étant mis davantage sur l'exploration et l'amélioration de la précision globale du méta-modèle.

Parmi ces critères, la probabilité d'amélioration (PI) et l'amélioration attendue (EI) ont montré une grande popularité. Tous ces critères de sélection de points présentés disposent d'un levier pour modifier l'équilibre entre l'exploitation et l'exploration des méta-modèles. L'inclinaison de la balance en faveur de l'un ou l'autre peut être faite de différentes façons. La stratégie de l'alternance (en anglais : *switching strategy*) propose d'alterner entre l'exploitation et l'exploration des méta-modèles, en passant de MaxVar au MinPred toutes les *k* itérations. Une autre stratégie, dite de la pondération (en anglais : *weighting strategy*) est basée sur le critère de l'amélioration attendue, en ajoutant des pondérations aux deux termes dans l'expression du *EI*. Cette stratégie propose de varier les valeurs de ses coefficients, afin de modifier le ratio entre l'exploration et l'exploitation des méta-modèles. Deux autres stratégies, la pondération adaptative de l'amélioration attendue (en anglais : *adaptive weighted expected improvement – AWEI*) et l'amélioration attendue pondérée à l'aide des méta-modèles (en anglais : *surrogate model based weighted expected improvement - SMWEI*) sont citées.

Un problème important avec l'algorithme EGO est représenté par la **gestion de contraintes** [SAS 02]. Initialement, l'algorithme EGO a été conçu pour résoudre des problèmes d'optimisation sans contrainte [SAS 01] et a ensuite été adapté pour traiter aussi des problèmes contraints.

Les fonctions de contraintes sont de deux types :

- contraintes coûteuses, calculées sur la base des résultats du modèle de simulation ;
- contraintes rapides ou contraintes pas chères, calculées en utilisant uniquement les variables de conception, à travers des expressions analytiques simples.

Les contraintes coûteuses nécessitent l'évaluation du modèle de simulation ou modèle fin afin de calculer leur valeur, exactement comme dans le cas de la fonction objectif. Ainsi, les contraintes coûteuses peuvent être traitées d'une manière similaire à la fonction objectif.

Les contraintes rapides sont pour la plupart des contraintes géométriques ou topologiques, qui sont imposées afin d'éviter d'obtenir des configurations aberrantes, physiquement irréalisables. Les valeurs de ces contraintes sont ainsi connues sans aucun effort de calcul ou un effort réduit. Leur prise en compte peut être faite directement par l'algorithme d'optimisation, dans la boucle interne.

Compte tenu de leurs natures différentes, les deux types de contraintes peuvent être traités de manière différente dans le processus d'optimisation. Plusieurs approches pour gérer à la fois les contraintes coûteuses et les contraintes rapides existent, chacune avec ses avantages et ses inconvénients. Une analyse d'un certain nombre de techniques de gestion de contraintes, spécifiques à l'algorithme EGO peuvent être trouvées dans [SCH 97]. Les techniques les plus représentatives pour la gestion de contraintes couteuses dans EGO sont présentées :

- la *fonction de pénalité* (en anglais : penalty function) ;
- la *probabilité de faisabilité* (en anglais : probability of feasibility) ;
- la *violation attendue* (en anglais : expected violation).

La technique de la *fonction de pénalité* est une façon basique de gérer les contraintes et consiste à pénaliser le critère de sélection de points à chaque fois que les fonctions de contraintes coûteuses ne sont pas respectés [SAS 00]. Une constante d'une valeur négative importante est ajoutée à l'amélioration attendue afin de l'empêcher de trouver des points infaisables. Cependant, un problème inhérent à cette approche consiste à sélectionner une valeur appropriée pour la pénalité. Pour pallier à cet inconvénient, une version simplifiée de cette procédure [SOB 04] consiste tout simplement à mettre l'*EI* à zéro au lieu de lui ajouter une pénalité.

La technique de la *probabilité de faisabilité* a été proposée par Schonlau dans [SCH 97]. Le procédé consiste à multiplier le critère de l'amélioration attendue par la probabilité que chaque fonction de contrainte soit respectée. Cette méthode est appelée par Forrester et al. [FOR 08] amélioration attendue contrainte (en anglais : constrained expected improvement – CEI). La méthode dirige progressivement le critère de sélection de points vers zéro dans la zone de transition entre les régions faisable et infaisable, lissant ainsi la forme du critère dans ce domaine [PAR 10].

La méthode de la *violation attendue* est une autre technique pour gérer les contraintes au sein de l'algorithme EGO, proposé dans [AUD 00]. Similaire au critère de l'amélioration attendue, Audet et al. ont proposé de quantifier le dépassement de contraintes, qu'ils appellent *violation attendue*.

Ces techniques de gestion de contraintes peuvent être employées non seulement avec le critère de l'amélioration attendue, mais aussi avec d'autres critères de sélection de points. Une discussion des avantages et inconvénients de ces techniques de gestion de contraintes, ainsi que des recommandations de leur utilisation, a été portée sur la base d'un exemple mathématique simple.

Pour analyser le comportement de l'algorithme EGO et évaluer sa performance, un problème électromagnétique classique a été considéré. Le dispositif à concevoir de manière optimale est un système de stockage d'énergie magnétique par anneaux supraconducteurs (en anglais : Superconducting Magnetic Energy Storage – SMES), et le problème d'optimisation de ce dispositif est connu par la communauté électromagnétique comme le problème TEAM22 [MAG 08].

L'algorithme EGO multi-objectif (algorithme MEGO)

Dans la pratique, la plupart des problèmes réels d'ingénierie présentent plusieurs critères d'optimisation qui doivent être pris en compte dans le processus de conception optimale. Pour cela, un critère de sélection de points multi-objectif est requis. En conséquence, le nombre d'évaluations des modèles de simulation requis par un algorithme d'optimisation multi-objectif afin de construire un front de Pareto est beaucoup plus important que dans le cas mono-objectif. Quelques extensions de l'algorithme EGO classique au cas multi-objectif sont présentées :

- le *critère multi-objectif de l'amélioration attendue* (en anglais : multi-objective EI) ;
- le *critère multi-objectif augmenté de la probabilité d'amélioration attendue* (en anglais : enhanced multi-objective PI) ;
- l'*algorithme ParEGO* ;
- l'*algorithme Multi-EGO* ;
- l'*algorithme avec transformation multi-mono objectif en une étape* (en anglais : scalarizing one-stage algorithm).

Le *critère multi-objectif de l'amélioration attendue* représente une version multi-objectif du critère de l'EI et a été proposé par Keane dans [KEA 06]. Ce critère a été développé pour le cas bi-objectif. Tout d'abord, une fonction de densité de probabilité bidimensionnelle est calculée. Au lieu de calculer la probabilité d'amélioration de chaque fonction objectif individuellement, la probabilité d'amélioration du front de Pareto courant est calculée. L'amélioration attendue du front de Pareto est obtenue ensuite en multipliant cette valeur avec la distance entre le centre d'EI et le front. Cependant, le calcul de ce critère est une tâche exigeante en ressources informatiques. En outre, la généralisation de ce critère à plus de deux fonctions objectif n'est pas du tout une tâche triviale.

Le *critère multi-objectif augmenté de la probabilité d'amélioration attendue* a été proposé par Hawe et Sykulsky dans [HAW 07a]. Ce critère a été introduit comme un concept pour le cas bi-objectif, consistant en la division graphique de la région en dessous du front de Pareto courant en différentes zones de probabilité d'améliorations différentes. La probabilité d'amélioration du front de Pareto est ainsi donnée par l'addition de toutes ces probabilités. L'extension de cette technique à des problèmes d'optimisation impliquant plus de deux fonctions objectifs est moins évidente.

L'*algorithme ParEGO* représente une tentative plus simple de rendre l'algorithme EGO multi-objectif, présentée par Knowles dans [KNO 05a]. Afin de déterminer la frontière de Pareto d'un problème d'optimisation multi-objectif, une technique de transformation par pondération à l'aide de la formulation de Tchebyshev augmentée est employée. Ensuite, le critère EI mono-objectif classique est utilisé, non pas avec les fonctions objectif, mais avec la fonction de Tchebyshev augmentée, pour la recherche de nouveaux points. A chaque itération, les coefficients de pondération sont changés, afin de donner différents poids à chaque fonction objectif.

L'*algorithme Multi-EGO* est une autre extension de l'algorithme EGO au cas multi-objectif, proposée par Jeong et Obayashi dans [JEO 05]. Au lieu d'agréger les améliorations attendues de chaque fonction objectif dans un seul critère, ils proposent d'utiliser les EI des fonctions objectif directement au sein d'une optimisation multi-objectif. Un algorithme évolutionniste multi-objectif (MOEA) est utilisé pour déterminer le front de Pareto des améliorations attendues de toutes les fonctions objectif. Le front de Pareto obtenu est composé d'un grand nombre de points. A partir de ce front de Pareto des EI, un ensemble réduit de solutions contenant les points avec la meilleure valeur d'EI pour une fonction objectif est sélectionné. En outre, pour une meilleure convergence, le point du milieu du front de Pareto est également sélectionné.

L'*algorithme avec transformation multi-mono objectif en une étape* a été proposé par Hawe et Sykulski dans [HAW 08]. Ils utilisent une technique de transformation pour rendre le problème mono-objectif. Les valeurs des fonctions objectif sont normées et une formulation de Tchebyshev augmenté est employée afin d'agréger toutes les fonctions en une seule. Une fois le problème multi-objectif transformé en problème mono-objectif, le critère basé sur la probabilité conditionnelle (en anglais : conditional likelihood criterion) [JON 01] est maximisé en tenant compte des fonctions de contraintes coûteuses.

Le critère de sélection de points de la pseudo-distance

Le critère multi-objectif de la pseudo-distance (PD), initialement proposé par Kreuawan dans [KRE 08], a été intégré dans l'algorithme MEGO (EGO multi-objectif) développé par Berbecea et al. [BER 10]. Ce critère de sélection de points est basé sur le concept de la non-dominance qui se trouve à la base de l'algorithme génétique multi-objectif NSGA-II [DEB 02].

Pour qu'un front de Pareto représente un bon compromis entre les fonctions objectif qui le définissent, deux conditions doivent être remplies, exprimées comme suit:

(i) les points non dominés qui composent le front de Pareto doivent être situés le plus près possible du point utopie du front de Pareto;
(ii) les points du front de Pareto doivent être espacés aussi uniformément que possible.

Ces deux caractéristiques d'un bon front de Pareto se trouvent à la base du mécanisme d'avancement du front de Pareto dans le critère de la pseudo-distance. L'expression mathématique du critère de sélection de la pseudo-distance à maximiser est donnée en (5).

$$\underset{\mathbf{x}}{\text{Maximise}} \quad PD = PD(\mathbf{x}) = D_d(\mathbf{x}) + D_n(\mathbf{x}) \tag{5}$$

$$D_d(\mathbf{x}) = \sum_{i=1}^{m} D_d^{f_i}(\mathbf{x}) = \sum_{i=1}^{m} \sum_{j=1}^{n_{dom}} \left(\left(\frac{f_i^{(s_j)} - \hat{f}_i(\mathbf{x})}{f_{i\_max} - f_{i\_min}} \right) \cdot \frac{1}{\hat{s}_i(\mathbf{x})} \right) \tag{6}$$

$$D_n(\mathbf{x}) = \sum_{i=1}^{m} D_n^{f_i}(\mathbf{x}) = \sum_{i=1}^{m} \left(\left(\frac{f_i^{(s_+)} - \hat{f}_i(\mathbf{x})}{f_{i\_max} - f_{i\_min}} \right) \cdot \hat{s}_i(\mathbf{x}) \right) \tag{7}$$

où m représente le nombre d'objectifs, n_{dom} est le nombre de points sur le front de Pareto courant qui sont dominées par le point testé $\mathbf{x}$, f_{i_max} et f_{i_min} sont respectivement le maximum et le minimum connu de la i-ème fonction objectif sur le front de Pareto courant, $\hat{f}_i$ et $\hat{s}_i$ sont

respectivement les prédictions des méta-modèles de krigeage pour la i-ème fonction objectif et l'erreur estimée associée à cette prédiction, $f_i^{(s_j)}$ est la valeur de la i-ème fonction objectif du j-ème point du front de Pareto actuel, qui est dominé par le point testé $\mathbf{x}$, $f_i^{(s+)}$ est la valeur de la i-ème fonction objectif du point du front de Pareto courant le plus proche du point testé.

Le problème de sélection de points consiste à maximiser le critère de la pseudo-distance exprimé en (5) - (7). Le critère de sélection de la pseudo-distance est composé de deux termes.

