Design of multiobjective optimization algorithms and theoretical analysis of evolution strategies

Cheikh Saliou Toure

INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2021IPPAX070

Thèse de doctorat

ÉCOLE POLYTECHNIQUE

IP PARIS

Design of multiobjective optimization algorithms and theoretical analysis of evolution strategies

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à Ecole Polytechnique

École doctorale n°574 Ecole Doctorale de Mathématiques Hadamard (EDMH)
Spécialité de doctorat: Mathématiques appliquées

Thèse présentée et soutenue à Palaiseau, le 22 Octobre 2021, par

CHEIKH SALIOU TOURÉ

Composition du Jury :

Randal Douc Professeur, Telecom SudParis (Département CITI)	Président
Aguêmon Yves Atchadé Professeur, Université de Boston (Department of Mathematics and Statistics)	Rapporteur
Kathrin Klamroth Professeur, Université de Wuppertal (School of Mathematics and Natural Sciences)	Rapporteur
Youhei Akimoto Professeur, Université de Tsukuba (Faculty of Engineering)	Examinateur
Tobias Glasmachers Professeur, Université de la Ruhr à Bochum (Theory of Machine Learning)	Examinateur
Anne Auger DR, Inria Saclay, IP Paris (CMAP)	Directrice de thèse
Frédéric Huguet Storengy	Invité

Abstract

This work is dedicated to zero-order black-box optimization, where only a sequence of function evaluations is available for the update of the optimization algorithm. Evolutionary algorithms are commonly used to solve this type of problems. Among them, evolution strategies like CMA-ES are state-of-the-art optimization algorithms for zero-order black-box optimization problems with a continuous search space. Particular aspects of the CMA-ES are the recombination mechanism and the non-elitist selection scheme, that are crucial to deal with local irregularities and multimodality. A multiobjective CMA-ES (with recombination and non-elitist selection scheme) is then particularly in demand for real world applications, to tackle multiobjective problems with local Pareto fronts.

We design that type of multiobjective optimizers. More specifically, a new multiobjective indicator called Uncrowded Hypervolume Improvement (UHVI) is created, along with a framework of multiobjective optimizers called Sofomore. By instantiating Sofomore with CMA-ES, COMO-CMA-ES is obtained. The COMO-CMA-ES algorithm is experimented on bi-objective functions that we analyze in details in this thesis, that are the bi-objective convex quadratic problems. Interestingly, linear convergence results are empirically observed, which is the optimal linear behavior we can get since CMA-ES converges linearly on strictly convex-quadratic functions. A Python package called $\mathrm{pycomocma}$ and a Matlab interface are developed in this work for COMO-CMA-ES and the Sofomore framework.

On a theoretical perspective, we analyze global linear convergence of evolution strategies with recombination that include well-known optimization algorithms, on a wide class of functions that are the scaling-invariant functions. Our main condition for convergence is that the expected logarithm of the step-size must increase on nontrivial linear functions. We analyze thoroughly the class of scaling-invariant functions and emphasize similar properties that they share with the positively homogeneous functions.

Keywords— CMA-ES, COMO-CMA-ES, Sofomore, UHVI, HVI, hypervolume, multiobjective optimization, Pareto front, Pareto set, optimization, scaling-invariant functions, positively homogeneous functions, xNES, CSA-ES, drift conditions, Markov chains, ergodicity, geometric ergodicity, Lyapunov function, selection function.

Résumé

Ce travail concerne les algorithmes d'optimisation de type black-box, où seulement une suite des valeurs de la fonction à optimiser est disponible pour mettre à jour l'instance de l'algorithme d'optimisation. Les algorithmes évolutionnaires ont une bonne réputation pour la résolution de ce genre de problèmes, notamment le CMA-ES. Des aspects particuliers du CMA-ES sont le mécanisme de recombinaison et la sélection non-élitiste, qui sont cruciaux pour l'optimisation des fonctions irrégulières et multimodales. Un CMA-ES multiobjectif (avec recombinaison et sélection non-élitiste) est ainsi en forte demande pour les applications du monde réel, notamment pour résoudre les problèmes multiobjectifs avec des fronts de Pareto locaux.

Nous concevons ce type d'algorithmes. Plus spécifiquement, un nouvel indicateur multiobjectif appelé Uncrowded Hypervolume Improvement (UHVI) est proposé, de même qu'un cadre d'algorithmes multiobjectifs appelé Sofomore. En instanciant Sofomore avec CMA-ES, nous obtenons COMO-CMA-ES. Ce nouvel algorithme multiobjectif est testé sur les fonctions bi-objectives quadratiques et convexes, que nous analysons en détail dans cette thèse. Nous observons une convergence linéaire, ce qui est le comportement optimal pour un CMA-ES multiobjectif puisque le CMA-ES converge linéairement sur les fonctions quadratiques strictemement convexes. Un package Python appelé $\mathrm{pycomocma}$ et une interface Matlab sont développés pour COMO-CMA-ES et pour Sofomore.

D'un point de vue théorique, nous analysons la convergence linéaire de stratégies d'évolution avec recombinaison contenant des algorithmes d'optimisation très connus, sur une classe de fonctions large constituée des fonctions scaling-invariant. Notre principale condition de convergence est que l'espérance du logarithme du step-size doit croître sur les fonctions linéaires non triviales, ce qui est optimal comme condition. Nous analysons la classe de fonctions scaling-invariant et mettons l'accent sur les propriétés qu'elle partage avec les fonctions homogènes.

Mots clés— CMA-ES, COMO-CMA-ES, Sofomore, UHVI, HVI, hypervolume, optimisation multi-objective, front de Pareto, ensemble de Pareto, optimisation, optimisation mono-objective, fonctions scaling-invariant, fonction homogène, xNES, CSA-ES, conditions de drift, chaînes de Markov, ergodicité, ergodicité géometrique, fonction de Lyapunov, fonction de selection.

Remerciements

Je remercie ma directrice Anne Auger de m'avoir constamment guidé durant ma thèse, Nikolaus Hansen également d'avoir beaucoup contribué dans ce travail, et Dimo Brockhoff de m'avoir beaucoup aidé durant toutes ces années de recherche. Merci à l'Inria et à Storengy d'avoir co-financé cette thèse. Je remercie aussi toute l'équipe Randopt : Kostas, Alann, Eugénie, Marie-Ange, Paul Dufossé, Jingyun, Armand, Asma, Alexandre. J'ai beaucoup appris à vos côtés à travers nos réunions et discussions. Je remercie également mon superviseur de Storengy Frédéric Huguet d'avoir souvent recadré mes recherches et de m'avoir beaucoup aidé grâce à sa vaste expérience. Merci également à Arnaud, Vincent et Benoît d'avoir toujours œuvré pour une bonne conduite de mes recherches à Storengy. Merci beaucoup aux professeurs Klamroth Kathrin et Aguêmon Yves Atchadé d'avoir généreusement accepté de rapporter mon manuscrit de thèse. Je remercie également les autres membres du jury, à savoir les professeurs Randal Douc, Tobias Glasmachers et Youhei Akimoto. Merci également à Nasséra Naar de l'Ecole Polytechnique, à Marie Enée et à Valérie Berthou de l'Inria, et à tout le personnel de l'Inria, de l'Ecole Polytechnique et de Storengy qui, de près ou de loin, m'a aidé durant ce travail.

Merci à mes amis Julie, Vianney et William, d'avoir passé tous ces nombreux moments ensemble entre doctorants, surtout pendant le confinement, même si j'étais toujours dernier dans les jeux en ligne. Coach Julie qui nous a transformés - de façon un peu tyrannique - en sportifs de haut niveau, William et ses blagues marrantes et Vianney mon compatriote de la Start-up Nation grâce à qui j'ai regardé (avec ses amis du ch'Nord) mon meilleur film de cette année. Je remercie également tous les membres du bureau 2016, dont j'ai eu le plaisir de côtoyer au fur de mes années de thèse : Julie, Alann, Paul Jusselin, Paul Thévenin, Jean-Bernard, Hadrien, Perle, Florian, Mathieu, Bowen, Baptiste, Louis, Naoufal, Joffrey. Encore merci à Hadrien et à Perle de m'avoir beaucoup aidé à Station F. Je remercie également tous les doctorants du CMAP. A mes amis du séminaire clandestin de l'ENS Paris-Saclay : William, Thibault, Clément et le très passionné de la théorie des nombres Quentin. Merci également à mes amis doctorants du *Romanus luxuria* avec qui j'ai partagé de beaux souvenirs : Ambra, Marco, Heythem, Chiaretta, Pierre, Arnaud, Luke, Safir.

Je dédie ce travail à ma mère qui m'a toujours encouragé, et à mes frères Ma-

madou Yadali et Serigne Massamba, qui m'ont toujours guidé et conseillé. Quant à Serigne Massamba, cette thèse est autant son travail que le mien : depuis que j'ai quitté la maison familiale en 2004 pour aller au Prytanée de Saint-Louis, il ne s'est passé une semaine sans qu'on n'ait échangé, et qu'il ne m'oriente avec toute son énergie. Je remercie également mes belle-sœurs Mame Saye et Seynabou. Egalement les très généreux Hamet et Amy Babou, qui m'ont beaucoup aidé durant tout mon séjour parisien.

Je dédie ce travail à mon professeur de Mathématiques de classes préparatoires, Monsieur Pascal Guelfi du lycée Henri Poincaré de Nancy. Je le remercie et je continuerai de le remercier pour toujours : pour son humanisme, sa bonté et son énergie débordante qu'il prodigue volontiers. Sans lui, je ne pourrais jamais pousser jusqu'ici.

A mes amis les plus proches, avec un nom de groupe qui change tous les mois (à chaque fois que quelqu'un pense avoir compris le monde, avant de redevenir lucide). Louis, Macoumba, Mamadou Diégane, Mamadou Mansour, Robert et Yague. Sans vous et vos énergies positives, je ne serais sans doute pas là. Merci beaucoup et je vous dédie ce travail. Merci beaucoup à Zanga Ben et à Dina de m'avoir bien accueilli durant mon séjour à Michigan pour la conférence EMO 2019. Vous avez rendu mon séjour inoubliable et joyeux. Encore merci à Ouattara pour toutes nos discussions qui me projettent toujours au delà de moi-même. Je remercie également mon homonyme et jumeau le médecin et capitaine Saliou Sarr. Bonne chance et bon courage dans toutes tes missions. J'espère que, malgré tout, tu continueras de vivre de façon aussi intense et inspirante. Je remercie aussi toute la promotion 2004 du Prytanée de Saint-Louis.

Je dédie ce travail à mon défunt père et à toute la famille *Touré*, je vous remercie du fond du cœur pour tout et pour tous !

Contents

Introduction

Contents

Multiobjective optimization consists in finding the optima of a vector-valued function $\mathbf{f} : x \in \mathbb{R}^n \mapsto (\mathbf{f}_1(x), \ldots, \mathbf{f}_m(x)) \in \mathbb{R}^m$, where the considered partial order in $\mathbb{R}^m$ depends on f and is called the weakly Pareto dominance relation. The set of non comparable points in the search space with that Pareto relation is called the Pareto set and its image by $\mathbf{f}$ is called the Pareto front. Typically it is not finite. With a fixed budget of function evaluations or a fixed number of solutions on the objective space $\mathbb{R}^m$ or on the search space $\mathbb{R}^n$, a multiobjective optimizer aims to return solutions with a certain quality. The indicators that measure the quality of the solutions and guide the optimization are called quality indicators.

In a zero-order black-box scenario, only a sequence of function evaluations are available to the optimizer. The information related to the gradients is usually not available. This situation is commonly found in industrial use cases, where a function value is returned after executing a complex simulation code. This scenario is also called derivative-free optimization (DFO).

The state-of-the-art algorithms to solve the multiobjective zero-order black-box problems are multiobjective evolutionary algorithms, mainly thanks to their population-based nature that allows the generation of a variety of Pareto optimal solutions in a single run [49].

Evolution strategies are a class of randomized black-box optimization algorithms [90, 159, 160, 172, 173], well-known for tackling real world optimization problems that often come with difficulties such as non-linearity, non-convexity, non-separability, ill-

conditioning and multimodality. For a continuous domain optimization problem, the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [88, 95] is among the state-of-the-art zero-order black-box single-objective optimization algorithm. Based on the comparisons of the function values observed, it learns second order information [91]. Most of its internal parameters are already set within the algorithm design. Its generic formulation, robustness and invariance properties [96] are among the reasons why CMA-ES is used on a wide range of real-world applications, with little parameter tuning required [37, 100, 166].

1 Context of the thesis

This work is based on a project of the underground gas storage company Storengy. Its operation is to store natural gas in an underground reservoir during summer when there is a lot of gas available, and produce the gas during winter when there is a large demand in cold period. Storengy needs to predict the reservoir behavior in order to optimize the operation, therefore a numerical reservoir simulator is used for this purpose. Many parameters of the simulator related to the reservoir porosity and permeability have to be history matched on well measurements: daily water volume produced, cumulative yearly water volume, bottom hole pressure for field wells and water-gas interface position. Two or three objective functions are constructed based on these measures. Optimizing them is much needed to calibrate the models in order to obtain a good predictive model.

Typically, these objective functions are multimodal, non-convex and non-smooth, and are in a zero-order black-box scenario due to their non-smoothness and the expensiveness of the function values. Storengy is therefore very interested in optimizers that are suitable to handle these type of irregularities, especially in a multiobjective optimizer that can approximate the global Pareto front despite the multimodality of the objective functions.

A trait that makes an evolution strategy more robust to local irregularities and multimodality is the recombination mechanism. An evolution strategy with recombination samples λ candidate solutions—the offspring—based on the favorite solution and other parameters, chooses the $\mu \leq \lambda$ best among them—the parents—, and recombines the parents to obtain the new favorite solution. Typically, $\mu \geq 2$ and $\lambda \approx 2\mu$.

This mechanism also makes the evolution strategy converge faster to the global optimum [13]. In various experiments, evolution strategies with recombination converge or diverge linearly, $i.e.$ geometrically fast, on a wide class of functions [90].

An interesting class of functions where linear convergence or divergence (linear behavior) is often observed is the class of scaling-invariant functions. It contains norms, convex-quadratic functions, nontrivial linear functions and more generally increas-

ing transformations of positively homogeneous functions. Scaling-invariant functions are defined as preserving the f-comparisons of points when the points are scaled by the same positive scalar (with respect to a reference point). Various unanswered questions arise related to the study of this class to see how wide it is, compared to increasing transformations of positively homogeneous functions. We suspect that understanding this class of functions is a necessary step towards proving linear convergence or divergence of Evolution Strategies with recombination on it. Therefore natural general questions are:

What are the interesting properties of scaling-invariant functions? Are they connected to positively homogeneous functions?

(Chapter 3)

Linear convergence is the fastest possible convergence that we can obtain in evolution strategies [114, 182]. It is often observed experimentally [23, 90, 96, 105, 162]. However it is usually difficult to prove. The mathematical work often consists in doing drift analysis on Markov chains, or applying a Law of Large Numbers to an underlying ergodic Markov chain. In the literature, linear behavior is theoretically obtained for evolution strategies with one parent population size [2, 3, 19, 24, 48, 106–108, 110]. Yet linear convergence analysis of evolution strategies with recombination mechanism is inexistent. This brings the general and difficult following question.

Can we prove that evolution strategies with recombination converge or diverge geometrically fast on scaling-invariant functions, under regularity assumptions?

(Chapter 4)

The above questions concern theoretical analysis on evolution strategies with recombination. The latter is crucial for various single-objective real-world problems that are irregular and multimodal, like Storengy's problems. We suspect that in a multiobjective scenario, the recombination mechanism along with non-elitism selection can help deal with local Pareto sets.

Yet, we believe that an efficient multiobjective optimizer has to at least converge linearly on multiobjective strictly convex-quadratic problems (with a convergence notion that will be defined later on). Therefore before designing a multiobjective optimizer based on evolution strategies with recombination and non-elitism, we try to build subclasses of bi-objective convex quadratic problems that highlight different difficulties of a problem, mainly curvature of the Pareto front, non-separability and ill-conditioning of the objective functions. This leads us to the following question.

How to build subclasses of bi-objective convex-quadratic problems that have typical difficulties ?

(Chapter 6)

The CMA-ES is one of the state-of-the-art evolution strategies with recombination.

The idea of designing a multiobjective optimizer based on the CMA-ES emerged more than fifteen years ago. Yet if we have several CMA-ES instances that constitute a multiobjective optimizer, the total function evaluations per iteration is the number of CMA-ES instances times the offspring population size λ. That typically large number of evaluations per iteration makes it difficult to create an efficient multiobjective CMA-ES version. In [104], Igel et al. design a CMA-ES version without the recombination and non-elitist mechanism, replaced with the so-called $(1+1)$-selection mechanism with $1/5$-success rule or a $(\mu+\lambda)$-selection scheme [121, 159]. It is an elitist version of CMA-ES with μ parents and λ offspring, where the next μ parents are the best among the $(\mu+\lambda)$ parents and offspring. Based on this new version, Igel et al. design a multiobjective CMA-ES optimizer called MO-CMA-ES [104, 193].

Yet, the CMA-ES is less prone to getting stuck to sub-optimal local optima, compared to its elitist variants [104]. Therefore the desire for industrials to have an efficient multiobjective optimizer based on the CMA-ES, and the scientific interest to see the benefits of recombination and non-elitism in multiobjective optimization, are still present. This guides us to ask the following question.

Can we design a robust multiobjective algorithm based on the CMA-ES (non-elitist with recombination), linearly convergent on multiobjective strictly convex-quadratic problems? How does it perform compared to existing multiobjective algorithms?

(Chapters 7 and 8)

The pycma [89] package for the CMA-ES (with recombination) is actively maintained and is updated regularly. Therefore if a multiobjective optimizer is based on the CMA-ES, it is crucial to make it as compatible as possible to the pycma package. That way, an improvement on the single-objective algorithm will be easily transferred to the multiobjective version.

In addition for industrial purpose with our partner Storengy, the designed multiobjective algorithm should be implemented in Matlab. It should also fulfil the standard needs for industrial use cases, such as the possibility to parallelize the optimizer. Therefore a natural question is:

How to efficiently implement the designed multiobjective algorithms both in Python and in Matlab? Can the algorithm have a robust parallelized version?

(Chapter 9)

2 Contributions of the thesis

In this thesis, we study the above research questions. In that process, we address them at some extent and develop various innovative results.

We study for the first time the linear convergence analysis of evolution strategies with recombination. The algorithms considered are step-size adaptive evolution strategies, represented by a stochastic process $\{(X_k, \sigma_k)\,; k \in \mathbb{N}\}$, where X_k is the favorite solution and σ_k is the step-size at iteration k. We analyze linear behavior of step-size adaptive evolution strategies with recombination on scaling-invariant functions.

In multiobjective optimization, we develop a framework of multiobjective optimization algorithms called Sofomore. We instantiate the framework with CMA-ES as single-objective optimizers to obtain the newly COMO-CMA-ES, that converges geometrically fast on strictly-convex quadratic functions. We carry out the implementation of Sofomore and COMO-CMA-ES, both in Python and in Matlab. The code is compatible with the pycma package and has various options needed for industrial purpose.

2.1 Global linear convergence of evolution strategies with recombination on scaling-invariant functions

In Chapter 3, we thoroughly analyze scaling-invariant functions and present that under specific conditions, they are composites of strictly monotonic functions with positively homogeneous functions.

Specifically, we give necessary and sufficient conditions for a scaling-invariant function to be a composite of a strictly monotonic function with a positively homogeneous function. We also present various properties satisfied by sublevel sets of scaling-invariant functions. Surprisingly, we highlight that scaling-invariant functions can have highly pathological behaviors, by showing real-valued scaling-invariant functions that are not monotonic on any nontrivial interval. The results of Chapter 3 are published in the Journal of Optimization Theory and Applications (JOTA) and are presented in [184]:
Cheikh Toure, Armand Gissler, Anne Auger and Nikolaus Hansen, *Scaling-invariant functions versus positively homogeneous functions*, Journal of Optimization Theory and Applications, 2021.

We present in Chapter 4 linear behaviors of a class of step-size adaptive evolution strategies with recombination mechanism, on a wide class of functions that are the scaling-invariant functions. Our framework includes two well-known step-size adaptation mechanisms derived from the Exponential Natural Evolution Strategy (xNES) [70] and the Cumulative Step-size Adaptation (CSA) without cumulation, where the CSA with cumulation is the default step-size mechanism of CMA-ES [88].

We show that the logarithm of the distance of the incumbent to the optimum (or to a reference point if the algorithm diverges) divided by the number of iterations converges to a limit r. In addition the logarithm of the step-size divided by the number of iterations converges to the same limit r. That limit is the rate of convergence or

divergence. We also prove that the expected logarithm of the step-size change and the expected log-progress both converge to the rate r.

The rate is expressed as an expectation under an unknown invariant probability measure that comes from stability properties of a joint Markov chain defined on a general state space. Therefore we also give a central limit theorem to approximate that rate by Monte Carlo simulations.

Our main condition to obtain linear behavior is that on expectation, the logarithm of the step-size must increase on nontrivial linear functions. This is the tightest condition we can have, since it is equivalent to stating that the algorithm diverges geometrically on nontrivial linear functions. Note also that this condition is new compared to former conditions for linear convergence that we can find in the literature [3, 19, 24].

We also show how practical this condition is for concrete evolution strategies like xNES [70] without covariance matrix adaptation or CSA-ES without cumulation [88], by expressing it with respect to the parameters of the algorithms and to order statistics of standard normal distributions.

The linear behaviors in Chapter 4 are obtained on a wide class of functions, that are strictly increasing transformations of either C^1 scaling-invariant functions with a unique global argmin, or nontrivial linear functions.

All the results in Chapter 4 are submitted in the Journal of Global Optimization, by the authors Cheikh Toure, Anne Auger and Nikolaus Hansen.

2.2 Multiobjective optimization: UHVI, Sofomore, COMO-CMA-ES, Implementations

We develop in Chapter 7 a framework of multiobjective optimization algorithms called Sofomore: Single-objective Optimization For Optimizing Multiobjective Optimization pRoblEms. This framework allows to optimize a changing single-objective fitness by various single-objective algorithms. An incumbent of a single-objective optimizer approximates an element of the Pareto set. The number of single-objective optimizers is fixed and denoted by p. The goal of the multiobjective optimization that we consider is then to maximize the so-called p-optimal distribution of the hypervolume indicator [16, 22].

In previous study such as SMS-EMOA [30] or MO-CMA-ES [104, 193], the performance indicator used to guide the multiobjective optimization tends to steer dominated solutions towards regions already occupied by a non-dominated solution. We then introduce a new performance indicator called UHVI: Uncrowded Hypervolume Improvement. It creates a search bias towards the uncovered domain of the Pareto set.

We instantiate the Sofomore framework with the single-objective optimizer CMA-ES [88, 95] to obtain the COMO-CMA-ES algorithm: Comma-Selection Multiobjective CMA-ES. The comma-selection highlights the non-elitist mechanism inherent to the standard CMA-ES, in contrary to the elitist $(1+1)$-CMA-ES version used in MO-CMA-ES [104, 193].

The content of Chapter 7 was presented in the GECCO 2019 Conference at Prague, in the EURO 2019 Conference at Dublin and in the MACODA 2019 Workshop at Leiden. It is published in [185]:
Cheikh Toure, Nikolaus Hansen, Anne Auger and Dimo Brockhoff, *Uncrowded hypervolume improvement: COMO-CMA-ES and the Sofomore framework*, Proceedings of the Genetic and Evolutionary Computation Conference, 2019, pp 638–646.

We believe that efficient multiobjective evolution strategies must converge (with a convergence definition related to the Pareto front) linearly on multiobjective strictly convex-quadratic functions, like the efficient single-objective evolution strategies do for strictly convex-quadratic functions.

We thoroughly present in Chapter 6 the analysis of bi-objective strictly convex and quadratic functions, and give various descriptions of the Pareto front and the Pareto set for this class of problems.

We also construct subclasses of bi-objective convex-quadratic problems that aim to test the behaviors of a multiobjective algorithm with respect to different features of a problem. The theoretical analysis of Chapter 6 are published in [183]:
Cheikh Toure, Anne Auger, Dimo Brockhoff and Nikolaus Hansen, *On Bi-Objective convex-quadratic problems*, International Conference on Evolutionary Multi-Criterion Optimization, 2019, pp 3–14.

In Chapter 8, we benchmark the new multiobjective evolution strategy COMO-CMA-ES in the COCO (COmparing Continuous Optimizers) platform. The latter allows to compare different continuous black-box optimizers with respect to separability, ill-conditioning, multimodality and the structure of a problem.

We compare COMO-CMA-ES to well-known multiobjective optimizers NSGA-II [56] and MO-CMA-ES [104, 193]. Although COMO-CMA-ES is not designed to perform on archive-based metric such as the one used by COCO, we observe good performances of COMO-CMA-ES on COCO. That highlights the power of evolution strategies with recombination like the underlying single-objective CMA-ES. It also underlines the usefulness of an uncrowded performance indicator like the UHVI that guarantees the quality of sampled non-dominated offspring, when the mulitobjective optimizer's incumbents are close to the Pareto set. The results of Chapter 8 are published in [61]:
Paul Dufosse and Cheikh Toure, *Benchmarking MO-CMA-ES and COMO-CMA-ES on the Bi-objective bbob-biobj Testbed*, Proceedings of the Genetic and Evolutionary Computation Conference Companion, 2019, pp 1920–1927.

In Chapter 9, we present the Python package developed for this work, that is the

pycomocma package available on this link: https://github.com/CMA-ES/pycomocma. Similar to the CMA-ES python package pycma [89], it is implemented with the so-called ask-and-tell paradigm. The latter allows to have access to the optimizer at each iteration between the generation of new solutions with the ask method, and the update of the optimization instance with the tell method.

This package implements the Sofomore framework, and gives the possibility to derive the COMO-CMA-ES algorithm by instantiating the framework with CMA-ES instances.

A Matlab interface of COMO-CMA-ES is also produced for our industrial partner Storengy. This Matlab code works for Matlab 2014b versions or later, as it uses the Matlab py command. The installation of the Python package pycomocma is also necessary.

3 Overview of the chapters

Chapter 1: Continuous black-box optimization

In a zero-order black-box continuous optimization scenario, the goal is to minimize a function $f : \mathbb{R}^n \to \mathbb{R}$ where only the f-values, *i.e.* the $f(x)$ for a sequence of candidate solutions $x \in \mathbb{R}^n$, are available. This situation arises when only an oracle returning $f(x)$ for a given x is available. Therefore the optimization problem is usually derivative-free as we have no information related to the gradient of f. Continuous optimization concerns the optimization problem where the search space is continuous.

We present deterministic continuous black-box optimization algorithms like the Nelder-Mead method and some Trust-region methods. We also present randomized algorithms more refined than Pure random search. Namely simulated annealing and evolutionary algorithms. We then focus on evolution strategies, with a detailed presentation of Cumulative Step-size Adaptation Evolution Strategy (CSA-ES) and Covariance Matrix Adaptation Evolution Strategy (CMA-ES). The Information-Geometric Optimization viewpoint is finally given to embed CMA-ES in a broader class of optimization algorithms.

Chapter 2: Convergence of evolution strategies

Analyzing an evolution strategy is important to assess its pertinence. Pure random search converges, but with a speed of convergence so slow that this type of optimization algorithm is typically not used in practice. We present in this chapter some notions traditionally used to measure the speed of convergence of an evolution strategy.

Namely the almost sure linear convergence or divergence, the expected log-progress, the progress rate, the quality gain, the expected hitting time and the expected running time.

We also present briefly some methods used to establish convergence results. Ordinary Differential Equations (ODE) are one of them, with the IGO flow that can be seen as an ODE. We sketch how the law of large numbers is used via Markov chain analysis to obtain almost sure linear convergence or linear convergence of the expected log-progress. A drift analysis technique to bound the expected hitting time is also briefly presented.

Chapter 3: Scaling invariant functions versus Positively homogeneous functions

A function $f : \mathbb{R}^n \to \mathbb{R}$ is scaling-invariant with respect to a reference point $x^\star \in \mathbb{R}^n$ if for all $x, y \in \mathbb{R}^n$ and $\rho > 0$:

$$f(x^\star + x) \leq f(x^\star + y) \iff f(x^\star + \rho x) \leq f(x^\star + \rho y) \ .$$

Positively homogeneous functions are scaling-invariant functions. Recall that a function $p : \mathbb{R}^n \to \mathbb{R}$ is positively homogenous with degree $\alpha > 0$ if for all $x \in \mathbb{R}^n$ and $\rho > 0$: $p(\rho\,(x - x^\star)) = \rho^\alpha p(x - x^\star)$. Linear functions, norms, quadratic functions are positively homogeneous. Composites of strictly monotonic functions with positively homogeneous functions are scaling-invariant.

We show in this chapter the necessary and sufficient conditions for a scaling-invariant function to be composite of a strictly monotonic function with a positively homogeneous function. In particular, we show that continuous scaling-invariant functions are composites of a homeomorphism with a continuous positively homogeneous function.

We study sublevel sets of scaling-invariant functions generalizing well-known properties of positively homogeneous functions. In particular we show that under mild conditions, a scaling-invariant function has a unique global argmin if and only if its sublevel sets are compact sets.

The content of this chapter is published in the Journal of Optimization Theory and Applications (JOTA). It is presented in [184]:
Cheikh Toure, Armand Gissler, Anne Auger and Nikolaus Hansen, *Scaling-invariant functions versus positively homogeneous functions*, Journal of Optimization Theory and Applications, 2021.

Chapter 4: Global linear convergence of step-size adaptive evolution strategies with recombination on scaling-invariant functions

We analyze in this chapter the linear behavior of evolution strategies with recombination, where only the step-size and the incumbent are adapted. The covariance matrix adaptation and other refined mechanisms are not considered, so that the covariance matrix is always the square of the step-size times the identity matrix.

We present a generic step-size update defined such that the invariances to order and angle preserving transformations are conserved, which are crucial for the conception of evolution strategies [83, 167]. We show that both CSA-ES without cumulation and xNES without covariance matrix adaptation are particular cases of our framework.

We then prove linear divergence on strictly increasing transformations of linear functions and linear behavior on strictly increasing transformations of C^1 scaling-invariant functions with a unique global argmin. These functions include non convex functions and non quasiconvex functions. The main condition for these proofs is equivalent to the geometric divergence of the step-size on nontrivial linear functions.

The method we use is based on a Markov chain technique for analyzing evolution strategies. More specifically, we exploit the methodology established in [25] by proving that the underlying normalized Markov chain is ergodic.

The main results of the chapter are linear convergence or linear divergence of the step-size and the distance to the optimum (or to a reference point). The rate for linear behavior depends on an unknown probability measure that comes from the underlying Markov chain. A central limit theorem is then given to approximate that rate.

The content of this chapter is submitted in the Journal of Global Optimization by Cheikh Toure, Anne Auger and Nikolaus Hansen.

Chapter 5: Background on multiobjective evolutionary algorithms

We present in this chapter basic notions used to define the optimization of a vector-valued function $\mathbf{f} : x \in \mathbb{R}^n \longmapsto \mathbf{f}(x) = (\mathbf{f}_1(x), \ldots, \mathbf{f}_m(x)) \in \mathbb{R}^m$ where $m \geq 2$. Defining an order in $\mathbb{R}^m$ that generalizes the standard $\leq$ order in $\mathbb{R}$ for single-objective optimization is itself an important task. It is tackled by the notions of weakly Pareto dominance that create a partial order in $\mathbb{R}^m$. For a finite and nonempty set P, the subset of points that are not Pareto dominated are the non-dominated points of P, and we say that they have a Pareto rank of 1. Recursively, the set P is partitioned into a finite number of

subsets $P_1, \ldots, P_k$ where P_i has a Pareto rank i and is Pareto dominated by P_{i-1}. Any two elements of a subset P_i are not Pareto comparable. This is a first step towards sorting the set P.

To fully sort P, total orders in $\mathbb{R}^m$ compatible with the weakly Pareto dominance are presented. The construction of these orders are typically based on quality indicators, like the crowding distance and the hypervolume indicator. For a set of non Pareto comparable points, the performance indicator is a real-valued function that assigns to each point a real value, that is the sorting we use in this context. Hence given a finite nonempty population P, it is first partitioned into $P_1, \ldots, P_k$ where the points in P_i have a Pareto rank i. Second, P_i is sorted with respect to a performance indicator, independently of $P \setminus P_i$. Overall a point with smaller Pareto rank is always better. Between two points with the same Pareto rank, a point with a larger performance indicator value is always better. This scheme is the so-called two-way ranking [104].

The hypervolume contribution and the hypervolume improvement are performance indicators of the hypervolume indicator. A specific aspect of the latter is its strictly monotone property, $i.e.$ its property to conserve the Pareto dominance relation. This feature helps to transform the problem of finding p solutions on the Pareto set into a single-objective problem in $\mathbb{R}^{n \times p}$ that consists of maximizing the hypervolume (defined on the search space) with respect to the multiobjective function to optimize [16, 22].

We present well-known multiobjective evolutionary algorithms (the family of NSGA algorithms [56, 178], SMS-EMOA [30] and MO-CMA-ES [104, 193]) and observe how an underlying two-way ranking is inherent to the designs of these algorithms.

We finally emphasize some limitations regarding the use of some performance indicators, namely the hypervolume contribution and the hypervolume improvement.

Chapter 6: Theoretical analysis of Pareto fronts and Pareto sets of bi-objective convex-quadratic functions

In single-objective optimization, efficient evolution strategies converge linearly on strictly convex-quadratic functions [90, 105, 162]. Multiobjective strictly convex-quadratic problems are vector-valued functions where each coordinate is a strictly convex-quadratic function.

We believe that on these multiobjective problems, efficient multiobjective evolution strategies should at least converge linearly to a subset of the Pareto set, with an appropriate convergence definition. Therefore as a first step, we present in this chapter theoretical results that thoroughly analyze the global optima for bi-objective strictly convex and quadratic problems, $i.e.$ the Pareto set and the Pareto front. In particular if the two Hessians of the objective functions are proportional, the Pareto set is a line

segment between the two optima and the Pareto front is the graph of a function similar to $x \mapsto \left(1 - \sqrt{x}\right)^2$ on $[0, 1]$.