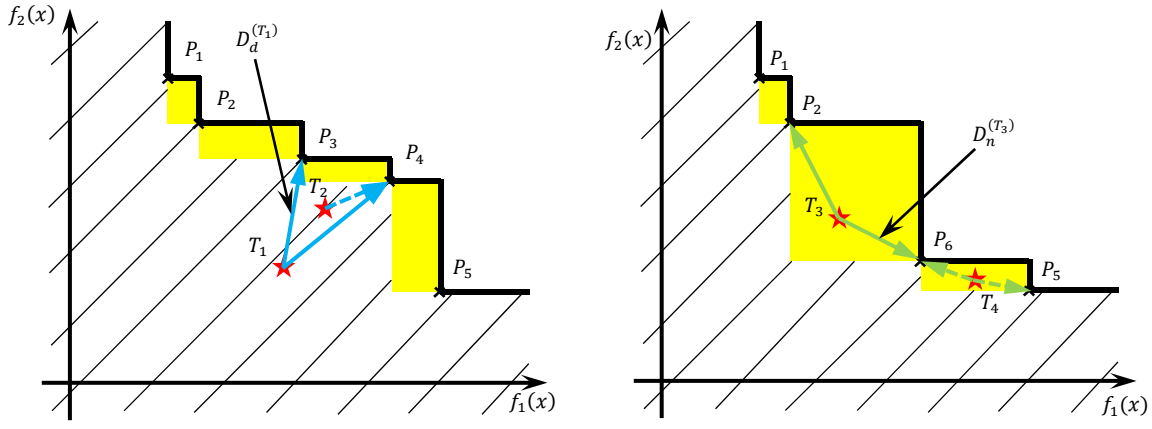
Le premier terme, $D_d(\mathbf{x})$, exprimé en (6) est appelée « distance de dominance » et il met l'accent sur les points qui dominent un ou plusieurs points du front de Pareto courant. Ce terme est destiné à répondre à la première caractéristique d'un bon front de Pareto, (i). Il est responsable de l'avancement du front de Pareto vers le point utopique.

Le second terme, $D_n(\mathbf{x})$, exprimé en (7), est nommé « distance de voisinage », abordant les points qui ne dominent pas l'un des points existants du front de Pareto courant, mais qui permettrait d'accroître la taille du front de Pareto. Ce terme répond donc à la deuxième caractéristique d'un bon front de Pareto, (ii), en essayant d'améliorer l'espacement entre les points du front de Pareto.

Dans l'expression du critère de la pseudo-distance, l'incertitude de la prédiction des méta-modèles se retrouve dans les deux termes du critère. En fonction du $\hat{s}$ deux cas se présentent :

- $\hat{s} \to 0$, la prédiction des méta-modèles présente une grande précision. Dans ce cas, $D_n(\mathbf{x}) \to 0$ et $D_d(\mathbf{x})$ devient important. Par conséquent, la maximisation du critère de la pseudo-distance devient équivalent à maximiser la distance de dominance, $D_d(\mathbf{x})$;
- $\hat{s} \to \infty$, la prédiction des méta-modèles pour ces points est très imprécise. Dans ce cas, $D_d(\mathbf{x}) \to 0$ et $D_n(\mathbf{x})$ devient dominant. Pour ce cas, maximiser la pseudo-distance se réduit à maximiser la distance de voisinage, $D_n(\mathbf{x})$.

Afin d'illustrer le comportement du critère de la pseudo-distance, un exemple d'un problème d'optimisation bi-objectif est considéré en Figure 2.



c) Itération i d) Itération i+1

Figure 2 : Exemple d'optimisation bi-objectif avec le critère de la pseudo-distance

Supposons qu'à la i-ème itération du processus d'optimisation avec le critère de sélection de la pseudo-distance, le front de Pareto est composé de cinq points, marqués P_1, P_2, P_3, P_4, P_5 dans la

Figure 2a). Les deux points de test T_1 et T_2 de la zone hachurée dominent un ou plusieurs points du front de Pareto courant. Parmi les deux, le point ayant la plus grande distance de dominance par rapport au front de Pareto courant est sélectionné, donc T_1. Ce point sera ensuite évalué en utilisant le modèle fin, ce qui donne un nouveau point sur le front de Pareto, P_6, et élimine deux points non-dominés existants, P_3 et P_4. A l'itération suivante du processus d'optimisation, le nouveau front de Pareto est celui donné dans la Figure 2b, composé de points P_1, P_2, P_6, P_5. Considérant deux autres points d'essai, T_3 et T_4, qui dans ce cas ne dominent aucun des points du front de Pareto, car ils appartiennent à la zone colorée en jaune dans la Figure 2b, la sélection sera faite au regard de la valeur de leur distance de voisinage. Par conséquent, le point T_3 sera sélectionné sur T_4.

L'algorithme EGO multi-objectif (MEGO)

Un algorithme multi-objectif d'optimisation globale efficace (MEGO) intégrant le critère de sélection de la pseudo-distance a été développé [BER 10]. L'organigramme de l'algorithme MEGO est présenté dans la Figure 3.

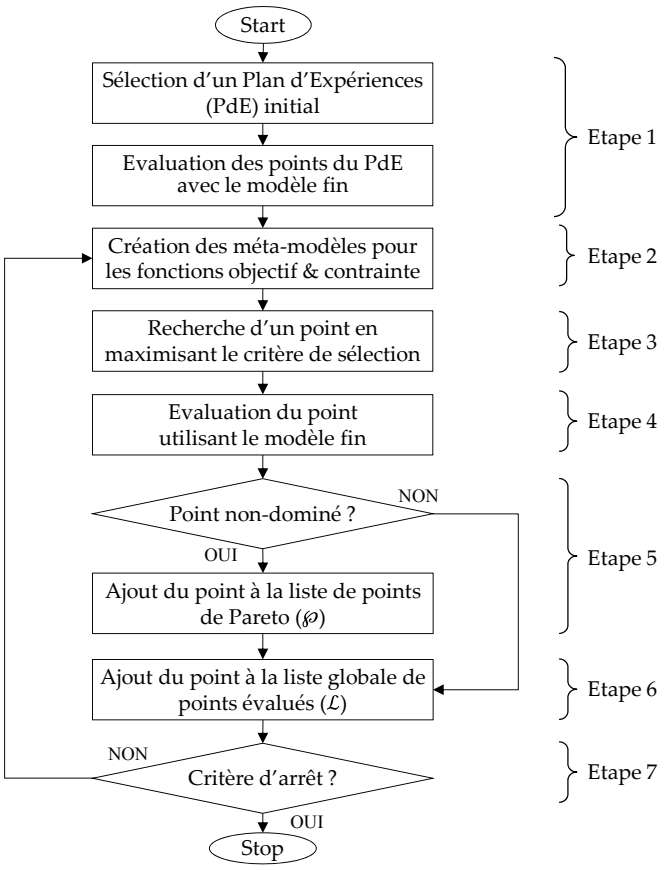


Figure 3 : L'organigramme de l'algorithme MEGO

Le processus d'optimisation en utilisant l'algorithme MEGO suit les axes suivants:

- Etape 1 : Un plan d'expériences initial est sélectionné en utilisant par exemple une technique de carré latin (en anglais : Latin Hypercube Sampling – LHS) et les points sont évalués en utilisant le modèle fin (par exemple le modèle de simulation);

- Etape 2 : Des méta-modèles de krigeage sont créés pour les fonctions objectif et contraintes, basé sur la liste globale de points évalués $\mathcal{L}$;
- Etape 3 : Un nouveau point est recherché en maximisant le critère de la pseudo-distance ;
- Etape 4 : Le point trouvé est évalué en utilisant le modèle fin ;
- Etape 5 : Le front de Pareto est recalculé. Si le point testé est non-dominé, il fait donc partie du nouveau front de Pareto $\wp$;
- Etape 6 : Le point est ajouté à la liste globale de points évalués ;
- Etape 7 : Le critère d'arrêt (par exemple : nombre maximum d'itérations) est vérifié. Si le critère est respecté, l'algorithme s'arrête ; sinon, l'algorithme continue avec l'étape 2.

L'algorithme MEGO avec distribution des calculs

Le modèle de simulation (par exemple, analyse éléments finis) nécessite généralement un temps de calcul important. L'idée de distribuer le calcul du modèle fin sur plusieurs ordinateurs, ou sur plusieurs noyaux de la même machine, fait ressortir un gain potentiel important de temps de calcul. Pour que cela soit possible, l'algorithme doit être capable de générer à chaque itération non seulement un point, mais plutôt un ensemble de points. Une fois générés, ces points sont ensuite évalués simultanément en utilisant le modèle fin, chacun sur un cœur différent de la machine. Le temps total des évaluations du modèle fin à une itération serait réduit à la plus lente des évaluations du modèle. Idéalement, le gain de temps serait directement proportionnel au nombre de cœurs qui effectuent les évaluations du modèle pour différentes configurations.

Deux stratégies différentes pour générer un ensemble de points à chaque itération ont été mises en œuvre et testées sur un serveur qui dispose de 2 processeurs avec 4 cœurs chacun :

- L'*approche par modèle hybride* ;
- L'*approche par pondération du critère de la pseudo-distance*.

L'*approche par modèle hybride* a été initialement proposée par Schonlau [SCH 97] dans un contexte d'optimisation mono-objectif basé sur l'amélioration attendue. L'idée consiste à utiliser la prédiction du méta-modèle de la fonction objectif afin de générer un ensemble de points à chaque itération de l'algorithme. Cette approche a été nommée ici « approche par modèle hybride » pour refléter l'évaluation hybride de points (la liste des points échantillonnés $\mathcal{L}$ contient un mélange des points évalués par les deux modèles, fine et méta-modèle de krigeage). La génération de nouveaux points reste séquentielle. Toutefois, cela n'a pas un fort impact sur le temps de calcul, comme la phase de recherche des points nécessite un temps négligeable par rapport à l'évaluation du point avec le modèle fin.

L'*approche par pondération du critère de la pseudo-distance* propose d'introduire une pondération contrôlée dans l'expression du critère multi-objectif de sélection de points de la pseudo-distance, qui établirait un équilibre entre les différentes fonctions objectif du problème multi-objectif. La pondération variable entre les fonctions objectif du problème multi-objectif vise à déterminer progressivement l'ensemble du front de Pareto. La recherche de points en utilisant différents coefficients de pondération peut être également distribuée, mais le gain de temps de calcul de cette étape serait toutefois négligeable par rapport à la phase d'évaluation du modèle fin.

Afin de faciliter la formulation du problème d'optimisation à l'utilisateur, ainsi que le paramétrage des différents modules de l'algorithme MEGO, une interface graphique (GUI) a été développée sous Matlab®. Les différents modules de cette interface sont présentés en détail.

L'algorithme MEGO développé a été couplé à une plateforme d'optimisation commerciale, ModeFRONTIER®. Le choix de cette plateforme a été strictement basé sur la disponibilité d'une licence pour ce logiciel au sein du laboratoire. Les avantages du couplage de l'algorithme MEGO développé avec un logiciel commercial d'optimisation sont multiples. Ce genre de logiciel contient une large palette d'outils d'analyse statistique (par exemple analyse de la variance, matrices de corrélation, matrices de nuage de points, boîte à moustaches, etc.), destinés à aider le concepteur avec l'analyse et la formulation du problème d'optimisation. La méthodologie de surface de réponse (RSM) est également disponible pour créer des méta-modèles polynômiales, RBF ou de krigeage. Des outils de visualisation et d'aide à la décision (par exemple, graphiques à bulles 3D et 4D, des tableaux d'historique, des représentations avec coordonnées parallèles, etc.) sont destinés à aider le concepteur dans le processus de prise de décision.

Application : l'optimisation du dispositif SMES

Afin d'évaluer la performance des différentes stratégies d'optimisation mono-objectif et multi-objectif à base de méta-modèles (MBDO) développées, un cas-test classique électromagnétique a été considéré. Le dispositif à concevoir de manière optimale représente un système de stockage d'énergie magnétique par anneaux supraconducteurs (en anglais : Superconducting Magnetic Energy Storage – SMES) [ALO 08], [MAG 08]. Le dispositif SMES se compose de deux bobines concentriques, composées de fils supraconducteurs. Les bobinages sont alimentés en énergie à partir du réseau avec des courants de sens opposés. Ensuite, les bobines sont court-circuitées, l'énergie étant stockée dans les champs magnétiques de la bobine. L'énergie est libérée en cas de besoin en reconnectant les bobines sur le réseau.

Le but du problème d'optimisation est de trouver les configurations (paramètres géométriques et électriques des bobines) qui donnent une capacité de stockage de 50 kWh (180 MJ) du dispositif et une valeur minimale pour le champ de dispersion à une distance donnée du dispositif. Ce cas-test d'optimisation, connu sous le nom de « problème TEAM22 » est présenté en détail. Une modélisation par éléments finis 2D a été choisie pour représenter ce dispositif. Les deux formulations pour ce cas-test, mono-objectif à 3 variables de conception et bi-objectif à 8 variables de conception, sont présentées.

Pour la résolution du problème d'optimisation avec la formulation mono-objectif à trois variables, les trois stratégies MBDO (séquentielle, adaptative et basée sur un critère statistique) ont été utilisées. Plusieurs critères (probabilité d'amélioration, amélioration espérée, amélioration espérée généralisée) ont été testés sur ce cas-test.

Le problème d'optimisation avec la formulation bi-objectif a été traité par l'algorithme MEGO développé et l'algorithme MEGO avec les deux stratégies pour l'intégration de la distribution des calculs. Les calculs ont été faits sur un serveur à 8 cœurs, avec différentes configurations de nombre de cœurs disponibles. Le gain de temps dû à la distribution des calculs a été quantifié pour chaque configuration. Un gain de temps maximum de 4 a été obtenu pour la phase d'évaluation du modèle de simulation dans l'algorithme MEGO, pour la configuration avec 8 cœurs disponibles.

Chapitre 3 Conception optimale de systèmes complexes basée sur la décomposition

Généralement, les dispositifs conçus par les ingénieurs sont trop complexes pour être traités globalement par un ingénieur ou une équipe d'ingénieurs [ALL 08]. Parmi les exemples les plus représentatifs de tels systèmes, les avions, les automobiles et les trains sont des systèmes complexes qui ne peuvent pas être traités comme un tout unitaire pour un grand nombre de raisons.

Le processus de conception de ces systèmes complexes implique des décisions combinées à partir des disciplines techniques et non techniques : comme la mécanique, l'hydraulique, la thermique, les vibrations et l'acoustique, l'électrotechnique et l'électronique, mais aussi avec l'impact environnemental, l'impact social et économique, et jusqu'à au financement, les ressources humaines, le marketing et la gestion. Exprimer tous les besoins et les exigences des domaines cités ci-dessus d'une manière générale et globale est peu pratique, difficile et en général cela ne se réalise jamais en pratique.

Le système à concevoir comporte généralement différents sous-systèmes physiques, comme la caisse, la transmission, le système de traction, le système de chauffage et de ventilation, les systèmes auxiliaires, etc. En outre, chacun de ces sous-systèmes peuvent être encore considérées comme étant composé de plusieurs composants différents, par exemple un système de traction de tramway se compose du moteur, convertisseur, transformateur, boîte de vitesse, résistance de freinage, etc. En fonction de l'organigramme de la société, en général, la conception de chaque sous-système est gérée par un autre département de l'entreprise et la conception de chaque composant d'un sous-système donné est géré par un ingénieur ou une équipe d'ingénieurs avec des compétences dans un domaine donné. Chaque équipe d'ingénieurs possède une expertise particulière et utilise ses propres outils pour la simulation, l'analyse et la conception du produit à développer. Cependant, la tâche initiale de conception est divisée en plusieurs problèmes de conception plus petits et plus faciles à gérer, chacun affecté à une équipe différente de spécialistes, qui utilisent leurs outils appropriés. La collaboration entre tous ces acteurs au sein du cycle de production d'un dispositif est essentielle et doit être faite de manière coordonnée afin d'aboutir à une configuration compétitive du dispositif.

La décomposition des systèmes complexes contribue également à une meilleure compréhension des relations fonctionnelles entre les différents éléments de la structure décomposée du système.

En décomposant le système, le problème d'optimisation initial est divisé en plusieurs problèmes d'optimisation plus petits pour chaque élément de la structure décomposée, qui sont résolus indépendamment. Ainsi, il devient possible d'utiliser une méthode d'optimisation différente pour résoudre chaque problème d'optimisation de la hiérarchie. De cette manière, on peut utiliser l'outil d'optimisation le plus approprié pour résoudre chacun des problèmes d'optimisation des sous-systèmes. De cette façon, un des problèmes de sous-système comportant un modèle « lisse » peut être résolu en utilisant un algorithme d'optimisation à base de gradient, telle que SQP. D'autres sous-problèmes qui emploient des modèles de calcul plus légers, mais avec un caractère multimodal peuvent être résolus avec un algorithme d'optimisation heuristique, comme GA, PSO ou recuit simulé. Pour un problème d'optimisation en utilisant un code de simulation lourd, qui

présente du bruit numérique, un algorithme d'optimisation basé sur méta-modèles pourrait être employé.

Cependant, un inconvénient important des méthodes d'optimisation à base de décomposition est le coût de calcul supplémentaire lié à la stratégie de coordination, qui a pour rôle d'assurer la cohérence entre les différents sous-systèmes du système complexe [GUA 11].

Pour toutes les raisons énoncées et beaucoup d'autres, ces systèmes complexes nécessitent l'utilisation d'une méthode de décomposition pour leur conception optimale.

Partitionnement du système complexe

Le partitionnement d'un système complexe peut être fait dans un certain nombre de façons différentes, suivant différents critères. Dans la littérature [CHO 05], plusieurs catégories de stratégies de décomposition de systèmes complexes sont évoquées: décomposition à base d'objets, selon les disciplines, basée sur les modèles et, moins courante, une stratégie de décomposition séquentielle ou basée sur le *workflow*.

En utilisant la décomposition à base d'objets, le système est divisé par des composants physiques, par exemple pour un système de traction ferroviaire: le moteur, l'électronique de puissance, les systèmes auxiliaires, etc. La décomposition basée sur l'aspect ou les disciplines implique la division du système en fonction de différentes disciplines et est utilisée lorsque plusieurs aspects de la performance d'un dispositif ou d'un système sont évalués ensemble. La décomposition basée sur les modèles est basée sur les dépendances fonctionnelles entre les variables et les fonctions d'un problème. La décomposition séquentielle traite la division d'un problème en sous-problèmes à l'aide d'une organisation basée sur le *workflow*. Dans l'électrotechnique et, à une plus large échelle, le domaine de l'ingénierie électrique, la conception optimale des systèmes complexes concerne principalement les trois dernières stratégies de décomposition. La décomposition du système complexe suivant ces trois stratégies les plus fréquentes est présentée schématiquement dans la Figure 4.

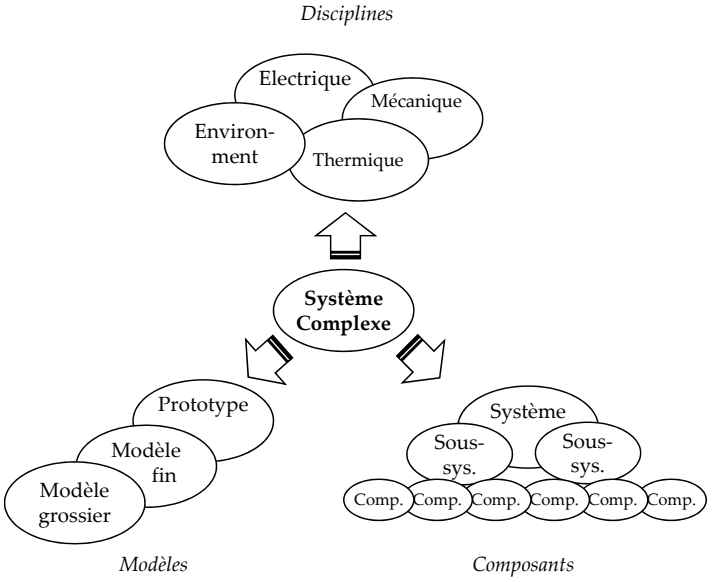


Figure 4: Décomposition du système complexe

Dans la pratique industrielle, la décomposition d'un système complexe est principalement faite conformément à l'organisation de l'entreprise qui le conçoit. Dans une entreprise, il peut y avoir plusieurs équipes d'ingénieurs, chacune spécialisée dans un domaine, tels que l'électricité, la mécanique, les structures, la thermique, etc. qui travaillent au développement d'un système complexe. Dans ce cas, les approches MDO sont bien adaptées à ce type d'organisation de l'entreprise. D'autres sociétés, comme c'est le cas d'Alstom, sont organisées hiérarchiquement et la conception d'un système complexe (métro, tramway, train, etc.) est le résultat d'un assemblage des différents éléments constitutifs du système à concevoir. Dans ce cas, les équipes d'ingénieurs sont spécialisées et affectées chacune à aborder la conception d'un composant du système complexe. D'autres départements ou équipes s'occupent de l'intégration des différents composants dans la structure du système complexe. Les approches de décomposition du système complexe à base d'objets sont bien adaptées pour traiter ce genre de structures hiérarchiques. En outre, cette méthode de décomposition est aussi motivée par l'organisation de l'entreprise sur plusieurs sites de production différents, chacun en charge de la production d'un composant du système de traction ferroviaire.

Selon le niveau de complexité du système, il y a deux types de formulations qui peuvent être utilisés pour l'optimisation d'un système complexe: formulations mono-niveau et formulations multi-niveaux. La différence entre les deux approches réside dans la façon dont la tâche d'optimisation est attribuée et exécutée. Pour les méthodes mono-niveau, un optimiseur unique gère la tâche d'optimisation de la structure décomposée du système. Lors de l'utilisation des méthodes multi-niveaux, la tâche d'optimisation est distribuée dans tout le système, en utilisant un optimiseur différent pour chaque élément de la structure décomposée. De nombreuses stratégies existent pour coordonner les différents optimiseurs engagés dans l'optimisation d'un système complexe. Bien que les méthodes appartenant à la première catégorie ont été étudiées pendant longtemps et ont déjà atteint la maturité, les dernières approches ne sont que dans leur période de genèse [DEP 07]; beaucoup de recherche a été consacrée récemment au développement, à l'amélioration et l'application de ces stratégies.

Les stratégies d'optimisation mono-niveau ont été beaucoup étudiées au cours d'une longue période de temps et ont déjà prouvé leurs capacités et leurs limites. Au laboratoire L2EP à l'Ecole Centrale de Lille, ces techniques ont fait l'objet de plusieurs recherches récemment [KRE 08], [BEN 12b]. La comparaison des différentes méthodes MDO mono-niveau avec application à l'optimisation d'un transformateur d'isolement de sécurité a été traitée par Ben-Ayed et al. [BEN 12]. Cependant, les stratégies MDO mono-niveau ne sont pas citées ici à des fins d'achèvement et ne seront pas étudiés ici. L'objectif principal de ce chapitre est centré sur les stratégies d'optimisation multi-niveaux, qui n'ont pas encore fait preuve de leurs capacités. A ces fins celles-ci sont analysées dans les paragraphes de ce chapitre.

Stratégies d'optimisation multi-niveaux

Les stratégies d'optimisation basées sur la décomposition multi-niveau impliquent le partitionnement du système complexe en différents éléments ou sous-systèmes, regroupées sur plusieurs niveaux, de manière hiérarchique et interconnectées. Les stratégies d'optimisation multi-niveaux utilisées avec de telles structures décomposées emploient un optimiseur différent pour chaque sous-système de la hiérarchie. Cependant, la charge de l'optimisation est distribuée le long

des différents éléments de la structure décomposée. L'attribut «hiérarchique» d'un partitionnement du système se réfère à la dépendance fonctionnelle entre les différents éléments de l'architecture. Une stratégie de coordination doit être mise en place afin de gérer le flux d'informations entre les différents éléments de la structure décomposée. Le rôle de la stratégie de coordination est de faire converger les différentes grandeurs échangées entre les éléments et obtenir à la fin du processus une conception qui est cohérente et optimale. La stratégie de coordination a également le rôle de définir l'ordre dans lequel les différentes optimisations des éléments sont exécutées et de passer l'information d'un élément à l'autre.

La décomposition du système peut être faite de différentes manières et la façon dont il est décomposé impose le choix d'une stratégie de conception optimale appropriée.

Comme mentionné précédemment, on retrouve donc des stratégies de décomposition :
- selon les modèles impliqués dans la représentation du système;
- disciplinaire, basée sur les différentes disciplines ;
- à base d'objets.

Ces trois décompositions sont les plus fréquemment rencontrées dans la pratique industrielle, en particulier les deux dernières, qui sont destinées à se plier à l'organisation structurelle de l'entreprise.

Décomposition du système complexe selon les modèles

La décomposition d'un système selon les modèles impliqués dans sa représentation implique l'existence de plusieurs modèles du système ou dispositif à concevoir, de granularité différente. En fonction du but de l'étude, le système complexe peut être représenté en utilisant différent types de modélisation.

Dans la pratique industrielle, pendant les phases préliminaires du processus de conception, il est nécessaire d'avoir une vision plus globale, une vision d'ensemble du système à concevoir. Ainsi, un modèle grossier, comme par exemple une simple surface de réponse qui donne les tendances des grandeurs du système peut suffire. Le grand avantage de ce genre de représentation est la rapidité de l'obtention des grandeurs définissant le système. L'inconvénient est le manque de précision.