We derive various classes of convex-quadratic problems to test specific features of multiobjective algorithms. Namely the sensitivity with respect to separability, ill-conditioning, rotational invariance and Pareto set alignment with the coordinate axis.

The content of this chapter is presented in the EMO 2019 Conference at Michigan and published in [183]:
Cheikh Toure, Anne Auger, Dimo Brockhoff and Nikolaus Hansen, *On Bi-Objective convex-quadratic problems*, International Conference on Evolutionary Multi-Criterion Optimization, 2019, pp 3–14.

Chapter 7: Uncrowded Hypervolume Improvement: COMO-CMA-ES and the Sofomore framework

In state-of-the-art multiobjective evolutionary algorithms like SMS-EMOA [30] and MO-CMA-ES [104, 193], a particular trait is the use of the hypervolume contribution (or similarly the hypervolume improvement) as performance indicator. This helps to obtain a total order for the sorting of a population at each iteration. Therefore for a population P with non-dominated points P_1, the two-way ranking with the hypervolume contribution unflattens the fitness of the set of dominated points $P \setminus P_1$. However by observing the level sets of this new fitness, the dominated points of P typically steer towards regions already occupied by non-dominated points in P_1. Therefore this creates a crowdedness towards non-dominated points, and questions the quality assessment of this two-way ranking scheme. A particular characteristic of this two-way ranking is that the sorting of points within a non Pareto comparable subset P_i is independent of $P \setminus P_i$.

We propose in this chapter a fitness that removes the crowdedness defect observed in the two-way ranking scheme: it is called Uncrowded Hypervolume Improvement (UHVI). For a multiobjective function $\mathbf{f}$, the UHVI of a point with respect to a set P is its hypervolume improvement to P minus the distance of its image by $\mathbf{f}$ to the region dominating a so-called reference point and not dominated by any image by $\mathbf{f}$ of P. The UHVI definition is compliant with the Pareto dominance relation. It is used as a total order to sort any set P.

At each iteration during a multiobjective optimization, a single-objective fitness that depends on the population (set) to rank is defined. This idea is formalized into the Sofomore framework. It consists of using single-objective optimizers that are adapted based on the changing fitness obtained with the UHVI, in order to solve the overall multiobjective optimization problem that consists of finding the optimal distribution of p points maximizing the hypervolume.

COMO-CMA-ES is the multiobjective evolution strategy that we obtain by instantiating the Sofomore framework with the CMA-ES with recombination [88, 95]. Empirically, COMO-CMA-ES converges linearly towards the p-optimal distribution of the hypervolume, on various bi-objective convex quadratic problems. Some interesting features of the algorithm are also oberved, namely robustness to independently rotating randomly the Hessian matrices and bending the Pareto set.

COMO-CMA-ES is compared with MO-CMA-ES [104, 193], SMS-EMOA [30] and NSGA-II [56] on different classes of bi-objective convex quadratic problems. Globally COMO-CMA-ES performs better on two different metrics: the convergence gap that only depends on the incumbents of single-objective CMA-ES (the parents for the other algorithms), and the archive gap that takes into account all non-dominated solutions found so far by an algorithm. Although the evolution of COMO-CMA-ES does not depend on the archive gap, its well performance on that metric with respect to the other algorithms is explained by two different features. First the uncrowded spaces between two incumbents are more explored so that we should expect to sample more non-dominated points. Last, the particular non-elitist aspect of the CMA-ES with recombination [88,95] brings the large stationary variance typically observed with non-elitist evolution strategies.

The content of this chapter is presented in the GECCO 2019 Conference at Prague, in the EURO 2019 Conference at Dublin and in the MACODA 2019 Workshop at Leiden. It is published in [185]:
Cheikh Toure, Nikolaus Hansen, Anne Auger and Dimo Brockhoff, *Uncrowded hypervolume improvement: COMO-CMA-ES and the Sofomore framework*, Proceedings of the Genetic and Evolutionary Computation Conference, 2019, pp 638–646.

Chapter 8: Comparing COMO-CMA-ES to well-known multi-objective evolutionary algorithms

This chapter presents various benchmarks comparing three algorithms: NSGA-II [56], MO-CMA-ES [104, 193] and COMO-CMA-ES [185] introduced in Chapter 7. The benchmark is done with the COCO (COmparing Continuous Optimizers) platform [80] based on archive performance, $i.e.$ on all the points evaluated so far by the algorithm. The benchmarking platform contains classes of separable, ill-conditioned, multimodal and weak-structured problems. It allows to test the optimizers with respect to these features of optimization problems.

The well-known NSGA-II algorithm is used as a baseline, and various population sizes are considered for both MO-CMA-ES and COMO-CMA-ES algorithms. Although COMO-CMA-ES is designed to find an optimal distribution of p points on a Pareto set with respect to the hypervolume indicator, we observe in this chapter that it also performs well on COCO.

With these two CMA-ES multiobjective variants, we observe that the performance depends strongly on the fixed population size. For small budgets, algorithms with smaller population size perform better. For large budgets, the ones with larger population size reach better target precisions.

The content of this chapter is presented in the GECCO 2019 Workshop at Prague. It is published in [61]:
Paul Dufosse and Cheikh Toure, *Benchmarking MO-CMA-ES and COMO-CMA-ES on the Bi-objective bbob-biobj Testbed*, Proceedings of the Genetic and Evolutionary Computation Conference Companion, 2019, pp 1920–1927.

Chapter 9: Implementations

This chapter is devoted to the various implementations done during my PhD thesis. They are all based on the ask-and-tell paradigm [51]. On an optimization algorithm, this paradigm allows to separate the generation of candidate solutions and the update of the optimizer's state.

The Python implementation of COMO-CMA-ES [185] introduced in Chapter 7, is presented. It is the `pycomocma` package, that uses CMA-ES as single optimizers [95]. Similar to the ask-and-tell implementation of CMA-ES, the ask method of `pycomocma` generates candidate solutions for the multiobjective optimizer, and the tell method updates its state. In between, the candidate solutions are evaluated on a changing fitness given by the UHVI introduced in Chapter 7.

Finally, a Matlab implementation of COMO-CMA-ES [185] is presented. It is a Matlab interface where essentially the `pycomocma` Python package is used within, with continual conversion of data structures from Python to Matlab and vice versa. Due to Storengy demand, the interface has the same calling sequence as the Matlab implementations of MO-CMA-ES and SMS-EMOA [41, 186].

Introduction en français

Contents

L'optimisation multiobjective consiste à trouver les optima d'une fonction à valeurs vectorielles $\mathbf{f} : x \in \mathbb{R}^n \mapsto (\mathbf{f}_1(x), \ldots, \mathbf{f}_m(x)) \in \mathbb{R}^m$, où la relation d'ordre partielle considéré dans $\mathbb{R}^m$ dépend de f et est appelée "weakly Pareto dominance". L'ensemble des points non comparables dans $\mathbb{R}^n$ avec cette relation de Pareto est appelé l'ensemble de Pareto et son image par $\mathbf{f}$ est appelée le front de Pareto. Souvent cet ensemble a un nombre infini d'éléments. Avec un budget fini d'évaluations de fonctions ou un nombre fixé de solutions sur l'espace objectif $\mathbb{R}^m$ ou sur l'espace de recherche $\mathbb{R}^n$, un optimiseur multiobjectif vise à fournir des solutions d'une certaine qualité. Les indicateurs qui mesurent la qualité des solutions et guident l'optimisation sont appelés les indicateurs de qualité.

Dans un scénario de boîte noire d'ordre zéro, seule une séquence d'évaluations de fonctions est disponible pour l'optimiseur. Les informations relatives aux gradients ne sont pas disponibles. Cette situation est courante dans les cas d'utilisation industrielle, où une valeur de fonction est renvoyée après l'exécution d'un code de simulation complexe.

Les algorithmes de pointe pour résoudre les problèmes multiobjectifs de boîte noire d'ordre zéro sont les algorithmes évolutionnaires multiobjectifs, principalement grâce à leur nature basée sur la population qui permet de générer diverses solutions Pareto optimales en une seule exécution [49].

Les stratégies d'évolution sont une classe d'algorithmes d'optimisation aléatoire de type boîte noire [90, 159, 160, 172, 173], bien connues pour s'attaquer aux problèmes

d'optimisation du monde réel qui présentent souvent des difficultés telles que la non-linéarité, la non-convexité, la non-séparabilité, le mauvais conditionnement et la multimodalité. Pour un problème d'optimisation avec un domaine de définition continu, le CMA-ES [88, 95] fait partie des algorithmes d'optimisation mono-objective de boîte noire d'ordre zéro les plus avancés. Sur la base des comparaisons des valeurs de fonction observées, il apprend des informations de second ordre [91]. Une grande partie de ses paramètres internes est déjà définie lors de la conception de l'algorithme. Sa formulation générique, sa robustesse et ses propriétés d'invariance [96] sont parmi les raisons pour lesquelles le CMA-ES est utilisé dans une large gamme d'applications réelles, avec peu de réglages de paramètres nécessaires [37, 100, 166].

1 Contexte de la thèse

Ce travail est basé sur un projet de la société de stockage souterrain de gaz Storengy. Son fonctionnement consiste à stocker du gaz naturel dans un réservoir souterrain pendant l'été, lorsqu'il y a beaucoup de gaz disponibles (faible demande de gaz en période chaude), et à fournir le gaz pendant l'hiver, avec la forte demande de gaz en période froide. Storengy a besoin de prédire le comportement du réservoir afin d'optimiser son fonctionnement : un simulateur numérique de réservoir est utilisé à cette fin. De nombreux paramètres du simulateur liés à la porosité et à la perméabilité du réservoir doivent être mis en correspondance avec les mesures des puits : volume d'eau quotidien produit, volume d'eau annuel cumulé, pression de fond de puits pour les puits de terrain et position de l'interface eau-gaz. Deux ou trois fonctions objectives sont construites sur la base de ces mesures. Leur optimisation est primordiale pour calibrer les modèles afin d'obtenir un bon modèle prédictif.

Généralement, ces fonctions objectives sont multimodales, non convexes et non différentiables, et se trouvent dans un scénario de boîte noire d'ordre zéro en raison de l'absence de gradient et du caractère coûteux des valeurs de la fonction. Storengy est donc très intéressé par les optimiseurs qui peuvent traiter ce type d'irrégularités, en particulier par un optimiseur multiobjectif qui peut approcher le front de Pareto global malgré la multimodalité des fonctions objectives.

Le mécanisme de recombinaison est une caractéristique qui rend une stratégie d'évolution plus robuste aux irrégularités locales et à la multimodalité. Une stratégie d'évolution avec recombinaison échantillonne λ solutions candidates–la progéniture– en fonction de la solution favorite et d'autres paramètres, choisit les $\mu \leq \lambda$ meilleures parmi elles–les parents—, et recombine les parents pour obtenir la nouvelle solution favorite. Typiquement, $\mu \geq 2$ et $\lambda \approx 2\mu$.

Ce mécanisme permet également à la stratégie d'évolution de converger plus rapidement vers l'optimum global [13]. Dans diverses expériences, les stratégies

d'évolution avec recombinaison convergent ou divergent linéairement, $i.e.$ avec une vitesse de convergence géométrique, sur une large classe de fonctions [90].

Une classe intéressante de fonctions où la convergence ou la divergence linéaire (comportement linéaire) est souvent observée est la classe des fonctions scaling-invariant. Elle contient les normes, les fonctions convexes-quadratiques, les fonctions linéaires non triviales et plus généralement les transformations strictement croissantes de fonctions positivement homogènes. Les fonctions scaling-invariant sont définies comme préservant les f-comparaisons de points lorsque les points sont multipliés par le même nombre strictement positif (par rapport à un point de référence). Diverses questions se posent quant à l'étude de cette classe pour voir quelle est son étendue, par rapport aux transformations strictement croissantes des fonctions positivement homogènes. Nous pensons que la compréhension de cette classe de fonctions est une étape nécessaire pour prouver la convergence ou la divergence linéaire des stratégies d'évolution avec recombinaison sur cette classe. Les questions suivantes se posent naturellement :

Quelles sont les propriétés intéressantes des fonctions scaling-invariant ? Sont-elles liées aux fonctions positivement homogènes ?

(Chapitre 3)

La convergence linéaire est la convergence la plus rapide que l'on puisse obtenir dans les stratégies d'évolution [114, 182]. Elle est souvent observée de façon empirique [23, 90, 96, 105, 162]. Cependant, il est généralement difficile de la prouver. Le travail mathématique consiste souvent à analyser des conditions de drift sur des chaînes de Markov, ou à appliquer une loi des grands nombres à une chaîne de Markov ergodique sous-jacente. Dans la littérature, un comportement linéaire est théoriquement obtenu pour les stratégies d'évolution avec une population constituée d'un seul individu [2, 3, 19, 24, 48, 106–108, 110]. Mais, l'analyse de convergence linéaire des stratégies d'évolution avec un mécanisme de recombinaison est inexistante. Cela amène la question générale et difficile suivante.

Pouvons-nous prouver que les stratégies d'évolution avec recombinaison convergent ou divergent de façon géométrique sur les fonctions scaling-invariant, sous des hypothèses de régularité ?

(Chapitre 4)

Les questions ci-dessus concernent l'analyse théorique des stratégies d'évolution avec recombinaison. Cette dernière est cruciale pour divers problèmes mono-objectifs du monde réel qui sont irréguliers et multimodaux, comme les problèmes de Storengy. Nous pensons que dans un scénario multi-objectif, le mécanisme de recombinaison et la sélection non-élitiste peuvent aider à traiter les ensembles de Pareto locaux.

Pourtant, nous pensons qu'un optimiseur multiobjectif efficace doit au moins converger linéairement sur les problèmes multiobjectifs constitués de fonctions stricte-

ment convexes et quadratiques (avec une notion de convergence qui sera définie plus tard). Par conséquent, avant de concevoir un optimiseur multiobjectif basé sur des stratégies d'évolution avec des mécanismes de recombinaison et de non-élitisme, nous essayons de construire des classes de problèmes bi-objectifs à base de fonctions convexes et quadratiques, qui mettent en évidence différentes difficultés d'un problème, principalement la courbure du front de Pareto, la non-séparabilité et le mauvais conditionnement des fonctions objectives. Ceci nous amène à la question suivante.

Comment construire des classes de problèmes bi-objectifs convexes-quadratiques qui présentent des difficultés typiques ?

(Chapitre 6)

Le CMA-ES est l'une des stratégies d'évolution avec recombinaison les plus modernes. L'idée de concevoir un optimiseur multiobjectif basé sur le CMA-ES est apparue il y a plus de quinze ans. Pourtant, si nous avons plusieurs instances CMA-ES qui constituent un optimiseur multiobjectif, le nombre total d'évaluations de fonctions par itération est égal au nombre d'instances CMA-ES multiplié par la taille de la population des enfants λ. Ce nombre généralement élevé d'évaluations par itération rend difficile la création d'une version CMA-ES multiobjectif efficace. Dans [104], Igel et al. conçoivent une version du CMA-ES sans mécanisme de recombinaison et de non-élitisme, qu'ils ont remplacés par le mécanisme de sélection dit $(1 + 1)$ avec règle de succès $1/5$ ou un schéma de sélection $(\mu + \lambda)$ [121, 159]. Il s'agit d'une version élitiste de CMA-ES avec des parents μ et des enfants λ, où les prochains parents μ sont les meilleurs parmi les parents et les enfants $(\mu + \lambda)$. Sur la base de cette nouvelle version, Igel et al. conçoivent un optimiseur CMA-ES multiobjectif appelé MO-CMA-ES [104, 193].

Pourtant, le CMA-ES est moins enclin à rester coincé dans des optima locaux sous-optimaux, comparé à ses variantes élitistes [104]. Par conséquent, le besoin des industriels de disposer d'un optimiseur multiobjectif efficace basé sur le CMA-ES, et l'intérêt scientifique de voir les avantages de la recombinaison et du non-élitisme en optimisation multiobjective, sont toujours présents. Ceci nous guide à poser la question suivante.

Peut-on concevoir un algorithme multiobjectif robuste basé sur le CMA-ES (non-élitiste avec recombinaison), convergeant linéairement sur des problèmes multiobjectifs strictement convexes-quadratiques ? Comment se comporte-t-il par rapport aux algorithmes multiobjectifs existants ?

(Chapitres 7 et 8)

Le package pycma [89] pour le CMA-ES (avec recombinaison) est activement maintenu et est mis à jour régulièrement. Par conséquent, si un optimiseur multiobjectif est basé sur le CMA-ES, il est crucial de le rendre aussi compatible que possible avec le package pycma. De cette façon, une amélioration de l'algorithme

mono-objectif sera facilement transférée à la version multiobjective.

En outre, à des fins industrielles avec notre partenaire Storengy, l'algorithme multiobjectif conçu doit être implémenté dans Matlab. Il devrait également répondre aux besoins standards des cas d'utilisation industrielle, tels que la possibilité de paralléliser l'optimiseur. Une question naturelle se pose donc :

Comment implémenter efficacement les algorithmes multiobjectifs conçus à la fois en Python et en Matlab ? L'algorithme peut-il avoir une version parallélisée robuste ?