Pendant les phases finales du processus de conception, des simulations de haute précision sont employés par les ingénieurs concepteurs, ayant pour but d'analyser et valider le comportement du dispositif. Au cours des deux dernières décennies, avec le développement important des ressources informatiques, l'utilisation des logiciels de conception assistée par ordinateur (CAO) est devenue très fréquente dans la conception. Ce genre d'outils, comme l'analyse par éléments finis (en anglais : Finite Element Analysis – FEA), sont souvent utilisés pour la simulation et la validation des concepts potentiellement optimaux. Une telle analyse nécessite généralement beaucoup de temps de calcul. Ce fait compromet l'évaluation de nombreuses configurations potentiellement optimales. En fonction du type d'application, la précision des résultats fournis par les simulations peut ne pas suffire, auquel cas le processus de conception peut être amené au développement des prototypes à échelle réduite. Il est évident que ce genre de prototypes sont très couteux, d'où la nécessité de limiter le développement de ces dispositifs. Des techniques telles que le mappage de l'espace (en anglais : Space Mapping – SM) sont destinées à résoudre des problèmes de conception optimale

portant sur plusieurs modèles de granularité différente du même dispositif à concevoir. Cela exploite les synergies de l'utilisation combinée des modèles de fidélité différente d'un système ou dispositif. La représentation générale d'une décomposition du système complexe suivant plusieurs modèles de précision différente est présentée schématiquement dans la Figure 5.

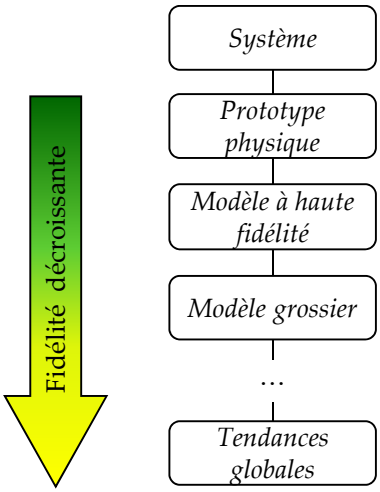


Figure 5: Différents modèles utilisés dans une décomposition à base de modèles

Ce genre de structure est organisé de manière hiérarchique, avec une fidélité décroissante des modèles à partir du haut vers le bas du schéma.

La technique SM est une méthode de projection de l'espace [BAN 94], [BAN 04]. Il s'agit d'une approche courante pour l'optimisation des systèmes représentés par plusieurs modèles de granularité différente. Cette approche nécessite l'utilisation d'au moins deux modèles d'un même dispositif, un modèle fin et un autre grossier, respectivement un modèle FE et un modèle analytique ou FE avec un maillage grossier. Les différentes optimisations de fidélités différentes sont imbriquées l'une dans l'autre dans une architecture hiérarchique. Cette méthode a été appliquée avec succès dans le domaine électromagnétique pour l'optimisation des convertisseurs électromagnétiques [BAN 94], [ENC 07]. Plusieurs versions de la technique du mappage de l'espace ont été développées. Parmi celles-ci, les plus courantes sont le mappage agressive de l'espace (en anglais : Agressive Space Mapping – ASM) [BAN 95], le mappage de l'espace sur les sorties (en anglais : Output Space Mapping – OSM) [ENC 07], [ENC 08], [TRA 09] et le mappage multiple (en anglais : Manifold Mapping – MM) [ECH 08]. La première approche mentionnée (ASM) propose une technique pour l'accélération de la convergence de cette stratégie, ce qui conduit à une réduction de l'effort de calcul. Cependant, l'ASM fait usage d'un procédé complexe d'extraction de paramètres, ce qui est très sensible et difficile à mettre en œuvre. Les deux dernières versions, l'OSM et MM, évitent le processus délicat d'extraction de paramètres et sont facilement réalisables. La technique MM utilise ce qu'on appelle une fonction de mappage multiple pour modifier les valeurs de consigne à chaque itération. La version OSM aligne de manière itérative le modèle grossier au modèle fin par le biais d'un vecteur de coefficients de correction.

La méthode d'optimisation « Output Space Mapping » (OSM) (pp. 132-136)

Dans ce paragraphe, le mécanisme de fonctionnement de la méthode d'optimisation OSM est décrit. La méthode OSM fonde son principe sur l'alignement systématique d'un modèle grossier à un modèle fin. Cette méthode comporte deux boucles d'optimisation imbriquées – une à « haute fidélité » et une à « faible fidélité », avec un lien hiérarchique. La boucle d'optimisation à « faible fidélité » utilise un processus d'optimisation classique (tel que SQP) avec le modèle grossier qui bénéficie d'une évaluation rapide. La boucle supérieure, à « haute fidélité » corrige le modèle grossier en utilisant le modèle fin. Dans la boucle supérieure, il n'y a pas de problème d'optimisation formel. Il ne s'agit ici que d'une seule étape avec une seule évaluation par le modèle fin de la configuration précédemment obtenue par la boucle inférieure. Une fois la configuration évaluée en utilisant le modèle fin, un vecteur de coefficients de correction est calculé afin d'aligner le modèle grossier au modèle fin. Le processus se poursuit jusqu'à ce que les deux modèles, fin et grossier, soient alignés à l'emplacement de l'optimum, avec une tolérance prédéfinie.

Afin d'illustrer le mécanisme de la stratégie de l'OSM, un exemple simple consistant en un problème mathématique a été construit. Les fonctions objectif et contrainte du modèle fin ont été représentées par deux polynômes d'ordre 2. Egalement, deux polynômes du $4^{ème}$ et $2^{ème}$ degré ont été construits et considérés comme des modèles grossiers pour la fonction objectif et la fonction contrainte, respectivement. L'optimum du modèle fin est obtenu en 8 itérations de l'OSM. La tendance globale des modèles fins est capturée avec succès par les modèles grossiers, ce qui représente la clé de la réussite du processus d'optimisation en utilisant la méthode OSM.

Des généralisations de cette méthode au cas d'un problème d'optimisation à trois niveaux [BEN 12a], [BEN 12b] et le cas général à « n » niveaux [BEN 11a], [GON 11a] sont citées. Néanmoins, dans la pratique, un nombre maximum de trois niveaux de modèles est à considérer pour la configuration de la stratégie OSM, pour des raisons liés à la fois au gain en temps de calcul et de difficultés d'implémentation.

En regard du caractère multi-modèle de la méthode, un nombre de similarités existe entre cette méthode et la méthode MBDO analysée dans le chapitre précédent. Alors que dans le cas présent un modèle grossier est systématiquement amélioré en l'alignant à un modèle fin, en utilisant le vecteur de coefficients de correction, les approches MBDO du chapitre précédent proposent l'utilisation d'un méta-modèle qui est progressivement améliorée par l'ajout des nouveaux points support. Dans une certaine mesure, les approches MBDO peuvent ainsi être perçues comme des stratégies d'optimisation basées sur la décomposition multi-niveaux des modèles. Cela donne droit à une comparaison entre deux approches similaires, la stratégie OSM et la stratégie MEGO développé dans le chapitre précédent. En utilisant la même application d'optimisation du transformateur d'isolement traité par Ben-Ayed et al. dans [BEN 12], une première comparaison entre l'OSM et l'algorithme MEGO a été présenté dans [BEN 11] sur la base d'une optimisation mono-objectif. Les résultats ont montré un avantage de la méthode OSM sur l'approche MEGO, ce qui est principalement attribuable à la bonne qualité du modèle analytique utilisé comme modèle grossier dans OSM. Pour une comparaison complète entre les deux approches présentées, une comparaison multi-objectif a été également réalisée pour le problème de l'optimisation du transformateur d'isolement par Berbecea et al. dans [BER 12]. Les comparaisons des deux approches, OSM et MEGO en mono-objectif et multi-objectif ayant comme application le cas-test du

transformateur d'isolation ont été réalisées et se retrouvent dans la partie applications du 3ème chapitre, paragraphe 3.7.3. Ces comparaisons ont permis d'identifier les similitudes et les différences entre les deux approches à la fois dans un cadre mono-objectif et dans un cadre multi-objectif, ainsi que l'élaboration des recommandations utiles pour les futures améliorations des deux méthodes.

Décomposition du système complexe basée sur les disciplines

Le développement de stratégies d'optimisation par décomposition disciplinaire multi-niveaux du système complexe est venu comme une solution pour repousser les limites rencontrées par les stratégies MDO mono-niveau. Lorsque le problème de conception multidisciplinaire devient très complexe, avec de nombreuses disciplines impliquées dans le processus de conception optimale présentant des connexions interdisciplinaires très fortes, comme c'est le cas de la plupart des problèmes industriels, les formulations MDO mono-niveau classiques ne peuvent pas gérer la complexité de ce genre de problèmes.

Plusieurs stratégies MDO multi-niveaux ont été développées au cours des deux dernières décennies. La plus courante est la stratégie de l'optimisation collaborative (en anglais : Collaborative Optimization – CO) proposée par Braun et Krou dans [BRA 96]. La méthode propose une plus grande autonomie des différents sous-systèmes de la structure décomposée afin de satisfaire les contraintes de compatibilité interdisciplinaires. Ainsi, avec l'autonomie disciplinaire, cette méthode permet l'intervention des spécialistes d'une certaine discipline dans le sous-système correspondant, sans aucun égard pour les autres disciplines impliquées dans le processus de conception optimale. Une autre méthode MDO multi-niveaux consiste en l'optimisation concurrentielle de sous-espaces (en anglais : Concurrent Subspace Optimization – CSSO), proposée plus tôt par Sobieszczanski-Sobieski en 1988 [SOB 88] et développée au centre de recherche Langley de la NASA. Comme avec la méthode CO précédente, la stratégie CSSO plaide également en faveur de l'autonomie disciplinaire. Un optimiseur au niveau système gère le couplage entre les disciplines et guide le processus d'optimisation vers une solution cohérente. Au sein de chaque sous-système disciplinaire, une approximation des variables de couplage est utilisée afin d'estimer leur influence sur la fonction objectif et les fonctions contraintes. Durant les optimisations simultanées des sous-systèmes, les variables partagées sont considérées comme des constantes. L'approximation des variables de couplage est obtenue grâce à l'utilisation de méta-modèles ou des modèles approximations. La synthèse bi-niveaux du système (en anglais : Bi-level System Synthesis – BLISS) est une autre stratégie MDO multi-niveaux proposée également par Sobieszczanski-Sobieski une dizaine d'années après CSSO, en 1998 [SOB 98], et a été également développée au centre de recherche Langley de la NASA. Comme pour les deux méthodes précédentes, un optimiseur système est en charge de la coordination des différents processus d'optimisation des sous-systèmes. Afin d'améliorer la satisfaction des contraintes de cohérence interdisciplinaire, le problème au niveau système emploie également une analyse multidisciplinaire pour mieux estimer les valeurs des variables disciplinaires partagés et locales. La méthode de BLISS utilise une approche basée sur le calcul du gradient pour optimiser les contributions des variables de conception partagées et celles relatives à chaque discipline sur la fonction objectif. Une version améliorée de cette méthode, nommée BLISS2000 emploie des méta-modèles pour les différentes analyses disciplinaires, améliorant ainsi l'efficacité de calcul de la méthode [SOB 00a]. Une méthode similaire pour les

problèmes MDO multi-niveaux, nommés DIVE (en anglais : Disciplinary Interaction Variable Elimination Approach) a été développée par Parte et al. dans [PAR 09]. La méthode DIVE fait usage de méta-modèles pour les différentes optimisations disciplinaires. Cette méthode propose une autonomie complète de disciplines individuelles pour gérer leurs variables disciplinaires.

Une caractéristique commune à toutes les stratégies d'optimisation multi-niveaux basées sur la décomposition disciplinaire du système complexe est qu'elles ont toutes deux niveaux. Comme il n'y a pas de priorité entre les disciplines, tous les sous-systèmes disciplinaires sont disposés sur le même plan, étant évalués en même temps. Un optimiseur au niveau système est chargé de coordonner les différents optimiseurs des sous-systèmes du niveau disciplinaire.

La représentation de plusieurs niveaux de décomposition du système complexe selon les différentes disciplines concernées est présentée schématiquement sur la Figure 6.

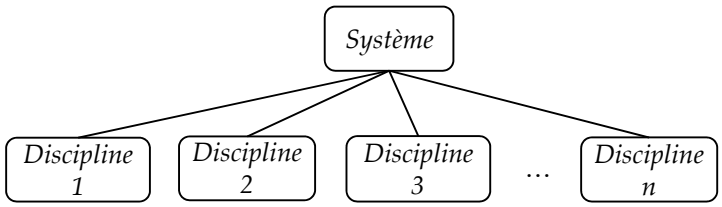


Figure 6: Schéma d'une décomposition disciplinaire

La stratégie d'optimisation « Collaborative Optimization » (CO) (pp. 136-141)

L'optimisation collaborative (en anglais : Collaborative Optimization – CO) est une classe de stratégies d'optimisation multidisciplinaire bi-niveaux, destinée à la résolution de problèmes de conception optimale de dispositifs avec une modélisation multidisciplinaire. La structure de la CO comporte deux niveaux : un niveau supérieur (niveau système) et un niveau inférieur (niveau disciplinaire). La stratégie CO a été introduite en 1996 par Braun dans [BRA 96]. L'architecture de CO favorise l'autonomie disciplinaire tout en conservant la compatibilité interdisciplinaire [BRA 96a], [KRO 00]. Les premières applications de CO ont traité des problèmes de conception optimale multidisciplinaires de l'industrie aéronautique et aérospatiale pour résoudre les applications de conception optimale des avions [BRA 96], [MOO 96], [ALL 06a], [WAN 09]. Un certain nombre de similitudes existent entre la CO et une autre approche d'optimisation basée sur la décomposition, l'ATC présentée au paragraphe 3.5. Dans [ALL 05], Allison et al. ont passé en revue la terminologie utilisée par la CO et l'ATC et ont clarifié les différences entre les deux méthodes à travers deux applications : un problème de conception optimale des structures et un problème de conception optimale d'une pompe à eau électrique.