(Chapitre 9)

2 Les contributions de la thèse

Dans cette thèse, nous étudions les questions de recherche posées ci-dessus. Nous y répondons dans une certaine mesure et développons divers résultats innovants.

Nous étudions pour la première fois l'analyse de convergence linéaire de stratégies d'évolution avec recombinaison. Les algorithmes considérés sont des stratégies d'évolution représentées par un processus stochastique $\{(X_k, \sigma_k)\,; k \in \mathbb{N}\}$, où X_k est la solution favorite et σ_k est le step-size à l'itération k. Nous analysons le comportement linéaire de ces stratégies d'évolution sur des fonctions scaling-invariant.

En optimisation multiobjective, nous concevons un cadre d'optimiseurs multiobjectifs appelé Sofomore. Nous instancions ce cadre avec CMA-ES comme optimiseurs mono-objectifs pour obtenir le nouvellement créé COMO-CMA-ES, qui converge géométriquement vite sur des fonctions quadratiques strictement convexes. Nous réalisons l'implémentation de Sofomore et de COMO-CMA-ES, en Python et en Matlab. Le code est compatible avec le package pycma et dispose de diverses options utiles pour un usage industriel.

2.1 Convergence linéaire des stratégies d'évolution avec recombinaison sur des fonctions scaling-invariant

Dans le Chapitre 3, nous analysons en détail les fonctions scaling-invariant et présentons que sous certaines conditions, elles sont des composées de fonctions strictement monotones avec des fonctions positivement homogènes.

Plus précisément, nous donnons les conditions nécessaires et suffisantes pour qu'une fonction scaling-invariant soit une composée de fonction strictement monotone avec une fonction positivement homogène. Nous présentons également diverses propriétés satisfaites par les sublevel sets des fonction scaling-invariant. De manière surprenante, nous mettons en évidence que les fonctions scaling-invariant peuvent avoir

des comportements hautement pathologiques, en montrant des fonctions scaling-invariant à valeurs réelles qui ne sont monotones sur aucun intervalle non trivial. Les résultats du chapitre 3 sont publiés dans le Journal of Optimization Theory and Applications (JOTA) et sont présentés dans [184]:
Cheikh Toure, Armand Gissler, Anne Auger and Nikolaus Hansen, *Scaling-invariant functions versus positively homogeneous functions*, Journal of Optimization Theory and Applications, 2021.

Nous présentons dans le Chapitre 4 les comportements linéaires d'une classe de stratégies d'évolution avec mécanisme de recombinaison, sur une large classe de fonctions qui sont les fonctions scaling-invariant. Notre analyse inclut deux mécanismes d'adaptation de step-size bien connus, dérivés de l'Exponential Natural Evolution Strategy (xNES) [70] et du Cumulative Step-size Adaptation (CSA) sans cumul, où le CSA avec cumul est le mécanisme d'adaptation de step-size par défaut du CMA-ES [88].

Nous montrons que le logarithme de la distance du candidat favori à l'optimum (ou à un point de référence si l'algorithme diverge) divisé par le nombre d'itérations converge vers une limite r. De plus, le logarithme du step-size divisé par le nombre d'itérations converge vers la même limite r. Cette limite est le taux de convergence ou de divergence. Nous prouvons également que l'espérance de la variation du logarithme du step-size et l'espérance du log-progress convergent tous deux vers le taux r.

Le taux est exprimé comme une espérance sous une mesure de probabilité invariante inconnue qui provient des propriétés de stabilité d'une chaîne de Markov conjointe définie sur un espace d'états abstrait. Par conséquent, nous donnons également un théorème central limite pour estimer ce taux par des méthodes de Monte-Carlo.

Notre principale condition pour obtenir un comportement linéaire est que le logarithme du step-size doit croître strictement en espérance sur des fonctions linéaires non triviales. Il s'agit de la condition la plus stricte que nous puissions avoir, car elle équivaut à dire que l'algorithme diverge géométriquement sur les fonctions linéaires non triviales. Notons également la nouveauté de cette condition par rapport aux autres conditions de convergence linéaire que nous pouvons trouver dans la littérature [3, 19, 24].

Nous montrons également que cette condition est pratique pour des stratégies d'évolution concrètes comme xNES [70] sans adaptation de la matrice de covariance ou CSA-ES sans cumul [88], en l'exprimant par rapport aux paramètres des algorithmes et aux ordres statistiques de lois normales standards.

Les comportements linéaires du Chapitre 4 sont obtenus sur une large classe de fonctions, qui sont des transformations strictement croissantes soit de fonctions scaling-invariant C^1 avec un argmin global unique, soit de fonctions linéaires non

triviales.

Tous les résultats du chapitre 4 sont soumis sous forme d'un article dans le Journal of Global Optimization, par les auteurs Cheikh Toure, Anne Auger et Nikolaus Hansen.

2.2 Optimisation multiobjective : UHVI, Sofomore, COMO-CMA-ES, Implémentations

Nous développons dans le chapitre 7 un cadre d'algorithmes d'optimisation multiobjective appelé Sofomore. Ce cadre permet d'optimiser une fonction mono-objective changeante par divers algorithmes mono-objectifs. Un candidat favori d'un optimiseur mono-objectif se rapproche d'un élément de l'ensemble de Pareto. Le nombre d'optimiseurs mono-objectifs est fixé et noté p. Le but de l'optimisation multiobjective que nous considérons est alors de maximiser la distribution dite p-optimale de l'indicateur d'hypervolume [16, 22].

Dans les études précédentes telles que SMS-EMOA [30] ou MO-CMA-ES [104, 193], l'indicateur de performance utilisé pour guider l'optimisation multiobjective tend à orienter les solutions dominées vers des régions déjà occupées par une solution non dominée. Nous introduisons alors un nouvel indicateur de performance appelé UHVI: Uncrowded Hypervolume Improvement. Il crée un biais de recherche vers le domaine non couvert de l'ensemble de Pareto.

Nous instancions le cadre Sofomore avec l'optimiseur mono-objectif CMA-ES [88, 95] pour obtenir l'algorithme COMO-CMA-ES. Il est doté d'un mécanisme de sélection non élitiste inhérent au CMA-ES, ce que n'a pas la version élitiste $(1 + 1)$-CMA-ES utilisée dans MO-CMA-ES [104, 193].

Le contenu du chapitre 7 a été présenté à la conférence GECCO 2019 à Prague, à la conférence EURO 2019 à Dublin et au workshop MACODA 2019 à Leiden. Il est publié dans [185]:
Cheikh Toure, Nikolaus Hansen, Anne Auger and Dimo Brockhoff, *Uncrowded hypervolume improvement: COMO-CMA-ES and the Sofomore framework*, Proceedings of the Genetic and Evolutionary Computation Conference, 2019, pp 638–646.

Nous pensons que les stratégies d'évolution en optimisation multiobjective efficaces doivent converger (avec une définition de convergence liée au front de Pareto) linéairement sur les problèmes multiobjectifs strictement convexes-quadratiques, comme le font les stratégies d'évolution efficaces en optimisation mono-objective pour les fonctions strictement convexes-quadratiques.

Nous présentons en détail dans le chapitre 6 l'analyse de problèmes bi-objectifs strictement convexes et quadratiques, et donnons diverses descriptions du front de Pareto et de l'ensemble de Pareto pour cette classe de problèmes.

Nous construisons également des classes de problèmes bi-objectifs convexes-quadratiques qui visent à tester les comportements d'un algorithme multiobjectif par rapport à différentes difficultés. L'analyse théorique du chapitre 6 est publiée dans [183]:
Cheikh Toure, Anne Auger, Dimo Brockhoff et Nikolaus Hansen, *On Bi-Objective convex-quadratic problems*, International Conference on Evolutionary Multi-Criterion Optimization, 2019, pp 3–14.

Dans le chapitre 8, nous évaluons la nouvelle stratégie d'évolution multiobjective COMO-CMA-ES dans la plateforme COCO (COmparing Continuous Optimizers). Cette dernière permet de comparer différents optimiseurs continus de type boîte noire par rapport à la séparabilité, le mauvais conditionnement, la multimodalité et la structure d'un problème.

Nous comparons COMO-CMA-ES aux optimiseurs multiobjectifs bien connus NSGA-II [56] et MO-CMA-ES [104, 193]. Bien que COMO-CMA-ES ne soit pas conçu pour fonctionner sur une métrique basée sur l'archive (l'ensemble des points non-dominés) telle que celle utilisée par COCO, nous observons de bonnes performances de COMO-CMA-ES sur COCO. Cela souligne la puissance des stratégies d'évolution avec recombinaison comme le CMA-ES (mono-objectif) sous-jacent. Cela souligne également l'utilité d'un indicateur de performance qui oriente l'espace d'exploration vers les régions non couvertes, comme le UHVI qui garantit la qualité des solutions non-dominées qui sont générées, lorsque les solutions favorites de l'optimiseur multi-objectif sont proches de l'ensemble de Pareto. Les résultats du chapitre 8 sont publiés dans [61]:
Paul Dufosse and Cheikh Toure, *Benchmarking MO-CMA-ES and COMO-CMA-ES on the Bi-objective bbob-biobj Testbed*, Proceedings of the Genetic and Evolutionary Computation Conference Companion, 2019, pp 1920–1927.

Dans le chapitre 9, nous présentons le package Python développé pour ce travail, à savoir le package `pycomocma` disponible sur ce lien : https://github.com/CMA-ES/pycomocma. Semblable au package python CMA-ES `pycma` [89], il est implémenté avec le paradigme dit "ask-and-tell". Ce dernier permet d'avoir accès à l'optimiseur à chaque itération entre la génération de nouvelles solutions avec la méthode `ask`, et la mise à jour de l'instance d'optimisation avec la méthode `tell`.

Ce package implémente le framework Sofomore, et donne la possibilité de dériver l'algorithme COMO-CMA-ES en instanciant le framework avec des instances CMA-ES.

Une interface Matlab de COMO-CMA-ES est également produite pour notre partenaire industriel Storengy. Ce code Matlab fonctionne pour les versions Matlab 2014b ou ultérieures, car il utilise la commande Matlab `py`. L'installation du package Python `pycomocma` est également nécessaire.

3 Vue globale des chapitres

Chapitre 1 : Optimisation de type boîte noire en milieu continu

Dans un scénario d'optimisation en milieu continu de type boîte noire d'ordre zéro, l'objectif est de minimiser une fonction $f : \mathbb{R}^n \to \mathbb{R}$ où seules les valeurs de f, c'est-à-dire les $f(x)$ pour une séquence de solutions candidates $x \in \mathbb{R}^n$, sont disponibles. Cette situation se présente lorsque seul un oracle retournant $f(x)$ pour un x donné est disponible. Par conséquent, le problème d'optimisation est généralement sans dérivée, car nous n'avons aucune information relative au gradient de f. L'optimisation continue concerne le problème d'optimisation où l'espace de recherche est continu.

Nous présentons des algorithmes déterministes d'optimisation continue de type boîte noire comme la méthode de Nelder-Mead et certaines méthodes de Trust-region. Nous présentons également des algorithmes aléatoires plus raffinés que la recherche aléatoire pure. Il s'agit du recuit simulé et des algorithmes évolutionnaires. Nous nous concentrons ensuite sur les stratégies d'évolution, avec une présentation détaillée du Cumulative Step-size Adaptation Evolution Strategy (CSA-ES) et du Covariance Matrix Adaptation Evolution Strategy (CMA-ES). Le point de vue de l'Information-Geometric Optimization est enfin donné pour intégrer le CMA-ES dans une classe plus large d'algorithmes d'optimisation.

Chapitre 2 : Convergence des stratégies d'évolution

L'analyse d'une stratégie d'évolution est importante pour évaluer sa pertinence. La recherche aléatoire pure converge, mais avec une vitesse de convergence si lente que ce type d'algorithme d'optimisation n'est généralement pas utilisé en pratique. Nous présentons dans ce chapitre quelques notions traditionnellement utilisées pour mesurer la vitesse de convergence d'une stratégie d'évolution. A savoir la convergence ou divergence linéaire presque sûrement, l'espérance du log-progress, le "progress rate", le "quality gain", l'espérance du "hitting time" et l'espérance du "running time".

Nous présentons également brièvement quelques méthodes utilisées pour établir les résultats de convergence. Les équations différentielles ordinaires (EDO) sont l'une d'entre elles, avec le flux de l'Information-Geometric Optimization qui peut être vu comme une EDO. Nous décrivons comment la loi des grands nombres est utilisée via l'analyse de la chaîne de Markov pour obtenir une convergence linéaire presque sûrement ou une convergence linéaire de l'espérance du log-progress. Nous présentons également brièvement une technique d'analyse de drift permettant d'obtenir des bornes pour l'espérance du "hitting time".

Chapitre 3 : Fonctions scaling-invariant versus fonctions positivement homogènes

Une fonction $f : \mathbb{R}^n \to \mathbb{R}$ est scaling-invariant par rapport à un point de référence $x^\star \in \mathbb{R}^n$ si pour tout $x, y \in \mathbb{R}^n$ et $\rho > 0$:

$$f(x^\star + x) \leq f(x^\star + y) \iff f(x^\star + \rho x) \leq f(x^\star + \rho y) \ .$$

Les fonctions positivement homogènes sont des fonctions scaling-invariant. Rappelons qu'une fonction $p : \mathbb{R}^n \to \mathbb{R}$ est positivement homogène avec un degré $\alpha > 0$ si pour tout $x \in \mathbb{R}^n$ et $\rho > 0$: $p(\rho\,(x - x^\star)) = \rho^\alpha p(x - x^\star)$. Les fonctions linéaires, les normes, les fonctions quadratiques sont positivement homogènes. Les composées de fonctions strictement monotones avec des fonctions positivement homogènes sont scaling-invariant.

Nous montrons dans ce chapitre les conditions nécessaires et suffisantes pour qu'une fonction scaling-invariant soit la composée d'une fonction strictement monotone avec une fonction positivement homogène. En particulier, nous montrons que les fonctions continues scaling-invariant sont les composées d'un homéomorphisme avec une fonction continue positivement homogène.

Nous étudions les sublevel sets des fonctions scaling-invariant en généralisant des propriétés bien connues des fonctions positivement homogènes. Nous montrons notamment que sous des conditions légères, une fonction scaling-invariant a un unique argmin global si et seulement si ses sublevel sets sont des ensembles compacts.

Le contenu de ce chapitre est publié dans le Journal of Optimization Theory and Applications (JOTA). Il est présenté dans [184]:
Cheikh Toure, Armand Gissler, Anne Auger and Nikolaus Hansen, *Scaling-invariant functions versus positively homogeneous functions*, Journal of Optimization Theory and Applications, 2021.

Chapitre 4 : Convergence linéaire de stratégies d'évolution avec une adaptation de step-size et un mécanisme de recombinaison sur des fonctions scaling-invariant

Nous analysons dans ce chapitre le comportement linéaire des stratégies d'évolution avec recombinaison, où seuls le step-size et la solution favorite sont adaptés. L'adaptation de la matrice de covariance et d'autres mécanismes raffinés ne sont pas considérés, de sorte que la matrice de covariance est toujours le carré du step-size multiplié par la matrice identité.

Nous présentons une mise à jour générique du step-size définie de manière à ce que les invariances aux transformations préservant l'ordre et l'angle soient con-

servées, ce qui est crucial pour la conception des stratégies d'évolution [83, 167]. Nous montrons que le CSA-ES sans cumul et le xNES sans adaptation de la matrice de covariance sont des cas particuliers de notre cadre.

Nous prouvons ensuite la divergence linéaire sur des transformations strictement croissantes de fonctions linéaires et le comportement linéaire sur des transformations strictement croissantes de fonctions scaling-invariant C^1 avec un unique argmin global. Ces fonctions incluent les fonctions non convexes et les fonctions non quasi-convexes. La condition principale pour prouver nos résultas est équivalente à la divergence géométrique du step-size sur les fonctions linéaires non triviales.

La méthode que nous utilisons est basée sur une technique de chaîne de Markov pour l'analyse de stratégies d'évolution. De façon plus spécifique, nous exploitons la méthodologie établie dans [25] en prouvant que la chaîne de Markov normalisée sous-jacente est ergodique.

Les principaux résultats du chapitre sont la convergence linéaire ou la divergence linéaire du step-size et de la distance à l'optimum (ou à un point de référence). Le taux pour le comportement linéaire dépend d'une mesure de probabilité inconnue qui provient de la chaîne de Markov sous-jacente. Un théorème central limite est ensuite donné pour approximer ce taux.

Le contenu de ce chapitre a été soumis dans le Journal of Global Optimization par Cheikh Toure, Anne Auger et Nikolaus Hansen.