La présentation de la stratégie d'optimisation CO commence par la représentation graphique de la structure de CO, ainsi que le flux d'informations entre les éléments de cette structure. Les différentes notations employées sont définies. La formulation mathématique de base de la stratégie CO est introduite. Deux formulations différentes sont utilisées pour les optimisations du niveau système et du niveau disciplinaire.

Formulation du problème d'optimisation du niveau système

Le rôle de la boucle d'optimisation au niveau système est de coordonner les différents problèmes d'optimisation disciplinaires situés au niveau disciplinaire inférieur. Au niveau système, aucune analyse formelle n'est lancée. L'objectif du problème d'optimisation du système est de minimiser la fonction objectif globale du problème. Ainsi, aucune contrainte locale spécifique d'inégalité ou d'égalité n'est exprimée ici. Les variables de conception du niveau système sont constituées des variables globales de conception et des variables de couplage interdisciplinaire. Les dernières représentent les variables partagées entre les différentes disciplines impliquées dans le processus de conception optimale. La cohérence entre les différentes disciplines impliquées dans le processus de conception optimale est assurée par des contraintes interdisciplinaires.

Formulation d'un problème d'optimisation du niveau disciplinaire

Au niveau inférieur de la hiérarchie multidisciplinaire de CO, un problème d'optimisation différent est défini pour chaque discipline. Chaque problème d'optimisation du niveau disciplinaire de la hiérarchie joue sur ses variables locales de conception afin de trouver un accord avec les autres disciplines sur les variables de couplage, tout en respectant les contraintes locales spécifiques à la discipline.

Coordination du processus CO

La coordination du processus de CO est réalisée par le problème d'optimisation au niveau système et par la gestion des contraintes de cohérence interdisciplinaire. A la convergence, les contraintes interdisciplinaires doivent être actives afin d'avoir une cohérence entre les différentes disciplines.

Un problème connu du processus de CO est lié à la non-robustesse liée à une instabilité à la convergence [BAL 11]. En outre, le processus de CO a montré un taux lent de convergence, voire même a divergé, en raison des contraintes de compatibilité interdisciplinaire exprimées comme des contraintes d'égalité. Le potentiel bruit dans les modèles d'analyse disciplinaire au niveau sous-système, combiné avec les contraintes d'égalité, introduisent des difficultés de calcul [ALE 00], [KRO 00], [ZAD 09].

Un certain niveau d'incohérence doit être accepté pour le problème au niveau système afin que le processus puisse converger vers un optimum. Pour cela, les contraintes d'égalité interdisciplinaires peuvent être assouplies [ALE 00], [ALL 05], par une transformation en des contraintes d'inégalité.

En utilisant une relaxation des contraintes, l'incohérence interdisciplinaire est autorisée au cours des itérations au niveau système. Pour que le processus d'optimisation puisse converger, il est nécessaire d'imposer une tolérance assez lâche. Cependant, une valeur de tolérance très lâche emmènera le processus d'optimisation vers une configuration physiquement incohérente. Ainsi, une stratégie avec une tolérance variable peut être utilisée avec CO, en commençant le processus itératif du niveau système avec une valeur plus élevée de la tolérance et en la réduisant vers la fin du processus d'optimisation, afin d'assurer la cohérence entre les disciplines.

La formulation de la stratégie CO multi-niveaux est destinée à se conformer très bien avec des entreprises qui sont organisés en plusieurs départements, chacun étant chargé d'aborder une

discipline du système à concevoir. Cette formulation permet aux différents spécialistes de la discipline ou à une équipe de spécialistes au sein de l'entreprise de travailler au sous-système de leur expertise sans prêter attention aux autres sous-systèmes disciplinaires.

Différentes techniques pour l'amélioration de l'efficacité de la formulation CO sont citées. La stratégie CO, comme la plupart des techniques d'optimisation basées sur la décomposition, donne lieu à des difficultés informatiques dans le processus de coordination. Plusieurs études ont abordé ces questions de coordination et ont tenté, par différents moyens, d'améliorer la convergence du processus de coordination. Li et al. ont identifié et utilisé certaines propriétés géométriques du processus de CO dans [LI 08].

Pour alléger la tâche du calcul précis mais lourd de l'analyse disciplinaire, Sobieski et Kroo [SOB 00], Jang et al. [JAN 05], Zadeh et al. [ZAD 09] et Wang et al. [WAN 09] ont proposé de remplacer les modèles de simulation disciplinaire par des méta-modèles dans le cadre de CO. [JAN 05] utilise une combinaison de réseaux de neurones et méta-modèles de krigeage à la fois dans un contexte d'optimisation CO mono-objectif et multi-objectif, comme substitut pour les simulations disciplinaires consommatrices de temps.

La stratégie classique de CO utilise une formulation imbriquée, ce qui signifie que, pour chaque itération de la boucle du niveau système, une optimisation complète des sous-systèmes du niveau disciplinaire est effectuée. Cela engendre un nombre important d'évaluations des modèles disciplinaires, qui sont généralement des modèles de simulation coûteux en temps de calcul. Afin de réduire le coût de calcul de CO, une méthode d'optimisation collaborative renforcée (en anglais : Enhanced Collaborative Optimization – ECO) a été proposée par Roth et al. [ROT 08], [ROT 08a]. La nécessité d'une optimisation complète de sous-systèmes à chaque itération de la boucle système est éliminée par l'emploi d'une relaxation lagrangienne augmentée pour la gestion des contraintes interdisciplinaires et l'utilisation des méta-modèles. Une amélioration de l'ECO par l'emploi des méta-modèles de krigeage a été proposée par Xiao et al. dans [XIA 10].

L'intégration de l'aspect multi-critères dans le processus de CO a été introduite par Rabeau et al. dans [RAB 07], implémenté sous leur plate-forme d'optimisation COSMOS. Cette approche, basée sur l'utilisation conjointe de la stratégie CO classique et des algorithmes évolutionnaires multi-objectif, permet de trouver un ensemble de solutions de compromis entre les différentes disciplines impliquées dans le processus d'optimisation.

Une nouvelle stratégie d'optimisation multi-niveaux, nommée BLISCO et combinant les fonctionnalités de CO traditionnelle et de BLISS a été proposée dans [ZHA 11]. Cette approche a été couplée au logiciel commercial d'optimisation iSIGHT™.

Amélioration de l'efficacité du processus CO par la technique OSM

Le processus de CO effectue une optimisation multidisciplinaire des systèmes ou des dispositifs complexes. Cependant, l'intégration de modèles de simulation précis dans le processus de CO implique une augmentation conséquente de la charge de calcul, principalement en raison du caractère imbriqué des boucles d'optimisation du processus CO. Lorsque les modèles disciplinaires employés sont des modèles avec une évaluation rapide, tels que les modèles analytiques - comme c'est le cas du transformateur d'isolement - ou modèles approximatifs, tels que les surfaces de réponse, l'optimisation peut être exécutée pour un coût de calcul raisonnable, quel que soit le

nombre d'appels aux modèles disciplinaires au sein du processus de CO. L'intégration de la précision offerte par les modèles de simulation, tels que les modèles éléments finis, dans le processus d'optimisation multidisciplinaire CO, en remplaçant simplement l'appel aux modèles rapides par un appel à ces modèles lents est peu judicieuse.

Une nouvelle idée est introduite ici, pour intégrer des modèles de simulation dans la formulation de CO. Elle est basée sur l'existence de deux niveaux de modélisation pour le même dispositif à concevoir. L'idée proposée consiste à mettre en œuvre une technique de mappage de l'espace pour résoudre les optimisations disciplinaires au niveau inférieur de la structure CO, pour le problème de conception optimale du transformateur d'isolement. La mise en œuvre de cette idée est basée sur l'existence de deux modèles distincts, un modèle FE et un modèle analytique pour chaque discipline impliquée dans la simulation du transformateur. La technique de l'OSM a montré une bonne performance pour résoudre le problème d'optimisation du transformateur d'isolement avec la formulation AIO [BEN 11], [BER 12]. En utilisant la technique OSM, la convergence de l'optimisation du transformateur a été atteinte avec un nombre très réduit d'appels au FEM. Ainsi, la technique OSM est proposé ici pour résoudre chaque problème d'optimisation disciplinaire du transformateur d'isolement.

Décomposition du système complexe selon les composants physiques

La décomposition orientée objet porte sur le système complexe vu comme l'ensemble de ses sous-systèmes et composants physiques constitutifs. Ainsi, le système complexe peut être décomposé en ses sous-systèmes correspondants. A leur tour, les sous-systèmes peuvent être décomposés en des composants et la décomposition peut se poursuivre vers le bas pour les composants élémentaires de la structure analysée. Tous les composants et sous-systèmes interconnectés fonctionnent ensemble, régissant les fonctionnalités du système.

Les stratégies d'optimisation portant sur les systèmes décomposés à base d'objets ont pour rôle de concevoir de façon optimale les différents composants de la structure décomposée, tout en gérant les relations entre tous ces composants et sous-systèmes, assurant la cohérence du système dans son ensemble. En fonction du niveau de décomposition souhaité, la structure peut être partitionnée en plusieurs éléments, hiérarchiquement disposés sur deux ou plusieurs niveaux. Pour chaque élément de la hiérarchie, un problème d'optimisation est formulé. Ainsi, comme pour les méthodes de décomposition disciplinaire, la charge de calcul de l'optimisation est distribuée à travers la structure hiérarchique. Par rapport à la décomposition disciplinaire d'un système, où aucune priorité d'évaluation ne peut être établie entre les disciplines et que la décomposition a été effectuée sur deux niveaux, la décomposition à base d'objets des systèmes permet une généralisation de la décomposition à autant de niveaux que nécessaire. Dans une telle structure décomposée, un élément à un niveau intermédiaire de la hiérarchie ne dépend directement que de l'unique sous-système auquel il appartient, appelé élément parent. Egalement, il dépend de ses éléments constitutifs, situés au niveau inférieur et qui sont appelés éléments enfants.

Parmi les stratégies portant sur la décomposition à base d'objets des systèmes complexes, deux sont les plus représentatives, la cascade analytique des cibles (en anglais : Analytical Target Cascading – ATC), ou simplement la cascade des cibles (TC) et la Coordination par le Lagrangien Augmenté (en anglais : Augmented Lagrangian Coordination – ALC). La stratégie ATC introduite

par Michelena et al. au début des années 2000 [MIC 99], [MIC 03], [KIM 01] a été proposée comme une technique pour formaliser le processus de développement de produits industriels. Cette technique consiste à propager « en cascade » les spécifications générales du produit à tous les éléments constitutifs dans la hiérarchie décomposée. Les différents éléments de la hiérarchie visent à atteindre les cibles imposées en modifiant leurs variables locales de conception. Ainsi, les objectifs sont propagés du haut vers le bas jusqu'aux éléments les plus bas de la hiérarchie et leurs réponses sont remontées jusqu'en haut de la hiérarchie. Différentes techniques de coordination sont utilisées pour assurer la cohérence et la convergence de la conception optimale. Selon la stratégie de l'ATC, un élément de la hiérarchie échange des informations seulement avec ses éléments parent et enfants situés aux niveaux adjacents. La formulation ATC n'autorise pas le couplage horizontal direct entre les éléments du même niveau ou de niveaux différents, autres que ses éléments parent et enfants. Néanmoins, dans la pratique, il existe également des problèmes de conception optimale basés sur la décomposition objet qui présentent des couplages forts entre les éléments de la hiérarchie. Pour résoudre ce genre de problèmes, une version modifiée de l'ATC, nommé ALC, présentant la même structure que celle de l'ATC a été proposée par Tosserams et al. [TOS 07], [08] TOS. Cette version permet le couplage entre différents éléments dans la structure décomposée, ce qui représente cependant une généralisation de structures organisées non-hiérarchiques de l'ATC. La méthode de coordination employée par l'ALC utilise des fonctions globales du système et son fonctionnement est basé sur la théorie de la dualité du Lagrangien pour assurer la convergence de l'optimisation et la cohérence de la conception.

La décomposition hiérarchique d'un système complexe, basée sur les objets, est présentée dans la Figure 7.

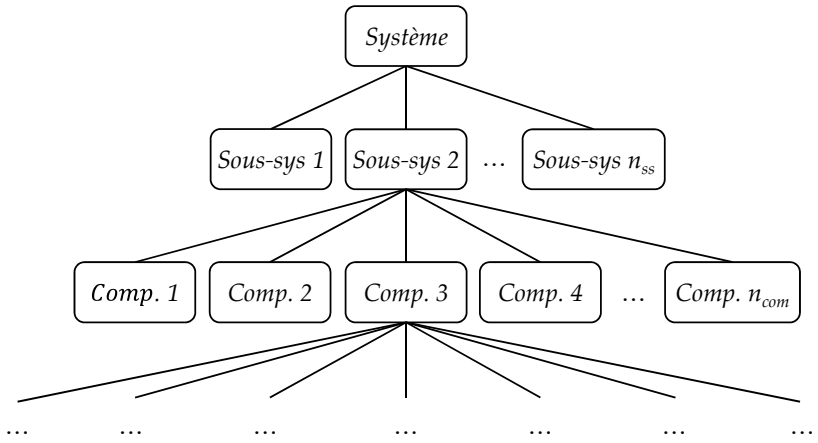


Figure 7: Décomposition du système complexe basée sur les objets

La stratégie d'optimisation « Analytical Target Cascading » (ATC) (pp. 142-169)

La stratégie d'optimisation multi-niveaux par mise en cascade des cibles (en anglais : Analytical Target Cascading – ATC ou Target Cascading – TC) est une stratégie basée sur la décomposition d'un système complexe en plusieurs niveaux hiérarchiques. Cette stratégie a été introduite au début des années 2000 comme une technique de formalisation du processus de développement de produits industriels [MIC 99a].

Le mécanisme de cette stratégie consiste à propager les objectifs généraux du cahier des charges du produit à tous les éléments de la hiérarchie décomposée. Les différents éléments de la hiérarchie visent à atteindre les objectifs imposés par la modification de leurs variables locales de conception. Ainsi, les objectifs sont propagés du haut vers le bas jusqu'aux éléments les plus bas de la hiérarchie et leurs réponses sont renvoyés vers le haut de la hiérarchie. Différentes techniques de coordination sont utilisées pour assurer la cohérence et la convergence de la configuration optimale. Selon la stratégie de l'ATC, un élément de la hiérarchie échange des informations seulement avec son élément parent, et ses enfants situés aux niveaux adjacents. Le couplage horizontal direct entre éléments du même niveau ou de niveaux différents, autres que les éléments parent et enfants est autorisé par la formulation ATC.

Les premières applications de l'ATC ont été dans l'industrie automobile, pour la conception optimale de véhicules [KIM 01], [KIM 03], [KIM 03a]. Par la suite, l'application de la stratégie d'optimisation ATC a été étendue au domaine de la conception des avions [ALL 06a]. En outre, cette méthode a été appliquée avec succès non seulement pour l'optimisation d'un produit, mais aussi pour une famille de produits, par exemple, la conception d'une famille de voitures [KOK 02], la conception d'une famille d'avions [ALL 06], la conception d'une famille de voitures de course [FER 09]. Un large éventail d'applications, telles que la conception des bâtiments [CHO 05], des problèmes d'optimisation structurale [MIC 06], et la conception d'une pompe à eau électrique [ALL 08], [ALL 09] ont été également abordées en utilisant la formulation ATC. Un problème de conception multidisciplinaire impliquant à la fois des décisions de l'ingénierie et du domaine de marketing a été abordé dans [MIC 05]. Au laboratoire L2EP à l'Ecole Centrale de Lille, les thèses de Moussouni-Messad et Kreuawan ont porté sur la conception optimale des systèmes ferroviaires [MOU 09a], [KRE 08].

La stratégie d'optimisation ATC a été initialement conçue avec une structure hiérarchique. Néanmoins, des systèmes complexes avec une décomposition non hiérarchique peuvent être également traités par l'ATC. L'élargissement de l'ATC aux problèmes multidisciplinaires non-hiérarchique a été abordé par Tosserams et al. [TOS 08a]. Cela a été utilisé pour la conception optimale d'un avion supersonique.

La plupart des problèmes réels d'ingénierie contiennent des variables discrètes de conception, par exemple le nombre de conducteurs, les sections standards de fils, le nombre de circuits électriques, etc. Habituellement, ces variables sont traitées comme des variables continues dans le processus ATC. Michalek et Papalambros ont proposé dans [MIC 06] une version modifiée de l'ATC, qu'ils appelé BB-ATC, qui intègre un algorithme classique d'optimisation discrète, Branch-and-Bound pour résoudre un problème de conception optimale des structures avec variables mixtes. Cependant, cela ajoute une charge de calcul importante pour le processus d'optimisation ATC. Cela n'est pratique que pour des grands problèmes hiérarchiques avec relativement peu de variables discrètes.

Les propriétés de convergence de l'architecture ATC ont été vérifiées pour une classe donnée de stratégies de coordination, sous certaines hypothèses relatives à l'attente des cibles [MIC 03]. Des stratégies de coordination plus complexes ont été développées dans la littérature pour le cas où les cibles globales du problème d'optimisation au niveau haut sont inatteignables.

Même si ATC n'appartient pas à la classe des méthodes MDO en raison de sa décomposition orientée objet, de nombreux auteurs la considèrent au sein de la même classe que MDO. L'ATC a des éléments communs à d'autres méthodes MDO, notamment la stratégie CO. Néanmoins, en comparaison avec la CO, qui est une stratégie strictement bi-niveaux, l'ATC est destinée à résoudre des problèmes avec une décomposition hiérarchique en deux ou plusieurs niveaux.

La présentation de la stratégie ATC débute par la formulation mathématique de cette méthode.

Formulation mathématique de base

En utilisant la stratégie de l'ATC, le problème complexe d'optimisation est décomposé en plusieurs problèmes plus petits et plus simples à résoudre, avec une disposition hiérarchique sur deux ou plusieurs niveaux. L'information est échangée entre ces problèmes sous la forme de cibles (en anglais : targets) et réponses. Ces infos sont exprimées en termes de variables partagées.

La présentation de la formulation mathématique de l'ATC commence par l'expression du problème d'optimisation initial du système non-décomposé, également dénommé formulation AIO.

$$\begin{aligned} \underset{\mathbf{z}}{\text{Minimise}} \quad & f(\mathbf{z}) \\ \text{avec} \quad & \mathbf{g}(\mathbf{z}) \leq \mathbf{0} \\ & \mathbf{h}(\mathbf{z}) = \mathbf{0} \end{aligned} \tag{8}$$

où $\mathbf{z}$ représente le vecteur de toutes les variables de conception du problème d'optimisation, f est la fonction objectif, $\mathbf{g}$ et $\mathbf{h}$ représentent les fonctions contrainte d'inégalité, respectivement égalité.

En employant une hypothèse liée au caractère additivement séparable de la fonction objectif, le problème initial est reformulé avec la séparation de la fonction objectif en plusieurs fonctions objectif, chacune liée à un sous-système du système à concevoir. Cette formulation est également nommée problème AIO structuré. Pour illustrer les propos, un exemple sur trois niveaux est donné.

Afin de pouvoir décomposer le problème global d'optimisation en plusieurs problèmes pour les sous-systèmes, une différence est faite entre les cibles imposées à chaque sous-système et les réponses de ce sous-système aux cibles reçues, par l'introduction de copies des variables échangées dans chaque sous-système. La structure décomposée pour l'exemple à trois niveaux est présentée dans la Figure 8.

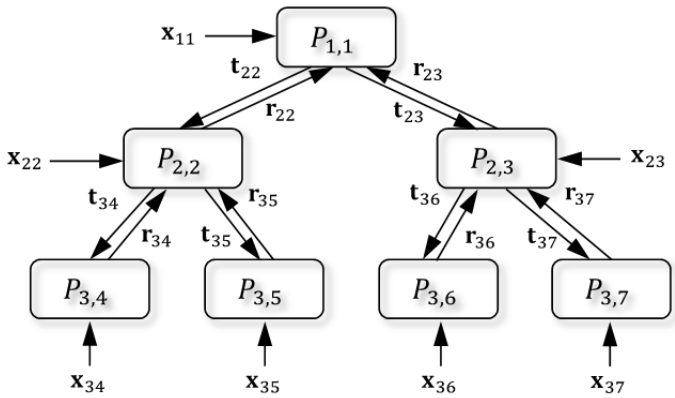


Figure 8 : Structure du problème initial, décomposée en trois niveaux

où P_{ij} représente le problème d'optimisation correspondant au sous-système j situé sur le niveau i, $\mathbf{x}_{ij}$ représente le vecteur de variables locales du problème P_{ij}, $\mathbf{t}_{ij}$ et $\mathbf{r}_{ij}$ représentent les vecteurs des cibles, respectivement réponses du problème P_{ij}.

Le problème d'optimisation AIO prêt à être décomposé est formulé dans (9).

$$
\begin{aligned}
\underset{\bar{\mathbf{x}}_{11},\cdots,\bar{\mathbf{x}}_{NM}}{\text{Minimise}} \quad & f = \sum_{i=1}^{N} \sum_{j \in \mathcal{E}_i} f_{ij}\left(\bar{\mathbf{x}}_{ij}\right) \\
\text{avec} \quad & \mathbf{g}_{ij}\left(\bar{\mathbf{x}}_{ij}\right) \leq \mathbf{0} \\
& \mathbf{h}_{ij}\left(\bar{\mathbf{x}}_{ij}\right) = \mathbf{0} \\
& \mathbf{c}_{ij} = \mathbf{t}_{ij} - \mathbf{r}_{ij} = \mathbf{0} \\
\text{et} \quad & \bar{\mathbf{x}}_{ij} = \left[\mathbf{x}_{ij}, \mathbf{r}_{ij}, \mathbf{t}_{(i+1)k_1}, \cdots, \mathbf{t}_{(i+1)k_{C_{ij}}}\right] \\
& k \in \mathsf{C}_{ij} \qquad j \in \mathcal{E}_i \qquad i = 1, \cdots, N
\end{aligned}
\tag{9}
$$

où f_{ij} représente la fonction objectif du problème P_{ij}. $\bar{\mathbf{x}}_{ij}$ est le vecteur de variables de conception propre au problème P_{ij}. $\mathbf{g}_{ij}$ et $\mathbf{h}_{ij}$ sont respectivement les contraintes d'inégalité et d'égalité du P_{ij}. $\mathbf{t}_{ij}$ et $\mathbf{r}_{ij}$ sont respectivement les vecteurs des cibles et réponses du problème P_{ij} et C_{ij} représente le nombre d'enfants du problème P_{ij}.

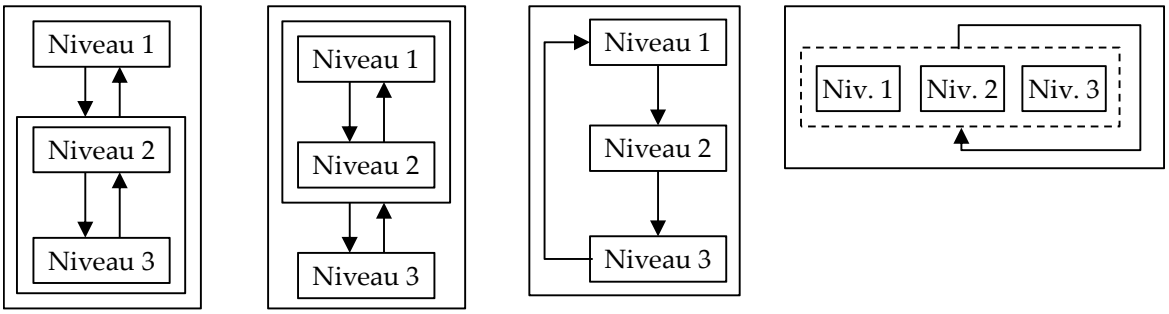
Pour que le problème décomposé soit cohérent, les variables réponse $\mathbf{r}_{ij}$ doivent attendre les cibles reçues $\mathbf{t}_{ij}$. Cette condition est imposée par les contraintes additionnelles $\mathbf{c}_{ij} = \mathbf{t}_{ij} - \mathbf{r}_{ij} = \mathbf{0}$, qui représentent les contraintes de cohérence du problème d'optimisation décomposé.

La clé de l'optimisation par la formulation de l'ATC des problèmes décomposés est la gestion des contraintes qui assurent la cohérence du processus. Ces contraintes dépendent de toutes les variables échangées (cibles $\mathbf{t}_{ij}$ et réponses $\mathbf{r}_{ij}$) et donc elles ne sont pas séparables comme la fonction objectif. Une stratégie spéciale de coordination doit être mise en place, avec le but de forcer l'attente des cibles et donc assurer implicitement la convergence du processus de l'ATC.

Stratégies de coordination du processus de l'ATC

Les stratégies de coordination associées au processus de l'ATC ont pour rôle d'organiser le flux de l'information dans la hiérarchie, en définissant l'ordre dans lequel les éléments d'un niveau donné de la hiérarchie sont évalués par l'ATC. Différentes stratégies convergentes peuvent être employées pour la coordination du processus d'optimisation de l'ATC.