Chapitre 5 : Les algorithmes évolutionnaires multiobjectifs

Nous présentons dans ce chapitre les notions de base utilisées pour définir l'optimisation d'une fonction à valeurs vectorielles $\mathbf{f} : x \in \mathbb{R}^n \longmapsto \mathbf{f}(x) = (\mathbf{f}_1(x), \ldots, \mathbf{f}_m(x)) \in \mathbb{R}^m$ où $m \geq 2$. Définir une relation d'ordre dans $\mathbb{R}^m$ qui généralise l'ordre standard $\leq$ dans $\mathbb{R}$ (pour l'optimisation mono-objectif) est en soi une tâche importante. Elle est abordée par les notions de dominance de Pareto faible qui créent un ordre partiel dans $\mathbb{R}^m$. Pour un ensemble fini et non vide P, le sous-ensemble des points qui ne sont pas Pareto-dominés sont les points non dominés de P, et nous disons qu'ils ont un rang de Pareto égal à 1. De manière récursive, l'ensemble P est partitionné en un nombre fini de sous-ensembles $P_1, \ldots, P_k$ où P_i a un rang de Pareto de i et est Pareto-dominé par P_{i-1}. Deux éléments quelconques d'un sous-ensemble P_i ne sont pas comparables au sens de Pareto. Ceci est un premier pas vers le classement des éléments de P.

Pour trier complètement P, on présente des ordres totaux dans $\mathbb{R}^m$ compatibles avec la dominance faible de Pareto. La construction de ces ordres est typiquement basée sur des indicateurs de qualité, comme le crowding distance et l'indicateur d'hypervolume. Pour un ensemble de points non comparables par Pareto, l'indicateur

de performance est une fonction à valeurs réelles, c'est le tri que nous utilisons dans ce contexte. Ainsi, étant donné une population finie non vide P, elle est d'abord partitionnée en $P_1, \ldots, P_k$ où les points dans P_i ont un rang de Pareto i. Ensuite, P_i est trié par rapport à un indicateur de performance, indépendamment de $P \setminus P_i$. Globalement, un point avec un rang de Pareto plus petit est toujours meilleur. Entre deux points ayant le même rang de Pareto, un point ayant une plus grande valeur d'indicateur de performance est toujours meilleur. Ce schéma est appelé le "two-way ranking" [104].

La contribution à l'hypervolume et l'amélioration de l'hypervolume sont des indicateurs de performance de l'indicateur d'hypervolume. Un aspect spécifique de ce dernier est sa propriété de stricte monotonie, $i.e.$ sa propriété de conserver la relation de dominance de Pareto. Cette caractéristique permet de transformer le problème de trouver p solutions sur l'ensemble de Pareto en un problème mono-objectif dans $\mathbb{R}^{n \times p}$ qui consiste à maximiser l'hypervolume (défini sur l'espace de recherche) par rapport au problème multiobjectif à optimiser [16, 22].

Nous présentons des algorithmes évolutionnaires multiobjectifs bien connus (la famille d'algorithmes NSGA [56, 178], SMS-EMOA [30] et MO-CMA-ES [104, 193]) et observons comment un "two-way ranking" sous-jacent est inhérent à la conception de ces algorithmes.

Enfin, nous soulignons certaines limites concernant l'utilisation de certains indicateurs de performance, à savoir la contribution à l'hypervolume et l'amélioration de l'hypervolume.

Chapitre 6 : Analyse théorique des fronts et ensembles de Pareto pour des problèmes bi-objectifs convexes-quadratiques

En optimisation mono-objective, les stratégies d'évolution efficaces convergent linéairement sur des fonctions strictement convexes-quadratiques [90, 105, 162]. Les problèmes multiobjectifs strictement convexes-quadratiques sont des fonctions à valeurs vectorielles où chaque coordonnée est une fonction strictement convexe-quadratique.

Nous pensons que sur ces problèmes multiobjectifs, les stratégies d'évolution multiobjectives efficaces devraient au moins converger linéairement vers un sous-ensemble de l'ensemble de Pareto, avec une notion de convergence appropriée. Par conséquent, dans un premier temps, nous présentons dans ce chapitre des résultats théoriques qui analysent en détail les optima globaux pour les problèmes bi-objectifs strictement convexes et quadratiques, c'est-à-dire l'ensemble de Pareto et le front de Pareto. En particulier, si les deux matrices hessiennes des fonctions objectives sont proportionnelles, l'ensemble de Pareto est un segment de droite entre les deux optima et le front de Pareto est le graphe d'une fonction similaire à $x \mapsto \left(1 - \sqrt{x}\right)^2$ sur $[0, 1]$.

Nous dérivons diverses classes de problèmes convexes-quadratiques pour tester des caractéristiques spécifiques des algorithmes multiobjectifs. En particulier, la sensibilité à la séparabilité, au mauvais conditionnement, à l'invariance rotationnelle et à l'alignement de l'ensemble de Pareto avec l'axe des coordonnées.

Le contenu de ce chapitre est présenté à la conférence EMO 2019 au Michigan et publié dans [183]:
Cheikh Toure, Anne Auger, Dimo Brockhoff et Nikolaus Hansen, *On Bi-Objective convex-quadratic problems*, International Conference on Evolutionary Multi-Criterion Optimization, 2019, pp 3–14.

Chapitre 7 : Uncrowded Hypervolume Improvement: COMO-CMA-ES et le cadre Sofomore

Dans les algorithmes évolutionnaires multiobjectifs de pointe comme SMS-EMOA [30] et MO-CMA-ES [104, 193], un trait particulier est l'utilisation de la contribution à l'hypervolume (ou de manière similaire l'amélioration de l'hypervolume) comme indicateur de performance. Cela permet d'obtenir un ordre total pour le tri d'une population à chaque itération. Ainsi, pour une population P avec des points non dominés P_1, le "two-way ranking" avec la contribution de l'hypervolume permet d'obtenir une fonction non constante pour évaluer l'ensemble des points dominés $P \setminus P_1$. Cependant, en observant les lignes de niveaux de cette nouvelle fonction d'évaluation, les points dominés de P se dirigent généralement vers des régions déjà occupées par des points non dominés dans P_1. Cela crée donc un encombrement vers les points non dominés, et remet en question l'évaluation de la qualité du "two-way ranking". Une caractéristique particulière du "two-way ranking" est que le tri des points dans un sous-ensemble non Pareto-comparable P_i est indépendant de $P \setminus P_i$.

Nous proposons dans ce chapitre une fonction d'évaluation qui supprime le défaut d'encombrement observé dans le schéma du "two-way ranking" : elle est appelée Uncrowded Hypervolume Improvement (UHVI). Pour un problème multiobjectif $\mathbf{f}$, l'UHVI d'un point par rapport à un ensemble P est son amélioration d'hypervolume par rapport à P, moins la distance de son image par $\mathbf{f}$ à la région dominant un point dit de référence et non dominée par aucune image par $\mathbf{f}$ de P. La définition de l'UHVI est conforme à la relation de dominance de Pareto. Elle est utilisée comme un ordre total pour trier tout l'ensemble P.

A chaque itération au cours d'une optimisation multiobjective, une fonction à valeurs réelles qui dépend de la population (ensemble) à classer est définie. Cette idée est formalisée dans le cadre que l'on appelle Sofomore. Il consiste à utiliser des optimiseurs mono-objectifs qui sont adaptés selon la fonction mono-objective changeante obtenue avec l'UHVI, afin de résoudre le problème global d'optimisation multi-objective qui consiste à trouver la distribution optimale de points p maximisant l'hypervolume.

COMO-CMA-ES est la stratégie d'évolution multiobjective que nous obtenons en instanciant Sofomore avec le CMA-ES muni de la recombinaison [88, 95]. De façon empirique, COMO-CMA-ES converge linéairement vers la distribution p-optimale de l'hypervolume, sur divers problèmes quadratiques convexes bi-objectifs. Certaines caractéristiques intéressantes de l'algorithme sont également observées, notamment la robustesse par rapport à une transformation qui courbe l'ensemble de Pareto, et celle par rapport à une rotation indépendante et aléatoire des matrices hessiennes.

COMO-CMA-ES est comparé à MO-CMA-ES [104,193], SMS-EMOA [30] et NSGA-II [56] sur différentes classes de problèmes quadratiques convexes bi-objectifs. Globalement, COMO-CMA-ES est plus performant sur deux mesures différentes : le "convergence gap" qui ne dépend que des solutions favorites des CMA-ES mono-objectifs (les parents pour les autres algorithmes), et l'"archive gap" qui prend en compte toutes les solutions non dominées trouvées jusqu'à présent par un algorithme. Bien que l'évolution de COMO-CMA-ES ne dépende pas de l'"archive gap", sa bonne performance sur cette métrique par rapport aux autres algorithmes s'explique par deux caractéristiques différentes. Tout d'abord, les espaces non occupés entre deux solutions favorites sont davantage explorés, de sorte que nous devrions nous attendre à générer davantage de points non dominés. Enfin, l'aspect non élitiste particulier du CMA-ES avec recombinaison [88, 95] apporte la grande variance stationnaire typiquement observée avec les stratégies d'évolution non élitistes.

Le contenu de ce chapitre est présenté à la conférence GECCO 2019 à Prague, à la conférence EURO 2019 à Dublin et au workshop MACODA 2019 à Leiden. Il est publié dans [185]:
Cheikh Toure, Nikolaus Hansen, Anne Auger and Dimo Brockhoff, *Uncrowded hypervolume improvement: COMO-CMA-ES and the Sofomore framework*, Proceedings of the Genetic and Evolutionary Computation Conference, 2019, pp 638–646.

Chapitre 8 : Comparaison de COMO-CMA-ES avec des algorithmes évolutionnaires multiobjectifs bien connus

Ce chapitre présente différents benchmarks comparant trois algorithmes : NSGA-II [56], MO-CMA-ES [104, 193] et COMO-CMA-ES [185] (introduit dans le chapitre 7). Les benchmarks sont réalisés avec la plateforme COCO (COmparing Continuous Optimizers) [80] sur la base des performances de l'archive, $i.e.$ sur tous les points évalués jusqu'à présent par l'algorithme. La plateforme de benchmarking contient des classes de problèmes séparables, mal conditionnés, multimodaux et faiblement structurés. Elle permet de tester les optimiseurs par rapport à ces caractéristiques des problèmes d'optimisation.

L'algorithme populaire NSGA-II est utilisé comme référence, et différentes tailles de population sont considérées pour les algorithmes MO-CMA-ES et COMO-CMA-

ES. Bien que COMO-CMA-ES soit conçu pour trouver une distribution optimale de p points sur un ensemble de Pareto par rapport à l'indicateur d'hypervolume, nous observons dans ce chapitre qu'il est également performant sur COCO.

Avec ces deux variantes multiobjectives du CMA-ES, nous observons que les performances dépendent fortement de la taille de la population fixée. Pour les petits budgets, les algorithmes avec une taille de population plus petite sont plus performants. Pour les budgets importants, les algorithmes avec une taille de population plus grande atteignent de meilleures précisions pour les targets fixés.

Le contenu de ce chapitre est présenté dans le workshop GECCO 2019 à Prague. Il est publié dans [61]:
Paul Dufosse et Cheikh Toure, *Benchmarking MO-CMA-ES et COMO-CMA-ES on the Bi-objective bbob-biobj Testbed*, Proceedings of the Genetic and Evolutionary Computation Conference Companion, 2019, pp 1920–1927.

Chapitre 9 : Implémentations

Ce chapitre est consacré aux différentes implémentations réalisées au cours de ma thèse de doctorat. Elles sont toutes basées sur le paradigme ask-and-tell [51]. Sur un algorithme d'optimisation, ce paradigme permet de séparer la génération de solutions candidates et la mise à jour de l'état de l'optimiseur.

L'implémentation Python de COMO-CMA-ES [185] (introduite dans le chapitre 7) est présentée. Il s'agit du package `pycomocma`, qui utilise les CMA-ES comme optimiseurs mono-objectifs [95]. D'une manière similaire à l'implémentation ask-and-tell des CMA-ES, la méthode ask de `pycomocma` génère des solutions candidates pour l'optimiseur multiobjectif, et la méthode tell met à jour son état. Entre-temps, les solutions candidates sont évaluées sur une fonction mono-objective changeante donnée par l'UHVI (introduit dans le chapitre 7).

Enfin, une implémentation Matlab de COMO-CMA-ES [185] est présentée. Il s'agit d'une interface Matlab dans laquelle on utilise essentiellement le package Python `pycomocma`, avec une conversion incessante des structures de données de Python à Matlab, vice versa. En raison d'une demande de Storengy, l'interface a le même "calling sequence" que les implémentations Matlab de MO-CMA-ES et de SMS-EMOA [41, 186].

Chapter 1

Continuous black-box optimization

Contents

We present optimizers that are iterative algorithms with abstract states $\{\theta_k = (X_k, s_k);\ k \in \mathbb{N}\}$, where $X_k \in \mathbb{R}^n$ is the favorite solution (or incumbent) and s_k is an endogenous parameter, such that the update of the state θ_k is either deterministic, or depends on points sampled from a probability distribution P_{θ_k}. We denote by Θ the set of all states. Then $\{P_\theta;\ \theta \in \Theta\}$ is the

family of probability distributions where the sampled points can be generated from.

In a black-box scenario, the function $f : \mathcal{X} \subset \mathbb{R}^n \longrightarrow \mathbb{R}$ is unknown and only the f-values of sampled points are available. It is also called a Derivative-Free Optimization (DFO) or a zero-order optimization problem.

We present in this chapter some deterministic derivative-free optimization algorithms such as Nelder-Mead methods and Trust region methods. Later, we present different randomized continuous black-box optimization algorithms such as Simulated Annealing and evolutionary algorithms. We give a detailed presentation of evolution strategies, with an emphasis on CMA-ES. Finally we introduce the Information-Geometric Optimization (IGO) framework with the natural gradient ascent, the IGO algorithm and the IGO flow. We also present how it embeds randomized continuous black-box optimization algorithms like the Natural Evolution Strategy and CMA-ES.

1.1 Deterministic continuous black-box optimization

Recently, there is a revival of interest related to derivative-free optimization with deterministic algorithms [53]. Those methods are widely known and often used to optimize functions without the information of the gradients. We give a brief view of those optimization algorithms by presenting in particular the often used Nelder-Mead method [146, 197] and the family of Trust-region methods [52].

1.1.1 The Nelder-Mead method

The Nelder-Mead method [146, 197] is a deterministic derivative-free optimization algorithm. It is represented at iteration k by $\theta_k = \left(X_k^1, \ldots, X_k^{n+1}\right)$ that are the vertices of a simplex, with $X_k^i \in \mathbb{R}^n$ for $i = 1, \ldots, n+1$.

The initial simplex is important and must be carefully chosen based on the problem being optimized: a too small initial simplex can lead to a local search that traps the algorithm away from the optimum. At iteration k, θ_k is sorted with respect to the f-values. We obtain $X_k^{1:n+1}, \ldots, X_k^{n+1:n+1}$ such that $f(X_k^{1:n+1}) \leq f(X_k^{2:n+1}) \cdots \leq f(X_k^{n+1:n+1})$.

Then we construct the centroid X_k^0 of the n smallest points with respect to their f-values: $X_k^0 = \dfrac{1}{n} \sum_{i=1}^{n} X_k^{i:n+1}$.

Based on the centroid, the reflected point X_k^r is defined as: $X_k^r = X_k^0 + \alpha\left(X_k^0 - X_k^{n+1:n+1}\right)$ with $\alpha > 0$. In the case where $f(X_k^{1:n+1}) \leq f(X_k^r) < f(X_k^{n:n+1})$, $i.e.$ when the reflected point is better than the second worst point, then $\theta_{k+1} = \left(X_k^{1:n+1}, X_k^{2:n+1}, \ldots, X_k^{n:n+1}, X_k^r\right)$.

If instead X_k^r is the point with the smallest f-value, $i.e.$ $f(X_k^r) < f(X_k^{1:n+1})$, then an expansion is set with the expanded point X_k^e defined as $X_k^e = X_k^0 + \gamma\left(X_k^r - X_k^0\right)$, where $\gamma > 1$.

If the expanded point is strictly smaller than the reflected point, $i.e.$ $f(X_k^e) < f(X_k^r)$, then $\theta_{k+1} = \left(X_k^{1:n+1}, X_k^{2:n+1}, \ldots, X_k^{n:n+1}, X_k^e\right)$. Otherwise we have $f(X_k^e) \geq f(X_k^r)$, and we set $\theta_{k+1} = \left(X_k^{1:n+1}, X_k^{2:n+1}, \ldots, X_k^{n:n+1}, X_k^r\right)$. And overall the algorithm is updated in the second case where $f(X_k^r) < f(X_k^{1:n+1})$.

In the last case where $f(X_k^r) \geq f(X_k^{n:n+1})$, the contraction step is triggered, with the contracted point X_k^c defined as $X_k^c = X_k^0 + \rho\left(X_k^{n+1:n+1} - X_k^0\right)$ with $0 < \rho < 0.5$. If the contracted point is strictly smaller than the worst point, $i.e.$ $f(X_k^c) < f(X_k^{n+1:n+1})$, then $\theta_{k+1} = \left(X_k^{1:n+1}, X_k^{2:n+1}, \ldots, X_k^{n:n+1}, X_k^c\right)$. Otherwise we reach the shrink step that works by only keeping the best point: $X_{k+1}^1 = X_k^{1:n+1}$, and replacing the other points as $X_{k+1}^i = X_k^1 + \sigma\left(X_k^i - X_k^1\right)$ for $i = 2, \ldots, n$, and σ often equal to $\frac{1}{2}$.

Note that the Nelder-Mead algorithm is a heuristic search method that can fail to converge, even for strictly convex functions [137]. An improved variant of the Nelder Mead algorithm is given in [103].