Un nombre de stratégies de coordination imbriquées et non-imbriquées sont cités. Plusieurs méthodes de coordination imbriquées sont présentées dans la Figure 9. Certains d'entre elles ont des preuves de convergence (Figure 9a et Figure 9b), tandis que d'autres ont fait leurs preuves de convergence de manière empirique, à travers différentes applications.



e) Imbriqué vers le bas f) Imbriqué vers le haut g) Boucle simple h) Boucle simple, évaluation parallèle de niveaux

Figure 9 : Stratégies de coordination imbriquées et non-imbriquées

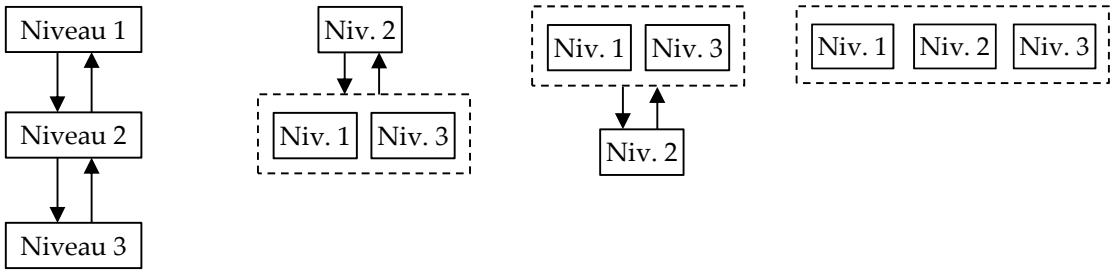
Les contours fermés dans la Figure 9 indiquent les boucles existantes entre les différents niveaux de la hiérarchie du problème décomposé.

Les deux premières stratégies de coordination, représentées graphiquement dans la Figure 9a et la Figure 9b consistent en deux boucles imbriquées. Les preuves de convergence de ces deux stratégies de coordination et d'autres stratégies similaires se trouvent dans [MIC 03]. La première stratégie, présentée dans la Figure 9a, implique la résolution complète de la boucle interne entre les deux niveaux inférieurs de la hiérarchie, pour chaque itération de la boucle externe. Une évaluation complète de la boucle interne est effectuée pour chaque itération de la boucle externe. La seconde stratégie, présentée dans la Figure 9b, est similaire à la première, avec cette différence que la boucle interne assure cette fois la cohérence entre les éléments du niveau supérieur et intermédiaire.

La troisième stratégie, présentée dans la Figure 9c, fait usage d'une seule boucle afin d'assurer la cohérence entre les différents niveaux de la hiérarchie. La boucle de coordination assure la convergence de cette séquence. Ce type de stratégie de coordination est appelé descente de coordonnées bloc (en anglais : Block Coordinate Descent – BCD) [LI 08].

La quatrième stratégie, présentée dans la Figure 9d, est en quelque sorte similaire à la stratégie présentée dans la Figure 9c mais avec un grand avantage ; tous les niveaux sont évalués simultanément. Celle-ci permet d'effectuer la tâche d'optimisation de tous les niveaux en parallèle. Une boucle de coordination s'assure de la cohérence entre tous les niveaux de la hiérarchie.

Les stratégies de coordination précédemment présentées emploient une ou plusieurs boucles pour assurer la cohérence entre les valeurs des cibles et les réponses générées par chaque sous-problème. La présence de ces boucles implique un effort de calcul conséquent, ce qui pénalise le temps global consacré à la tâche d'optimisation. Pour réduire la complexité de calcul ajoutée par ces boucles, les stratégies de coordination présentées dans la Figure 10 proposent la suppression de ces boucles et la réduction de la charge de calcul.



e) Un niveau évalué à la fois f) Evaluation paire-impaire g) Evaluation impaire-paire h) Evaluation simultanée de niveaux

Figure 10 : Stratégies de coordination sans boucle

La stratégie de coordination présentée dans la Figure 10a utilise une évaluation séquentielle des niveaux. La boucle globale du processus d'optimisation ATC est chargée d'assurer la cohérence entre les cibles et les réponses générées à chaque niveau. Les cibles sont mises en cascade du haut vers le bas de la hiérarchie, puis renvoyées au niveau le plus haut, en suivant le même chemin.

Les stratégies présentées dans les Figures 10b et 10c représentent deux versions de la stratégie précédente (Figure 10a). Ces stratégies exploitent la spécificité du flux d'informations entre les niveaux, principalement le fait que les éléments du niveau intermédiaire ont besoin d'informations à la fois des niveaux supérieur et inférieur, afin de calculer les réponses et les cibles pour le niveau supérieur, respectivement inférieur. Cependant, les éléments du niveau haut et bas peuvent être évalués en même temps. Une technique appropriée employant ces méthodes de coordination a été proposé par Tosserams et al. dans [TOS 06].

La dernière stratégie de coordination, présentée dans la Figure 10d est la plus généralisée, où tous les sous-problèmes de tous les éléments sont évalués en parallèle. Comme dans les trois stratégies précédentes, la boucle globale est en charge de l'accord entre les différentes cibles et réponses, mais cette fois-ci tous les niveaux sont évalués en même temps. La charge de calcul du problème d'optimisation est réduite au maximum au détriment de la complexité de mise en œuvre supplémentaires et la difficulté d'interpréter les différentes grandeurs échangées. Pour cette stratégie, une méthode d'approximation quadratique de la diagonale tronquée (en anglais : Truncated Diagonal Quadratic Approximation – TDQA), a été proposée par Li et al. dans [LI 08].

Toutes les stratégies de coordination présentées dans la Figure 9 et Figure 10 ont leurs avantages et leurs inconvénients et nécessitent l'association d'une certaine méthode pour gérer les contraintes de cohérence en local, pour chaque sous-problème de la hiérarchie. Cependant, l'évaluation de leur performance et leurs propriétés ne peut se faire indépendamment, mais avec une méthode de gestion des contraintes de cohérence. Différentes méthodes de gestion de contraintes de cohérence existent et celles-ci sont examinées par la suite.

Gestion des contraintes de cohérence

Les contraintes de cohérence supplémentaires introduites suite à la décomposition nécessitent une attention particulière. Ces contraintes de cohérence ne peuvent pas être considérées et traitées comme des contraintes strictes par chaque algorithme associé à chaque sous-problème d'optimisation, car elles dépendent toutes des variables cible et réponse et donc ne sont pas

séparables. Le non-respect de ces contraintes doit être accepté au cours du processus d'optimisation ATC, mais à la fin du processus d'optimisation, toutes les contraintes, notamment de cohérence, doivent être satisfaites afin d'obtenir une configuration faisable et un système complexe cohérent.

La présentation de la gestion des contraintes de cohérence débute par l'introduction de la première formulation mathématique de l'ATC, qui comporte des variables de tolérance additionnelles avec des contraintes associées [KIM 03], [KOK 02], [CHO 05]. Cette méthode de gestion des contraintes de cohérence a été associée aux deux premières stratégies de coordination imbriquées, présentées dans les Figures 9a et 9b.

Pour identifier les différentes variables de la formulation ATC, une représentation schématique d'un sous-problème d'optimisation P_{ij}, appartenant au niveau j de la hiérarchie, a été introduite. Les flux d'informations entre le sous-problème P_{ij} et ses éléments parent et enfants se matérialisent sous la forme de cibles et réponses, retrouvées sur le schéma présenté.

Comme mentionné auparavant, les contraintes de cohérence ne peuvent pas être imposées en tant que contraintes dures, afin que le processus de l'ATC puisse converger. Ainsi, ces contraintes doivent être relaxées à l'aide d'une fonction de pénalité qui est ajoutée à la fonction objectif.

Relaxation des contraintes de cohérence du problème d'optimisation AIO

Avec l'introduction de la fonction pénalité, la fonction objectif du problème AIO devient (10).

$$f = \sum_{i=1}^{N} \sum_{j \in \mathcal{E}_i} f_{ij}\left(\bar{\mathbf{x}}_{ij}\right) + \pi\left(\mathbf{c}(\bar{\mathbf{x}}_{11}, \cdots, \bar{\mathbf{x}}_{NM})\right) \tag{10}$$

où π représente la fonction de pénalité, qui a pour rôle de réduire les écarts ($\mathbf{c}$) entre les variables de liaison entre les différents sous-problèmes (variables de couplage et variables partagées).

C'est juste après avoir introduit le terme de pénalité, que le problème AIO peut être finalement décomposé suivant les différents sous-problèmes de la hiérarchie ATC.

La formulation générale d'un problème d'optimisation P_{ij} de la hiérarchie est exprimée en (11).

$$\begin{aligned} \underset{\bar{\mathbf{x}}_{ij}}{\text{Minimize}} \quad & f_{ij}\left(\bar{\mathbf{x}}_{ij}\right) + \pi\left(\mathbf{c}(\bar{\mathbf{x}}_{11}, \cdots, \bar{\mathbf{x}}_{NM})\right) \\ \text{avec} \quad & \mathbf{g}_{ij}\left(\bar{\mathbf{x}}_{ij}\right) \leq \mathbf{0} \\ & \mathbf{h}_{ij}\left(\bar{\mathbf{x}}_{ij}\right) = \mathbf{0} \\ \text{et} \quad & \bar{\mathbf{x}}_{ij} = \left[\mathbf{x}_{ij}, \mathbf{r}_{ij}, \mathbf{t}_{(i+1)k_1}, \cdots, \mathbf{t}_{(i+1)k_{c_{ij}}}\right] \end{aligned} \tag{11}$$

Le flux d'informations pour un sous-problème P_{ij} de la structure hiérarchique ATC avec l'expression (11) est présenté dans la Figure 11.

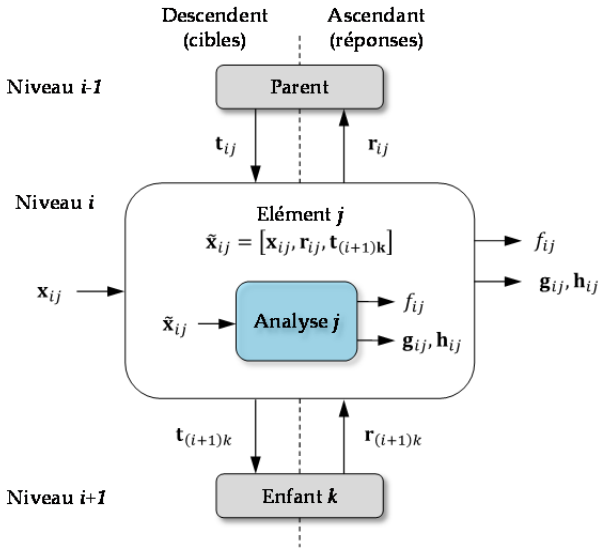


Figure 11 : Flux d'informations pour le sous-problème P_{ij}

Pour simplifier la notation de variables de la Figure 11, la notation $\mathbf{t}_{(i+1)\mathbf{k}} = \left[\mathbf{t}_{(i+1)k_1}, \cdots, \mathbf{t}_{(i+1)k_{c_{ij}}}\right]$ a été employée pour regrouper toutes les cibles calculées pour tous les enfants du problème P_{ij}.

Fonctions de pénalité pour la relaxation des contraintes de cohérence

Différentes méthodes générales de relaxation de contraintes existent dans la littérature et ont été adaptées pour assurer la cohérence du processus de l'ATC. Une taxonomie des différentes fonctions de pénalité employées avec les différentes stratégies de coordination est présentée dans [LI 08a]. La méthode de relaxation de contraintes la plus basique est représentée par la fonction de pénalité quadratique (Q). Des méthodes de pénalité plus complexes, telles que la relaxation lagrangienne (L) et le Lagrangien augmenté (AL) sont destinées à accroître la robustesse de l'optimisation.

Les trois fonctions de pénalité pour la relaxation des contraintes sont décrites. Les formulations mathématiques pour ces fonctions de pénalité sont données de manière générale dans le Tableau 1.

Tableau 1

Méthode de relaxation	**Formulation de la fonction pénalité, $\pi(\mathbf{c})$**
Pénalité Quadratique (Q)	$\pi_Q(\mathbf{c}) = \|\mathbf{w} \circ \mathbf{c}\|_2^2 = \sum_{i=1}^{N} \sum_{j \in \mathcal{E}_i} \left\|\mathbf{w}_{ij} \circ \mathbf{c}_{ij}\right\|_2^2 = \sum_{i=1}^{N} \sum_{j \in \mathcal{E}_i} \left\|\mathbf{w}_{ij} \circ \left(\mathbf{t}_{ij} - \mathbf{r}_{ij}\right)\right\|_2^2$
Relaxation lagrangienne (L)	$\pi_L(\mathbf{c}) = \mathbf{v}^T\mathbf{c} = \sum_{i=2}^{N} \sum_{j \in \mathcal{E}_i} \mathbf{v}_{ij}^T \mathbf{c}_{ij} = \sum_{i=2}^{N} \sum_{j \in \mathcal{E}_i} \mathbf{v}_{ij}^T \left(\mathbf{t}_{ij} - \mathbf{r}_{ij}\right)$
Lagrangien Augmenté (AL)	$\pi_{AL}(\mathbf{c}) = \mathbf{v}^T\mathbf{c} + \|\mathbf{w} \circ \mathbf{c}\|_2^2 = \sum_{i=1}^{N} \sum_{j \in \mathcal{E}_i} \left(\mathbf{v}_{ij}^T \mathbf{c}_{ij} + \left\|\mathbf{w}_{ij} \circ \mathbf{c}_{ij}\right\|_2^2\right)$ $= \sum_{i=1}^{N} \sum_{j \in \mathcal{E}_i} \left(\mathbf{v}_{ij}^T \left(\mathbf{t}_{ij} - \mathbf{r}_{ij}\right) + \left\|\mathbf{w}_{ij} \circ \left(\mathbf{t}_{ij} - \mathbf{r}_{ij}\right)\right\|_2^2\right)$

où le signe "∘" désigne la multiplication terme à terme, ou le produit de Hadamard, le symbole $\|\cdot\|_2^2$ représente le carré de la norme l_2, ou la norme euclidienne, $\mathbf{w}$ dans l'expression de la pénalité quadratique est le vecteur de coefficients de pondération, $\mathbf{v}^T$ représente l'ensemble de multiplicateurs du Lagrangien et $\mathbf{w}$ dans la formulation AL est un terme quadratique additionnel.