1.1.2 Trust-region methods

Trust-region methods [52, 177] are important numerical optimization methods that solve nonlinear programming problems. After choosing a favorite solution, a Trust-region method defines a region around it, in which a surrogate model approximating the objective function is built in order to find the next favorite solution within that region. Based on the quality of the model, the corresponding region can be expanded (if the model is suitable) or contracted (otherwise). Usually quadratic functions are used as models. For example in the Levenberg-Marquardt algorithm [142], the objective function is approximated at each iteration by a quadratic surface which is right after solved by a linear solver in order to update the favorite candidate.

The Powell methods [153–155, 157, 158], in particular the NEW Unconstraint Optimization Algorithm (NEWUOA) are state-of-the-art for the deterministic continuous black-box optimization. For instance NEWUOA interpolates a quadratic model using a specific number of points to interpolate a quadratic model [161].

1.2 Overview of randomized continuous black-box optimization

Randomized search algorithms are particularly interesting to solve difficult problems and to avoid being trapped in local optima (with the inherent randomness). A natural example is the Pure Random Search [202] where for all state θ_k, P_{θ_k} is the same probability distribution equals to P. While this algorithm is simple and converge to the global optimum if the support of P contains $\mathcal{X}$, it is also inefficient [46]. Therefore we present in the following more refined algorithms such as Simulated Annealing [35,57], evolutionary algorithms like genetic algorithms and evolution strategies [45, 90, 174], with an emphasis on a subclass of non-elitist evolution strategies [95].

1.2.1 Simulated annealing

The continuous optimization variant of the Simulated Annealing was introduced in [35, 57]. It can be defined with the following sequence $\{\theta_k = (X_k, T_k); k \in \mathbb{N}\}$ where $X_k \in \mathbb{R}^n$ represents the favorite solution and $T_k > 0$ is called a temperature and defined such that $\lim_{k\to\infty} T_k = 0$.

At iteration $k + 1$, a candidate solution y_k is generated from the probability distribution $P_{\theta_k} = P_{X_k}$. The new favorite solution x_{k+1} is then chosen as follows. If $f(y_k) < f(X_k)$ then $X_{k+1} = y_k$. Otherwise $X_{k+1} = y_k$ with probability $\exp\left(-\frac{f(y_k)-f(X_k)}{T_k}\right)$. This setting helps the optimizer not to be trapped in a local optimum, by allowing to choose y_k with a non-zero probability, even if $f(y_k) \geq f(X_k)$.

In [135], it is shown that for a continuous function $f : \mathcal{X} \to \mathbb{R}$ where $\mathcal{X}$ is compact, convex and full dimensional and f has a unique global optimum over $\mathcal{X}$, and for a family of probability distribution $\{P_x; x \in \mathcal{X}\}$ such that for all $x \in \mathcal{X}$ P_x is absolutely continuous with respect to the Lebesgue measure, modulo a lower bound condition on the sequence $(T_k \log(k))_{k\geq 2}$, the Simulated Annealing algorithm is arbitrarily close to the global optimum, asymptotically.

Note that the family of probability distributions of the Simulated Annealing algorithm, that derives from the Metropolis-Hasting algorithm [98, 138], can be replaced by newer type of distributions, namely those derived from Sequential Monte Carlo Samplers [58]. The latter is a class of flexible and general methods allowing to sample from distributions and estimate their normalizing constants.

1.2.2 Evolutionary algorithms

Evolutionary algorithms are inspired by biological evolution. Its iterative process is driven by the application of mutation, recombination and selection in populations of candidate solutions [26, 200]. A global scheme of an evolutionary algorithm is as follows. A population of μ parents is selected, with μ being a positive integer. Then the μ parents create λ candidate solutions called offspring, where λ is a positive number. The creation of an offspring is done through recombination techniques or mutation techniques [26]. For the final step that is the selection, only μ points among the offspring or among the parents and the offspring are selected. If the selection is done by using only the μ best points among the offspring, the evolutionary algorithm is said to be non-elitist and the selection is said to be a "comma-selection" and denoted (μ, λ)-selection. Otherwise if all the $\mu + \lambda$ points are used, then we say that the evolutionary algorithm is elitist, and the selection is called a "plus-selection" and denoted $(\mu + \lambda)$-selection.

A popular subclass of evolutionary algorithms is the class of genetic algorithms [72, 101]. Its recombination method is called crossover. A genetic algorithm in a continuous domain is introduced in [45].

1.2.3 Evolution strategies

Evolution strategies (ES) are a subclass of evolutionary algorithms, most commonly used in randomized black-box optimization in continuous search spaces [90, 159, 160, 172, 173]. They are particularly known for their robustness with respect to noisy problems such as several real-world continuous optimization problems [116].

We define by $\{\theta_k = (X_k, s_k); k \in \mathbb{N}\}$ the sequence of states of an evolution strategy, where X_k is the favorite solution at iteration k and s_k contains the control or endogenous strategy parameters. Mostly, P_{θ_k} is distributed according to the multivariate normal distribution $\mathcal{N}(X_k, C_k)$ where C_k is a matrix that solely depends on s_k. As in Section 1.2.2, a (μ, λ)-ES is a non-elitist ES and a $(\mu + \lambda)$-ES is an elitist ES. The recombination step plays a significant role in evolution strategies, and is therefore handled carefully. We can highlight two important recombination methods, for both "plus-selection" and "comma-selection":

- Intermediate recombination: an average of $\rho \leq \mu$ points out of μ parents are taken to generate offspring. It is denoted by $(\mu/\rho_I, \lambda)$-ES or $(\mu/\rho_I + \lambda)$-ES.
- Weighted multi-recombination: it generalizes the intermediate recombination by taking a weighted combination of $\rho \leq \mu$ points out of μ parents to generate offspring. It is denoted by $(\mu/\rho_w, \lambda)$-ES or $(\mu/\rho_w + \lambda)$-ES, depending on the elitism property of the ES.

For the sphere function in large dimension, comma-selection evolution strategies with optimal weighted multi-recombination [12] have nearly two times and a half faster convergence rate than the considered fast $(1+1)$-ES [13].

In the context of real-parameter black-box optimization, the non-elitist CMA-ES is less prone to getting stuck to sub-optimal local optima, compared to elitist variants of the CMA-ES [104]. Therefore these qualities invite us to adopt the $(\mu/\rho_w, \lambda)$-selection evolution strategies.

1.2.4 Cumulative Step-size Adaptation Evolution Strategy (CSA-ES)

The Cumulative step-size adaptation evolution strategy (CSA-ES) is a refined evolution strategy with weighted multi-recombination. The sequence of states of the algorithm is represented by $\{(X_k, \sigma_k, p_k); k \in \mathbb{N}\}$ where at iteration k, $X_k \in \mathbb{R}^n$ represents the incumbent (the favorite solution), σ_k is a positive scalar representing the step-size and p_k is the cumulative path [95].

We fix $(X_0, \sigma_0, p_0) \in \mathbb{R}^n \times (0, \infty) \times \mathbb{R}^n$ and consider an independent and identically distributed sequence of random input $u = \{u_k = (u_k^1, \ldots, u_k^\lambda); k \geq 1\}$ where for all k, $u_k = (u_k^1, \ldots, u_k^\lambda)$ is composed of λ independent random vectors following a standard multivariate normal distribution $\mathcal{N}(0, I_n)$ on $\mathbb{R}^n$. After obtaining (X_k, σ_k, p_k) at iteration k, the algorithm is updated by first sampling independently λ candidate solutions from a multivariate normal distribution $\mathcal{N}\left(X_k, \sigma_k^2 I_n\right)$ with mean vector X_k and covariance matrix $\sigma_k^2 I_n$ using the vector u_{k+1}, i.e., for $i = 1, \ldots, \lambda$,

$$X_{k+1}^i = X_k + \sigma_k u_{k+1}^i \quad . \tag{1.1}$$

Then the candidate solutions are evaluated on the objective function f to be minimized and the solutions are ranked according to their f-values. We then extract the permutation of the ranked candidate solutions, more precisely we denote $(1{:}\lambda, \ldots, \lambda{:}\lambda)$ the indices of the candidate solutions from the best to the worse, i.e. such that

$$f(X_{k+1}^{1:\lambda}) \leq \cdots \leq f(X_{k+1}^{\lambda:\lambda}) \quad .$$

To break possible ties in case of solutions with two equal rankings, we set the convention that if $i < j$ and $f(X_{k+1}^i) = f(X_{k+1}^j)$ then $i{:}\lambda < j{:}\lambda$. Those indices are also used for the vectors u_{k+1}^i in the following way:

$$X_{k+1}^{i:\lambda} = X_k + \sigma_k u_{k+1}^{i:\lambda}.$$

To update the mean vector X_k, we consider a weighted average of the $\mu \leq \lambda$ best solutions $\sum_{i=1}^{\mu} w_i X_{k+1}^{i:\lambda}$ where $w_1 \geq w_2 \cdots \geq w_\mu > 0$ and $\sum_{i=1}^{\mu} w_i = 1$. This latter quantity is situated in the convex hull of the μ best points.

The next incumbent X_{k+1} is constructed by combining X_k and $\sum_{i=1}^{\mu} w_i X_{k+1}^{i:\lambda}$ as

$$X_{k+1} = \left(1 - c_m \sum_{i=1}^{\mu} w_i\right) X_k + c_m \sum_{i=1}^{\mu} w_i X_{k+1}^{i:\lambda} \tag{1.2}$$

$$= X_k + c_m \sum_{i=1}^{\mu} w_i \left(X_{k+1}^{i:\lambda} - X_k\right) = X_k + c_m \sigma_k \sum_{i=1}^{\mu} w_i u_{k+1}^{i:\lambda} \tag{1.3}$$

where $0 < c_m \leq 1$ is a learning rate.

Before adapting the step-size, the cumulative path is updated via the following equation:

$$p_{k+1} = (1 - c_\sigma) p_k + \sqrt{c_\sigma (2 - c_\sigma)} c_m \sum_{i=1}^{\mu} \frac{w_i}{\|w\|} u_{k+1}^{i:\lambda}, \tag{1.4}$$

with $0 < c_\sigma < 1$. We say that $1/c_\sigma$ is the life span of the information contained in p_k, since after $1/c_\sigma$ generations p_k is multiplied by the factor $(1 - c_\sigma)^{1/c_\sigma}$ that approaches $\frac{1}{e} \approx 0.37$ when c_σ goes to 0. Usually the value of c_σ is between $1/\sqrt{n}$ and $1/n$. The constant $\sqrt{c_\sigma(2 - c_\sigma)}$ means that under random selection, if p_k is distributed according to the multivariate standard normal distribution, then p_{k+1} is also distributed according to the multivariate standard normal distribution. That way the length of the cumulative path $\|p_k\|$ can be compared to the length of $\|\mathcal{N}_n\|$, which is the expected length under random selection.

With that observation, the step-size is updated as follows:

$$\sigma_{k+1} = \sigma_k \exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\|p_{k+1}\|}{\mathbb{E}\|\mathcal{N}_n\|} - 1\right)\right), \tag{1.5}$$

where the damping parameter $d_\sigma \geq 1$ determines the possible change rate of the step-size. If as in [14], the square length of the cumulative path $\|p_k\|^2$ is compared to the square length of $\|\mathcal{N}_n\|^2$, which is equal to n in expectation, then the step-size is updated as follows:

$$\sigma_{k+1} = \sigma_k \exp\left(\frac{c_\sigma}{2d_\sigma}\left(\frac{\|p_{k+1}\|^2}{n} - 1\right)\right). \tag{1.6}$$

Although the cumulative step-size adaptation (CSA) is regarded as the first choice for step-size control in the $(\mu/\mu_w, \lambda)$-ES, it has some disadvantages [86]:

- In the case of very large noise levels, the target step-size becomes zero, while the optimal step-size is still positive [31].
- For large population sizes ($\lambda > 10n$), the original parameter setting seemed not to work properly [94]: the notion of tracking a long path history does not mate well with a population size large compared to the search space dimension. Even though there exists an improved setting that shortens the backward time horizon for the cumulation and performs well also with large population sizes [93].

- The expected value for the displacement of the population mean under random selection is required. For this reason, the principle axes of the search distribution are in demand, but they are more expensive to acquire than a simple matrix decomposition. The latter is necessary to sample a multivariate normal distribution with given covariance matrix.

For all these reasons, some alternatives are established in order to replace the CSA, although they also have their share of disadvantages. One of them is the Two-point step-size adaptation (TPA) that was introduced for backpropagation in [176] and later applied in Evolutionary Gradient Search [168].

These step-size adaptation mechanisms learn first order information by estimating the gradient, which is enough to optimize for example convex quadratic functions with a small condition number. But when the condition number becomes large, optimization algorithms that learn second order information are necessary for an effective approach of the optimum. One of the evolution strategy that learns second order information is the Covariance Matrix Adaptation Evolution Strategy (CMA-ES). It has a finer control on the probability distribution P_θ.

1.3 Covariance Matrix Adaptation Evolution Strategy (CMA-ES)

The CMA-ES is a randomized black-box optimizer and is introduced in [85, 93–95]. It is invariant against order-preserving transformations of the objective function value and in particular against rotation and translation of the search space. A CMA-ES state θ_k contains the favorite solution X_k, the step-size σ_k and also the so-called covariance matrix C_k adapted such that P_{θ_k} is the probability distribution of $\mathcal{N}(X_k, \sigma_k^2 C_k)$.

1.3.1 The CMA-ES

The sequence of states of a $(\mu/\mu_w, \lambda)$-CMA-ES instance is $\{(X_k, \sigma_k, C_k, p_k^c, p_k^\sigma); k \in \mathbb{N}\}$ where at iteration k, $X_k \in \mathbb{R}^n$ represents the incumbent (the mean, the favorite solution), σ_k is a positive scalar representing the step-size, C_k is the covariance matrix, p_k^c is the evolution path and p_k^σ is the conjugate evolution path.

We fix $(X_0, \sigma_0) \in \mathbb{R}^n \times (0, \infty)$, $C_0 = I_n$, $p_0^\sigma = 0$ and we fix an independent and identically distributed sequence of random input $u = \{u_k = (u_k^1, \ldots, u_k^\lambda); k \geq 1\}$ where for all k, $u_k = (u_k^1, \ldots, u_k^\lambda)$ is composed of λ independent random vectors following a standard multivariate normal distribution $\mathcal{N}(0, I_n)$ on $\mathbb{R}^n$. Given $\{(X_k, \sigma_k, C_k, p_k^c, p_k^\sigma)$, we consider the following iterative update at iteration $k + 1$.

First, we sample independently $\lambda \in \mathbb{N}$ candidate solutions from a multivariate normal distribution $\mathcal{N}\left(X_k, \sigma_k^2 C_k\right)$ with mean vector X_k and covariance matrix $\sigma_k^2 C_k$ using the vector u_{k+1}, i.e., for $i = 1, \ldots, \lambda$,

$$X_{k+1}^i = X_k + \sigma_k C_k^{\frac{1}{2}} u_{k+1}^i \quad . \tag{1.7}$$

Then as in Section 1.2.4, the candidate solutions are evaluated on f and ranked with respect to their f-values, with $(1{:}\lambda, \ldots, \lambda{:}\lambda)$ the indices of the candidate solutions from the best to the worse, i.e. such that

$$f(X_{k+1}^{1:\lambda}) \leq \cdots \leq f(X_{k+1}^{\lambda:\lambda}) \quad .$$

The same nomenclature for the vectors u_{k+1}^i is done, so that:

$$X_{k+1}^{i:\lambda} = X_k + \sigma_k C_k^{\frac{1}{2}} u_{k+1}^{i:\lambda}.$$

To update the mean vector X_k, we consider a weighted average of the $\mu \leq \lambda$ best solutions $\sum_{i=1}^{\mu} w_i X_{k+1}^{i:\lambda}$ where the positive (recombination) weights $(w_i)_{1 \leq i \leq \mu}$ verify $w_1 \geq w_2 \geq \cdots \geq w_\mu > 0$. With only positive weights summing to one, this latter quantity is situated in the convex hull of the μ best points.

The next incumbent X_{k+1} is constructed by combining X_k and $\sum_{i=1}^{\mu} w_i X_{k+1}^{i:\lambda}$ as

$$X_{k+1} = \left(1 - c_m \sum_{i=1}^{\mu} w_i\right) X_k + c_m \sum_{i=1}^{\mu} w_i X_{k+1}^{i:\lambda} \tag{1.8}$$

$$= X_k + c_m \sum_{i=1}^{\mu} w_i \left(X_{k+1}^{i:\lambda} - X_k\right) = X_k + c_m \sigma_k \sum_{i=1}^{\mu} w_i C_k^{\frac{1}{2}} u_{k+1}^{i:\lambda} \tag{1.9}$$

where $0 < c_m < 1$ is a learning rate, usually set to 1. This update moves the new mean vector towards the best solutions. Often in practice, c_m is set to $1/\sum_{i=1}^{\mu} w_i$ such that the new mean vector is then the weighted average of the μ best solutions.

The conjugate evolution path is updated as

$$p_{k+1}^{\sigma} = (1 - c_\sigma) p_k^{\sigma} + c_m \sqrt{c_\sigma (2 - c_\sigma)} \sum_{i=1}^{\mu} \frac{w_i}{\|w\|} u_{k+1}^{i:\lambda}, \tag{1.10}$$

with $0 < c_\sigma < 1$. We say that $\dfrac{1}{c_\sigma}$ is the backward time horizon of the conjugate evolution path.

The step-size is meticulously updated as [82]:

$$\sigma_{k+1} = \sigma_k \exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\|p_{k+1}^{\sigma}\|}{\mathbb{E}\|\mathcal{N}_n\|} - 1\right)\right), \tag{1.11}$$

where the damping parameter d_σ verifying $d_\sigma \approx 1$, scales the change magnitude of $\log \sigma_k$. In [14], $\|p^\sigma_{k+1}\|^2$ and the comparison to its expectation n are used for the update of the step-size:

$$\sigma_{k+1} = \sigma_k \exp\left(\frac{c_\sigma}{2d_\sigma}\left(\frac{\|p^\sigma_{k+1}\|^2}{n} - 1\right)\right). \tag{1.12}$$

The evolution path is updated as

$$p^c_{k+1} = (1 - c_\sigma)p^c_k + c_m\sqrt{c_c(2 - c_c)}\sum_{i=1}^{\mu}\frac{w_i}{\|w\|}C_k^{\frac{1}{2}}\, u^{i:\lambda}_{k+1}, \tag{1.13}$$

where $\frac{1}{c_c} \geq 1$ is the backward time horizon of the evolution path. And the covariance matrix is updated according to:

$$C_{k+1} = (1 - c_1 - c_\mu)C_k + c_1 p^c_{k+1}\left[p^c_{k+1}\right]^\top + c_\mu \sum_{i=1}^{\mu} w_i u^{i:\lambda}_{k+1}\left[u^{i:\lambda}_{k+1}\right]^T, \tag{1.14}$$

where $0 < c_1 < 1$ and $0 < c_\mu < 1 - c_1$. The so-called rank-one update corresponds to $p^c_{k+1}\left[p^c_{k+1}\right]^\top$, while $\sum_{i=1}^{\mu} w_i u^{i:\lambda}_{k+1}\left[u^{i:\lambda}_{k+1}\right]^T$ corresponds to the rank-μ update.

To obtain $C_k^{\frac{1}{2}}$, the spectral decomposition is used with $C_k = B_k D_k^2 B_k^\top$ where D_k is a diagonal matrix with positive eigenvalues and B_k is a rotation matrix. Therefore $C_k^{\frac{1}{2}} = B_k D_k B_k^\top$. The matrix $B_k^\top$ rotates the space such that the columns of B_k, $i.e.$ the principal axes of the distribution $\mathcal{N}(0, C_k)$, rotate into the coordinate axes.

For a convex-quadratic function, C_k is similar in the ideal case to the inverse Hessian matrix [87]. The $(\mu/\mu_w, \lambda)$-CMA-ES then learns second-order information, and is therefore a stochastic counterpart of quasi-newton methods [82].

A variant with Two-Point Step-Size Adaptation, instead of Cumulative Step-size Adaptation, is proposed in [86]. An adaptation variant with possible negative covariance matrix is proposed in [115]. Note that there exist also large-scale variants of CMA-ES [191].

1.3.2 An elitist variant of the CMA-ES: $(1 + \lambda)$-CMA-ES

In [104, 193], Igel et al. introduce an elitist variant of the CMA-ES for the purpose of designing an effective multiobjective evolution strategy. With this variant, the parent population size and the offspring population size can be chosen as small as one. We present in Section 5.2.3 the corresponding multiobjective algorithm that derives from it, that is the MO-CMA-ES.

The elitist $(1 + \lambda)$-CMA-ES is constructed by using the $(1 + \lambda)$-selection method of evolution strategies [32, 159, 172], and adding a covariance matrix adaptation. Each

individual is a 5-tuple $a = [x, \bar{p}_{succ}, \sigma, p_c, C]$ respectively representing the candidate solution $x \in \mathbb{R}^n$, an averaged success rate $\bar{p}_{succ} \in [0, 1]$, the global step-size $\sigma \in \mathbb{R}_+$, an evolution path $p_c \in \mathbb{R}^n$ and the covariance matrix $C \in \mathbb{R}^{n \times n}$.

Additionally, default parameters are set for the selection $(\lambda = 1)$, the step-size control $\left(d = 1 + \frac{n}{2\lambda},\ p_{succ}^{target} = \frac{1}{5 + \sqrt{\lambda}/2},\ c_p = \frac{\lambda p_{succ}^{target}}{2 + \lambda p_{succ}^{target}}\right)$ and the covariance matrix adaptation $\left(c_c = \frac{2}{n+2},\ c_{cov} = \frac{2}{n^2+6},\ p_{thresh} = 0.44\right)$. The step-size, the evolution path and the covariance matrix are updated according to Algorithm 1 and 2.

Algorithm 1 $\text{updateStepSize}\left(a = [x, \bar{p}_{succ}, \sigma, p_c, C], p_{succ}\right)$

$\bar{p}_{succ} \longleftarrow (1 - c_p)\bar{p}_{succ} + c_p p_{succ}$

$\sigma \longleftarrow \sigma \exp\left(\frac{1}{d}\frac{\bar{p}_{succ} - \bar{p}_{succ}^{target}}{1 - \bar{p}_{succ}^{target}}\right)$

Algorithm 2 $\text{updateCovariance}\left(a = [x, \bar{p}_{succ}, \sigma, p_c, C], x_{step} \in \mathbb{R}^n\right)$

if $\bar{p}_{succ} < p_{thresh}$ **then**
    $p_c \longleftarrow (1 - c_c)p_c + \sqrt{c_c(2 - c_c)}x_{step}$
    $C \longleftarrow (1 - c_{cov})\, C + c_{cov}\, p_c p_c^T$
else
    $p_c \longleftarrow (1 - c_p)p_c$
    $C \longleftarrow (1 - c_{cov})\, C + c_{cov}\, \left(p_c p_c^T + c_c(2 - c_c)\, C\right)$
end if

Then an iteration of the $(1 + \lambda)$-CMA-ES can be divided in three phases. First, λ new candidate solutions are sampled. Second, the step-size is updated with a success rate $p_{succ} = \frac{\lambda_{succ}}{\lambda}$ where λ_{succ} represents the number of new candidate solutions with smaller f-values than the parent. And finally if the chosen solution is different from the parent solution, the covariance matrix is adapted accordingly. Algorithm 3 presents these three parts in details.

Algorithm 3 $(1+\lambda)$-CMA-ES

Given: $g = 0$, initialize $a_{parent}^{(0)}$
while not stopping criterion **do**
  $a_{parent}^{(g+1)} \longleftarrow a_{parent}^{(g)}$
  for $k = 1, \ldots, \lambda$ **do**
    $x_k^{g+1} \sim \mathcal{N}\left(x_{parent}^{(g)}, {\sigma^{(g)}}^2 C^{(g)}\right)$
  end for
  $\text{updateStepSize}\left(a_{parent}^{(g+1)}, \dfrac{\lambda_{succ}^{(g+1)}}{\lambda}\right)$
  if $f\left(x_{1:\lambda}^{(g+1)}\right) \leq f\left(x_{parent}^{(g)}\right)$ **then**
    $x_{parent}^{(g+1)} \longleftarrow x_{1:\lambda}^{(g+1)}$
    $\text{updateCovariance}\left(a_{parent}^{(g+1)}, \dfrac{x_{parent}^{(g+1)} - x_{parent}^{(g)}}{\sigma_{parent}^{(g)}}\right)$
  end if
  $g \longleftarrow g + 1$
end while

1.3.3 The Information-Geometric Optimization viewpoint

The CMA-ES (with the same learning rate for the mean and the covariance matrix) belongs to another class of optimizers, called Information-geometric optimization algorithms [7, 70]. Instead of observing the search space $\mathbb{R}^n$ where the optimum belongs to, the viewpoint of the information-geometric optimization is to find at iteration k, the optimal state θ_k containing the favorite candidate solution X_k [149]. Therefore the search space of the optimization is Θ and the objective function to optimize is define as

$$\tilde{J}(\theta) = \int f(x) P_\theta(dx). \tag{1.15}$$

The optimum of $\tilde{J}$ is such that P_θ is concentrated to the global optimum of f. Usually, a finer and more complex objective function on Θ, invariant by strictly increasing transformation, is considered. It is defined for each state θ_k at iteration k as follows

$$J_{\theta_k}(\theta) = \int W_{\theta_k}^f(x) P_\theta(dx), \tag{1.16}$$

where $W_{\theta_k}^f : \mathbb{R}^n \to \mathbb{R}$ is defined based on a non-increasing function $w : [0,1] \to \mathbb{R}$ as follows. If $P_{x' \sim P_{\theta_k}}(f(x') = f(x)) = 0$, then $W_{\theta_k}^f(x) = w\left(P_{x' \sim P_{\theta_k}}(f(x') \leq f(x))\right)$.

Otherwise $W^f_{\theta_k}(x) = \dfrac{1}{P_{x'\sim P_{\theta_k}}(f(x') = f(x))} \displaystyle\int_{P_{x'\sim P_{\theta_k}}(f(x')<f(x))}^{P_{x'\sim P_{\theta_k}}(f(x')\leq f(x))} w(q)\mathrm{d}q.$

This objective function $W^f_{\theta_k}$ is preferred to f: its definition is invariant under strictly increasing transformations of f. Also in the case where P_θ is concentrated on the global minimum of f, then $J_{\theta_k}(\theta)$ is maximal. Therefore the problem is now to maximize J_{θ_k} at iteration k.

To do so, the notion of natural gradient ascent is used, where the gradient ascent operates in the space Θ [20, 149].

It is a gradient with respect to the Fisher metric and defined as

$$\widetilde{\nabla}_\theta = I^{-1}\frac{\partial}{\partial\theta}, \tag{1.17}$$

where I represents the Fisher matrix [11, 149], defined as:

$$I_{ij}(\theta) = \int \frac{\partial \log P_\theta(x)}{\partial\theta_i}\frac{\partial \log P_\theta(x)}{\partial\theta_j}P_\theta(\mathrm{d}x) \tag{1.18}$$

$$= -\int \frac{\partial^2 \log P_\theta(x)}{\partial\theta_i\partial\theta_j}P_\theta(\mathrm{d}x). \tag{1.19}$$

Maximizing J_{θ_k} with a natural gradient ascent is iteratively done as follows.

$$\theta_{k+\delta k} = \theta_k + \delta k\, \widetilde{\nabla} J_{\theta_k}(\theta)\Big|_{\theta=\theta_k}, \tag{1.20}$$

where δk is a time increment or step-size of the natural gradient ascent.

We also have that

$$\widetilde{\nabla}_\theta J_{\theta_k}(\theta)\Big|_{\theta=\theta_k} = \int W^f_{\theta_k}(x)\frac{\widetilde{\nabla}_\theta P_\theta(x)\Big|_{\theta=\theta_k}}{P_{\theta_k}(x)}P_{\theta_k}(x) \tag{1.21}$$

$$= \int W^f_{\theta_k}(x)\widetilde{\nabla}_\theta \log P_\theta(x)\Big|_{\theta=\theta_k} P_{\theta_k}(x)\mathrm{d}x \tag{1.22}$$

$$= I^{-1}(\theta_k)\int W^f_{\theta_k}(x)\frac{\partial \log P_\theta(x)}{\partial\theta}\Big|_{\theta=\theta_k} P_{\theta_k}(x)\mathrm{d}x. \tag{1.23}$$

Therefore the expression of the next state θ_{k+1} is as follows

$$\theta_{k+\delta k} = \theta_k + \delta k\, I^{-1}(\theta_k)\int W^f_{\theta_k}(x)\frac{\partial \log P_\theta(x)}{\partial\theta}\Big|_{\theta=\theta_k} P_{\theta_k}(\mathrm{d}x). \tag{1.24}$$

Now if we do a Monte-Carlo approximation of the integrand in (1.24) by generating λ candidate solutions $x_k^1, \dots, x_k^\lambda$ from the probability distribution P_{θ_k}, assuming that they have different f-values, we have that

$$w\left(P_{x' \sim P_{\theta_k}}(f(x') < f(x_k^i))\right) \approx w\left(\frac{\mathrm{rk}(x_k^i) + \frac{1}{2}}{\lambda}\right),$$

where $\mathrm{rk}(x_k^i)$ is the cardinal of $\{j | f(x_j) < f(x_i)\}$. By injecting this approximation in (1.24), we obtain the so-called Information-Geometric Optimization (IGO) algorithm iterated as:

$$\theta_{k+\delta k} = \theta_k + \delta k\, I^{-1}(\theta_k) \frac{1}{\lambda} \sum_{i=1}^{\lambda} w\left(\frac{\mathrm{rk}(x_k^i) + \frac{1}{2}}{\lambda}\right) \frac{\partial \log P_\theta(x)}{\partial \theta}\bigg|_{\theta=\theta_k}. \tag{1.25}$$

Another way to retrieve the IGO algorithm update is by setting

$$w_i = \frac{w\left((i - 1/2)/\lambda\right)}{\lambda} \tag{1.26}$$

and denoting $x_k^{i:\lambda}$ the th candidate solution with respect to their f-values (still assuming that the f-values are different): $f(x_k^{1:\lambda}) < \cdots < f(x_k^{\lambda:\lambda})$. The IGO algorithm update in (1.25) then becomes

$$\theta_{k+\delta k} = \theta_k + \delta k\, I^{-1}(\theta_k) \sum_{i=1}^{\lambda} w_i \frac{\partial \log P_\theta(x_k^{i:\lambda})}{\partial \theta}\bigg|_{\theta=\theta_k}. \tag{1.27}$$

We fall back to the CMA-ES context that we reparametrize as $\theta_k = (x_k, C_k)$ where the candidate solution x_k is the mean and C_k is the covariance matrix and P_{θ_k} the probability distribution of the multivariate normal distribution $\mathcal{N}(x_k, C_k)$. For all $x \in \mathbb{R}^n$ and $\theta \in \Theta$, $\log P_\theta(x) = -\frac{k}{2}\log(2\pi) - \frac{1}{2}\log det(C_k) - \frac{1}{2}\log\left((x - x_k)^\top C_k^{-1}(x - x_k)\right)$. Applying (1.27) to this family of probability distribution implies that

$$x_{k+\delta k} = x_k + \delta k \sum_{i=1}^{\lambda} w_i \left(x_k^{i:\lambda} - x_k\right), \tag{1.28}$$

$$C_{k+\delta k} = C_k + \delta k \sum_{i=1}^{\lambda} w_i \left((x_k^{i:\lambda} - x_k)(x_k^{i:\lambda} - x_k)^\top - C_k\right). \tag{1.29}$$

These equations define a variant of the CMA-ES where the learning rate for the mean and the covariance matrix are the same.

In [9], Akimoto et al. show that they can obtain under some types of parameterization of multivariate normal distribution the natural gradient of the expected fitness without the need for inversion of the Fisher information matrix.

An exponential parametrization of the covariance matrix allows to retrieve the so-called Exponential Natural Evolution Strategy, from the IGO algorithm [70].

Although the IGO framework gives a generalized view of evolution strategies and has a theoretical justification [10], it does not answer to the questions related to the convergence of its algorithms. However, there exists a continuous-time variant of (1.24) called IGO flow [149] that allows to state some convergence results. It is defined for all $t \geq 0$ as

$$\frac{\mathrm{d}\theta_t}{\mathrm{d}t} = I^{-1}(\theta_t) \int W^f_{\theta_t}(x) \frac{\partial \log P_\theta(x)}{\partial \theta}\bigg|_{\theta=\theta_t} P_{\theta_t}(\mathrm{d}x). \tag{1.30}$$

Hence in [4], convergence results are derived based on an ODE method with the underlying IGO flow, applied to the IGO algorithm for a multivariate standard normal distribution with covariance matrices equal to $\{\sigma_k I_n; k \geq 0\}$.

Chapter 2

Convergence of evolution strategies

Contents

The goal of a continuous optimization algorithm is essentially to get quickly as close as possible to the eventual global optimum. Typically, the optimum is not exactly reached, and therefore some methods are established to measure the performance of the algorithm, based on its behavior with respect to the global optimum.

For a sequence of states $\{\theta_k; k \geq 0\}$ with favorite solutions $\{X_k; k \geq 0\}$ and step-sizes $\{\sigma_k; k \geq 0\}$, it is important to measure the behavior of σ_k and $\|X_k - x^\star\|$ where $x^\star$ is the global optimum, with respect to the number of iterations k. In a black-box scenario, it is also interesting to quantify the evolution of σ_k and $\|X_k - x^\star\|$ with respect to the number of function evaluations.

In the following, we present some notions that measure the convergence speed of an optimization algorithm. We also highlight some methods used to analyze the convergence of evolution strategies, that are based on Markov chain theory and drift analysis.

2.1 Convergence and divergence speeds

We remind in the following different notions that measure the speed of convergence of the optimization algorithm. Namely convergence rate, divergence rate, expected log-progress, progress rate, quality gain, expected hitting time and expected running time.