Les différentes techniques pour l'implémentation de ces fonctions de pénalité et les formulations mathématiques associées sont présentées. Les méthodes pour l'estimation des différents coefficients et leur mise à jour sont décrites. Les difficultés liées à l'implémentation de chaque relaxation sont présentées ensemble avec les avantages et les inconvénients de chaque relaxation.

Une différence importante existe entre la formulation d'un problème d'optimisation avec des cibles atteignables et inatteignables. Pour un problème d'optimisation avec un cahier des charges faisable (cibles atteignables), une simple relaxation quadratique en combinaison avec une stratégie de coordination imbriquée, telle que celles présentées dans les Figures 9a et 9b, est faisable. Néanmoins, la résolution d'un problème d'optimisation avec des cibles inatteignables est beaucoup plus judicieuse. Elle nécessite une technique plus complexe (L ou AL) pour assurer à la fois la convergence du processus ATC et la cohérence de la configuration obtenue.

Le processus ATC avec la relaxation AL représente une mise en œuvre robuste du processus d'optimisation ATC, et bénéficie également de preuves mathématiques de convergence. Elle a été appliquée avec succès pour résoudre des problèmes mathématiques et des problèmes de conception industrielle, comme des problèmes de conception des aéronefs [ALL 06a]. Néanmoins, l'utilisation d'une technique de coordination imbriquée impact sur l'effort de calcul total de l'optimisation du processus ATC.

Pour éviter l'emploi d'une stratégie de coordination imbriquée avec la méthode de relaxation AL, la combinaison de la relaxation AL et la méthode de la direction alternée des multiplicateurs (AL-AD) a été proposé par Tosserams et al. dans [TOS 06]. La méthode AL-AD consiste à alterner l'évaluation des problèmes d'optimisation des éléments de chaque niveau de la hiérarchie ATC. Cette méthode est bien conçue pour fonctionner avec les stratégies de coordination avec évaluation alternée des niveaux paire-impaire et impaire-paire présentées dans la Figure 10b et 10c.

Cette formulation de l'ATC, en utilisant la relaxation par le Lagrangien augmenté (AL) en combinaison avec la méthode de la direction alternée des multiplicateurs (AL-AD) a été implémentée dans Matlab® sous la forme d'une plate-forme d'optimisation multi-niveaux.

Pour montrer le fonctionnement du processus ATC, un exemple simple, mathématique, a été choisi. Le problème d'optimisation a été décomposé en trois sous-problèmes, disposés de manière hiérarchique sur deux niveaux. Des formulations avec cibles atteignables et inatteignables ont été utilisées. Avec des cibles atteignables, le processus ATC arrive à converger beaucoup plus rapide que la formulation impliquant des cibles inatteignables (17 itérations par rapport à 123 itérations), ce qui montre la difficulté introduite par les cibles inatteignables. Des mesures sont introduites pour quantifier et suivre la convergence du processus ATC, ainsi que le degré de cohérence de la configuration optimale.

Un certain nombre d'améliorations de la formulation ATC sont cités. Ainsi, pour résoudre des problèmes d'optimisation avec des modèles bruités, Choudhary et al. proposent dans [CHO 05] d'utiliser des algorithmes non-déterministes pour les sous-problèmes, comme ceux de la classe

MBDO. Afin d'alléger la charge de calcul introduite par les variables partagées entre différents sous-problèmes d'un même niveau, Guarnieri et al. proposent dans [GUA 11] l'utilisation d'une version modifié de SQP pour résoudre le problème d'optimisation du niveau haut de la hiérarchie ATC. Ainsi, les calculs de gradient inutiles en fonction des variables partagées excluent les simulations du modèle du niveau haut. Pour les cas où les modèles des sous-problèmes sont des simulations lourdes avec des variables de couplage représentées par des vecteurs de grande dimension, Alexander et al. proposent dans [ALE 11] et [ALE 11a] proposent d'utiliser des représentations réduites pour les vecteurs, comme les réseaux de neurones avec des fonctions radial basiques (en anglais : Radial Basis Function Artificial Neural Network – RBF-ANN) et décomposition orthogonale appropriée (en anglais : Proper Orthogonal Decomposition – POD).

Dans ce manuscrit, il est proposé une amélioration de la formulation ATC par l'introduction des méta-modèles dans les optimisations des sous-systèmes, afin de réduire le temps de l'optimisation. Ainsi, une des techniques MBDO du chapitre 2, la technique d'approximation avec méta-modèle adaptative, a été intégrée dans les optimisations des sous-problèmes de la structure multi-niveaux. Cela a pour but d'alléger la charge de calcul des modèles fins, qui se traduit par une réduction importante du temps de calcul total du processus ATC.

Une plate-forme d'optimisation multi-niveaux a été implémentée dans Matlab® afin d'adresser des problèmes d'optimisation avec décomposition en un nombre quelconque de sous-problèmes repartis sur autant de niveaux que souhaité. Cette plate-forme offre la possibilité d'interaction en réseau entre différents spécialistes ou équipes de spécialistes, chacune avec une expertise différente.

La plate-forme d'optimisation développée a été utilisée pour résoudre un cas-test d'optimisation de la Société Alstom Transport. L'application consiste de la conception optimale d'un système de stockage d'énergie par super-capacités (SC), embarqué à bord d'un tramway (en anglais : ultra-capacitor energy storage system on-board a tramway – UC-ESS). Le but de l'étude est de trouver les dimensions optimales des composants du pack SC, en tenant compte de l'interaction du pack avec le système tramway. Le système complexe tramway avec pack SC a été décomposé en quatre sous-systèmes, répartis sur deux niveaux, selon la structure ATC. Au niveau haut de la hiérarchie, le sous-problème consiste en un modèle de simulation du parcours du tramway. Ce dernier modèle est le plus complexe parmi les quatre. Au niveau inférieur, trois sous-problèmes sont définis, pour chaque composant considéré : le pack SC, la bobine du hacheur et le radiateur du module de puissance. Pour chacun des composants, des modèles relativement simples sont considérés. Le but de l'optimisation est de minimiser le prix du pack SC et également le coût de l'énergie soutirée de la caténaire. Le problème d'optimisation multi-niveaux est résolu par la plate-forme développée et la formulation ATC avec les améliorations proposées.

Conclusion

Au sein de **l'équipe Optimisation du laboratoire L2EP de l'Ecole Centrale de Lille**, nous sommes engagés à **rechercher des méthodes et des méthodologies pour poser et résoudre des problèmes complexes de conception optimale**. Le travail présenté dans ce manuscrit est destiné à apporter une nouvelle perspective sur le processus de conception optimale des systèmes complexes. Les méthodes d'optimisation basées sur l'utilisation de méta-modèles montrent un

grand potentiel pour les processus réels d'ingénierie de conception optimale. Plusieurs améliorations pourraient être abordées dans les futurs travaux connexes. Une voie d'amélioration intéressante consisterait en l'intégration des variables discrètes dans le processus d'optimisation basé sur méta-modèles. L'approche d'optimisation basée sur la décomposition d'un système en ses composants offre la possibilité de la distribution de la complexité du calcul. Une application ferroviaire à deux niveaux a été abordée en utilisant la méthode de l'ATC basée sur la décomposition orientée objet. L'intégration des méthodes pluridisciplinaires au sein des formulations multi-niveaux peut être une voie de développement intéressant dans l'avenir, en vue d'aborder des problèmes plus complexes de conception optimale de systèmes.

L'**optimisation** s'avère un **outil incontournable**, lorsque la capacité de compréhension des ingénieurs est dépassée par la complexité sans cesse croissante de la tâche de **conception optimale**. La démarche de conception traditionnelle des ingénieurs consiste en une résolution du problème étape par étape, en adoptant diverses hypothèses simplificatrices en cours de route, principalement basée sur l'expérience et l'intuition. Plus d'efforts doivent être consacrés à la formalisation des tâches de conception optimale de la Société Alstom dans des problèmes d'optimisation appropriées, en amont du cycle de conception, de sorte qu'un plus grand nombre de degrés de liberté soient disponibles et inclus ainsi dans le processus d'optimisation.

Approches multi-niveaux pour la conception systémique optimale des chaînes de traction ferroviaire

Dans le contexte actuel de mondialisation des marchés, le processus classique de conception par essais et erreurs, traditionnellement employé par les ingénieurs, n'est plus capable de répondre aux exigences de plus en plus accrues en termes de délais très courts, réduction des coûts de production, etc. L'outil d'optimisation propose une réponse à ces questions, en accompagnant les ingénieurs dans la tâche de conception optimale.

L'objectif de cette thèse est centré sur la conception optimale des systèmes complexes. Deux approches d'optimisation sont abordées dans ce travail: l'optimisation par modèles de substitution et la conception optimale basée sur la décomposition des systèmes complexes.

L'utilisation de la conception assistée par ordinateur (CAO) est devenue une pratique régulière dans l'industrie. La démarche d'optimisation basée sur modèles de substitution est destinée à répondre à l'optimisation des dispositifs avec ce genre de modèles coûteux de simulation, tels que les éléments finis (EF) en électromagnétisme. L'optimisation multi-objectif se présente comme un outil d'aide à la décision, en aidant le concepteur à prendre une décision éclairée. Le calcul distribué est utilisé pour réduire le temps global du processus d'optimisation.

Les systèmes d'ingénierie tels que les chaînes de traction ferroviaires sont trop complexes pour être traités comme un tout. Les stratégies d'optimisation basées sur la décomposition cherchent à répondre à la conception optimale de ces systèmes. Les approches de décomposition par modèle, discipline ou objet visent à distribuer la charge de calcul. Des stratégies de coordination multi-niveaux sont utilisées pour gérer le processus d'optimisation. Ces approches permettent à chaque équipe de spécialistes de travailler sur leur expertise de façon autonome. Les techniques d'optimisation à base de modèles de substitution peuvent être intégrées dans les stratégies d'optimisation multi-niveaux, allégeant ainsi la charge de calcul.

Les approches d'optimisation développées au sein de ce travail sont appliquées pour résoudre plusieurs problèmes d'optimisation électromagnétiques, ainsi que la conception optimale d'un système de traction ferroviaire de la Société Alstom.

Mots-clés :

- Conception optimale des systèmes électromagnétiques
- Optimisation par modèles de substitution
- Conception optimale de systèmes basée sur décomposition
- Optimisation multi-niveaux
- Chaîne de traction ferroviaire

Multi-level approaches for optimal system design in railway applications

Within a globalized market context, the classical trial-and-error design process traditionally employed by engineers is no longer capable of answering to the ever-growing demands in terms of short deadlines, reduced production costs, etc. The optimization tool presents itself as an answer to these issues, accompanying the engineers in the optimal design task.

The focus of this thesis is centered on the optimal design of complex systems. Two main optimization approaches are addressed in this work: the metamodel-based design optimization approach and the decomposition-based complex systems optimal design.

The use of computer-aided design/engineering (CAD/CAE) software has become a regular practice in the engineering design process. The metamodel-based optimization approach is intended to address the optimization of devices represented by such computational expensive simulation models, as the finite element analysis (FEA) in electromagnetics. The multi-objective optimization stands as a decision-making support tool, helping the design engineer make an informed decision. The distributed computation is employed to reduce the overall time of the optimization process.

Engineering systems such as railway traction systems are too complex to be addressed as a whole. The decomposition-based optimization strategies are intended to address the optimal design of such systems. Model, discipline or object-based decomposition approaches intend to distribute the computational burden across the system. Multi-level coordination strategies are used to manage the optimization process. Each team of specialists can work independently at the object of their expertise. The metamodel-based optimization techniques can be integrated within the multi-level decomposition-based strategies, reducing the computational burden.

The optimization approaches developed in this work are applied for solving several electromagnetic optimization problems and a railway traction system optimal design problem of the Alstom Company.

Keywords:

- Optimal design of electromagnetic devices
- Metamodel-based optimization
- Decomposition-based complex systems optimal design
- Multi-level optimization
- Railway traction system