2.1.1 Linear behaviors

If $\{\theta_k; k \geq 0\}$ represents the states of a deterministic algorithm whose favorite solutions are $\{X_k; k \geq 0\}$, we say that X_k converges linearly or geometrically towards $x^\star$ at rate $r > 0$ if

$$\lim_{k\to\infty} \log \frac{\|X_{k+1} - x^\star\|}{\|X_k - x^\star\|} = -r. \tag{2.1}$$

Typically, linear convergence is defined as $\lim_{k\to\infty} \frac{\|X_{k+1} - x^\star\|}{\|X_k - x^\star\|} = c$ with $0 < c < 1$, which is equivalent to (2.1).

With the Cesàro lemma, (2.1) implies that

$$\lim_{k\to\infty} \frac{1}{k} \log \|X_{k+1} - x^\star\| = -r. \tag{2.2}$$

With (2.2), we observe that asymptotically, the logarithm of the distance between the favorite solution at iteration k and the optimum decreases linearly in k like $-rk$.

In the case where $r < 0$ in (2.1), we say that X_k diverges linearly or geometrically when k goes towards ∞. For $r = \infty$ in (2.1), X_k is said to converge superlinearly to $x^\star$ when k tends towards ∞. In that case for $q \in (1, \infty)$, we say that X_k converges with order q to $x^\star$ at rate $r > 0$ if

$$\lim_{k\to\infty} \log \frac{\|X_{k+1} - x^\star\|}{\|X_k - x^\star\|^q} = -r \quad . \tag{2.3}$$

In the case where q is equal to 2, we particularly say that the convergence is quadratic.

Finally if $r = 0$ in (2.1), X_k converges sublinearly when k goes to ∞.

If $\{\theta_k; k \geq 0\}$ represents the states of a stochastic algorithm whose favorite solutions are random vectors $\{X_k; k \geq 0\}$, we similarly say that X_k converges linearly or geometrically towards $x^\star$ at rate $r > 0$ if $\lim_{k\to\infty} \frac{1}{k} \log \frac{\|X_k - x^\star\|}{\|X_0 - x^\star\|} = -r$ almost surely. In the same way, X_k diverges linearly or geometrically if $\lim_{k\to\infty} \frac{1}{k} \log \frac{\|X_k - x^\star\|}{\|X_0 - x^\star\|} = r > 0$ almost surely.

It is also useful to analyze the behavior of other elements of a state θ_k. For example in evolution strategies like the CSA-ES algorithm in Section 1.2.4, the evolution of the step-size σ_k of θ_k is also analyzed. We say that σ_k converges linearly or geometrically towards 0 at rate $r > 0$ if $\lim_{k\to\infty} \frac{1}{k} \log \frac{\sigma_k}{\sigma_0} = -r$ almost surely. In the same way, σ_k diverges linearly or geometrically if $\lim_{k\to\infty} \frac{1}{k} \log \frac{\sigma_k}{\sigma_0} = r > 0$ almost surely.

2.1.2 Expected progress and expected stopping times

The expected log-progress is investigated for linear behaviors of evolution strategies [25]. For a step-size evolution strategy $\{\theta_k = (X_k, \sigma_k) ; k \in \mathbb{N}\}$ with initial condition $(X_0, \sigma_0) = (x, \sigma)$, we have a linear behavior of the expected log-progress of the favorite solution if $\lim_{k\to\infty} \mathbb{E}_{\frac{x-x^\star}{\sigma}} \left[\log \frac{\|X_{k+1} - x^\star\|}{\|X_k - x^\star\|} \right] = r$ where $r \in \mathbb{R}$. The expected log-progress of the step-size converges or diverges linearly if $\lim_{k\to\infty} \mathbb{E}_{\frac{x-x^\star}{\sigma}} \left[\log \frac{\sigma_{k+1}}{\sigma_k} \right] = r \in \mathbb{R}$.

Typically in evolution strategies, the explicit formula of the convergence or divergence rate is difficult to obtain. The expected amount of the optimization algorithm in one iteration is often considered. The progress rate is a common measure for such progress in term of the search space, and the quality gain is also used and expressed in term of function values.

The progress rate is introduced in [90, 159]. It measures the reduction of the distance to the optimum in one iteration of the optimization algorithm. The normalized progress rate is defined as the expected relative reduction of $\|x_k - x^\star\|$ where $x^\star$ is the optimum. It is explicitely defined as

$$\varphi_k^\star = n\, \mathbb{E}\left[\frac{\|X_k - x^\star\| - \|X_{k+1} - x^\star\|}{\|X_k - x^\star\|} \middle| \theta_k \right] = n \left(1 - \mathbb{E}\left[\frac{\|X_{k+1} - x^\star\|}{\|X_k - x^\star\|} \middle| \theta_k \right] \right) . \tag{2.4}$$

The quality gain is a measure similar to the progress rate, expressed in term of function values [6]. It is defined as the expected relative decrease of the function value. For a step-size evolution strategy $\{\theta_k = (X_k, \sigma_k) ; k \in \mathbb{N}\}$, if $x^\star$ is a global minimum of a function $f : \mathbb{R}^n \to \mathbb{R}$, $x \in \mathbb{R}^n$ not a global minimum and $\sigma > 0$ as

$$\Phi(x, \sigma) = \frac{\mathbb{E}\left[f(X_k) - f(X_{k+1}) \middle| X_k = x, \sigma_k = \sigma \right]}{f(x) - f(x^\star)} . \tag{2.5}$$

Another way to analyze the convergence speed of an optimization algorithm is by analyzing specific stopping times, that are the expected hitting time and the expected running time.

For a stochastic process $\{X_k \in \mathbb{R}^n; k \geq 0\}$, the first hitting time of the ball $\mathbf{B}(x^\star, \epsilon)$ where $x \in \mathbb{R}^n$ and $\epsilon > 0$ is the stopping time $\tau_\epsilon = \inf\{k \in \mathbb{N}; X_k \in \mathbf{B}(x^\star, \epsilon)\}$. The expected hitting time is defined as the expectation of the first hitting time, $i.e.$ as $\mathbb{E}[\tau_\epsilon]$. The expected hitting time $\mathbb{E}[\tau_\epsilon]$ is related to the notion of linear convergence at rate r if $-\mathbb{E}[\tau_\epsilon] / \log(\epsilon) \simeq 1/r$ when $\epsilon \to 0$ [6].

The runtime is similar to the hitting time, expressed in term of function values [46]. For a function $f : \mathbb{R}^n \to \mathbb{R}$ with global optimum $x^\star$, the running time to a precision $\epsilon > 0$ is defined as $\eta_\epsilon = \inf\{k \in \mathbb{N}; |f(X_k) - f(x^\star)| < \epsilon\}$. The expected running time is therefore defined as $\mathbb{E}[\eta_\epsilon]$.

Note that evolution strategies are typically invariant to strictly increasing transformations of the optimized function f. However this notion of running time does not reflect that invariance property. For a strictly increasing function g, the smallest iteration k_0 such that $|f(X_{k_0}) - f(x^\star)| \leq \epsilon$ is typically not equal to the smallest iteration k_0 such that $|g(f(X_{k_0})) - g(f(x^\star))| \leq \epsilon$. For this reason, the spatial suboptimality function is introduced in [3, 69]. For a function $f : \mathbb{R}^n \to \mathbb{R}$ measurable with respect to the Borel σ-algebra of $\mathbb{R}^n$, the spatial suboptimality function $f_{\mu^{Leb}}$ is defined for $x \in \mathbb{R}^n$ as

$$f_{\mu^{Leb}}(x) = \sqrt[n]{\mu^{Leb}\left(f^{-1}\left((-\infty, f(x)]\right)\right)} = \sqrt[n]{\mu^{Leb}\left(\mathcal{L}^{\leq}_{f,x}\right)} \ .$$

For a measurable function f and a strictly increasing function $g : \mathrm{Im} f \to \mathbb{R}$, f and $g \circ f$ have the same spatial suboptimality function equals to $f_{\mu^{Leb}}$, so that the running time in terms of f or $g \circ f$ are the same. This conserves the invariance under order-preserving transformations inherent to the design of evolution strategies [83, 167].

2.2 Theoretical analysis of evolution strategies

Analyzing evolution strategies is important to state whether or not a designed one is relevant. Especially knowing the convergence mode allows to separate practical algorithms from others.

For example with Pure random search, the slow expected hitting time $\mathbb{E}[\tau_\epsilon]$ has an order of $1/\epsilon^n$. Similarly, elitist evolution strategies with a step-size effectively bounded from above and below (also non-elitist evolution strategies when in addition the search space is bounded) converge with an expected hitting time $\mathbb{E}[\tau_\epsilon]$ of the order of $1/\epsilon^n$, under mild conditions on the objective function [18, 165]. An optimization algorithm with such slow speed of convergence is therefore not practically relevant.

In the contrary, comparison-based optimization algorithms with a bounded number of comparisons between function values have an expected hitting time $\mathbb{E}[\tau_\epsilon]$ lower bounded by a constant times $-n \log(\epsilon)$ when $\epsilon \to 0$ [68, 182]. Therefore it is interesting to find an algorithm reaching this lower bound, on a wide class of functions.

We present in the following different techniques often used to analyze the convergence of evolution strategies.

2.2.1 Ordinary differential equation methods

Ordinary Differential Equation (ODE) methods are commonly used to investigate the convergence of stochastic optimization algorithms [27,28,36,131,134]. ODE methods are applied on optimization algorithms whose state $\theta_k \in \Theta$ at iteration k verifies

$$\theta_{k+1} = \theta_k + \alpha_k F_{k+1} \tag{2.6}$$

where $\alpha_k > 0$ and F_{k+1} is a random state in Θ such that $\mathbb{E}\left[F_{k+1}|\theta_k\right] = \mathbb{E}\left[F_{k+1}|\mathscr{F}_k\right]$, which is the conditional expectation given the natural filtration $\{\mathscr{F}_k; k \geq 0\}$ associated to $\{\theta_k; k \geq 0\}$. The positive value α_k represents a step-size that can be seen as a learning rate.

The ODE method makes the link between the stochastic optimization algorithm in (2.6) and the solutions of the corresponding ODE

$$\frac{\mathrm{d}\theta}{\mathrm{d}t} = F(\theta),\ \theta(0) = \theta_0$$

where F is a suitable function on Θ verifying $F(\theta_k) = \mathbb{E}\left[F_{k+1}|\theta_k\right]$.

An ODE method to prove the geometric convergence of adaptive stochastic algorithms is introduced in [5]. The latter method tackles stochastic algorithms that come from optimization methods solving deterministic optimization problems. In particular some derivative-free optimization algorithms induced by stochastic approximation algorithms with a constant step-size.

This framework of ODE methods is also applied to the IGO flow introduced in Section 1.3.3. In [4], the IGO flow for multivariate standard normal distributions with covariance matrices $\{\sigma_k I_n; k \geq 0\}$ converges locally on twice differentiable functions to critical points of the objective function, under mild conditions.

2.2.2 Law of large numbers applied to ergodic Markov chains

Markov chain theory is used to study almost sure linear convergence and linear convergence of the expected log-progress of evolution strategies [19, 24, 25, 116, 117, 117]. The main idea is introduced in [33] for the analysis of a self-adaptation evolution strategy on the sphere function. It goes as follows, for a step-size adaptive evolution strategy $\{(X_k, \sigma_k)\,; k \in \mathbb{N}\}$.

Assume that f is a function with global argmin $x^\star$ and $\{Z_k = (X_k - x^\star)/\sigma_k \,; k \in \mathbb{N}\}$ is a homogenous Markov chain. Then we have the equation

$$\log \frac{\|X_{k+1} - x^\star\|}{\|X_k - x^\star\|} = \log \frac{\|Z_{k+1}\|}{\|Z_k\|} + \log\left(\frac{\sigma_{k+1}}{\sigma_k}\right) \quad .$$

If we take the expectation under $Z_0 = z$, then

$$\mathbb{E}_z\left[\log \frac{\|X_{k+1} - x^\star\|}{\|X_k - x^\star\|}\right] = \mathbb{E}_z\left[\log \frac{\|Z_{k+1}\|}{\|Z_k\|}\right] + \mathbb{E}_z\left[\log \frac{\sigma_{k+1}}{\sigma_k}\right] \quad .$$

Therefore if $\{Z_k \,; k \in \mathbb{N}\}$ is an ergodic homogeneous Markov chain with invariant probability π such that $z \mapsto \log \|z\|$ is π-integrable, then $\mathbb{E}_z\left[\log \frac{\|Z_{k+1}\|}{\|Z_k\|}\right]$ converges to 0 when k goes to ∞, so that $\mathbb{E}_z\left[\log \frac{\|X_{k+1} - x^\star\|}{\|X_k - x^\star\|}\right]$ and $\mathbb{E}_z\left[\log \frac{\sigma_{k+1}}{\sigma_k}\right]$ have the same limit if they converge. Further considerations on the algorithm and the function imply such convergence.

Similarly,

$$\frac{1}{k} \log \frac{\|X_k - x^\star\|}{\|X_0 - x^\star\|} = \frac{1}{k} \sum_{t=0}^{k-1} \log \frac{\|X_{t+1} - x^\star\|}{\|X_t - x^\star\|} = \frac{1}{k} \sum_{t=0}^{k-1} \log \frac{\|Z_{t+1}\|}{\|Z_t\|} + \frac{1}{k} \sum_{t=0}^{k-1} \log\left(\frac{\sigma_{t+1}}{\sigma_t}\right).$$

The latter equation suggests that if we can apply a Law of Large Numbers to the chain $\{Z_k \,; k \in \mathbb{N}\}$ then the term $\frac{1}{k} \sum_{t=0}^{k-1} \log \frac{\|Z_{t+1}\|}{\|Z_t\|}$ converges to 0, so that the step-size and the favorite solution have the same rate, if they converge or diverge.

This methodology is used in [19] and generalized in [25] for scaling-invariant functions. We use it in Chapter 4 to show global linear convergence of a subclass of step-size adaptive evolution strategies, on a subclass of scaling-invariant functions.

2.2.3 Bounds on the expected hitting time via drift analysis

Drift analysis on a stochastic process $\{X_k; k \in \mathbb{N}\}$ adapted to a natural filtration $\{\mathscr{F}_k; k \in \mathbb{N}\}$ consists in analyzing properties of the corresponding drift defined as $\mathbb{E}[V(X_{k+1})|\mathscr{F}_k] - V(X_k)$ where V is a real-valued function that is typically called a Lyapunov drift function. To imply bounds on the expected hitting time, the main idea relies on finding a Lyapunov function with lower and upper bounded expected drift [77].

Based on this idea, in [3, 143, 144] a potential function is constructed so that the upper bound of the expected hitting time of the $(1+1)$-ES with $1/5$-success rule and a covariance matrix has the order of $n\,(b - \log(\epsilon))$ with $b \in \mathbb{R}$ when $\epsilon \to 0$. In addition, this analysis is valid on a wide class of functions containing some smooth positively

homogeneous functions with a unique optimum and some smooth strongly convex functions.

The link to drift analysis is also unveiled for the results in [106–111]. The expected hitting time $\mathbb{E}(\tau_\epsilon)$ of step-size adaptive $(1+\lambda)$ or $(1,\lambda)$-ES is lower bounded by $-b\,n\lambda\log(\lambda)\log(\epsilon)$ when $\epsilon \to 0$ where $b > 0$ is a constant [109, 111]. Linear bounds of the expected hitting time for a variant of the $(1+1)$-ES with $1/5$-success rule are established in [106–108, 110] for the sphere and certain convex-quadratic problems.

Chapter 3

Scaling invariant functions versus Positively homogeneous functions

Contents

Note: the content of this chapter is published in the Journal of Optimization Theory and Applications (JOTA), and presented in [184]:
Cheikh Toure, Armand Gissler, Anne Auger and Nikolaus Hansen, *Scaling-invariant functions versus positively homogeneous functions*, Journal of Optimization Theory and Applications, 2021.

Scaling-invariant functions preserve the order of points when the points are scaled by the same positive scalar (usually with respect to a unique reference point). Composites of strictly monotonic functions with positively homogeneous functions are scaling-invariant with respect to zero. We prove in this paper that also the reverse is true for large classes of scaling-invariant functions. Specifically, we give necessary and sufficient conditions for scaling-invariant functions to be composites of a strictly monotonic function with a positively homogeneous function. We also study sublevel sets of scaling-invariant functions generalizing well-known properties of positively homogeneous functions.

3.1 Introduction

A function $f : \mathbb{R}^n \to \mathbb{R}$ is scaling-invariant (SI) with respect to a reference point $x^\star \in \mathbb{R}^n$ if for all $x, y \in \mathbb{R}^n$ and $\rho > 0$:

$$f(x^\star + x) \leq f(x^\star + y) \iff f(x^\star + \rho x) \leq f(x^\star + \rho y) \ , \tag{3.1}$$

that is, the f-order of any two points is invariant under a multiplicative change of their distance to the reference point—the order only depends on their direction and their *relative* distance to the reference. Scaling-invariant functions appear naturally when studying the convergence of comparison-based optimization algorithms where the update of the state of the algorithm is using f only through comparisons of candidate solutions [25, 68]. A famous example of a comparison-based optimization algorithm is the Nelder-Mead method [146].

A function $p : \mathbb{R}^n \to \mathbb{R}$ is positively homogeneous (PH) with degree $\alpha > 0$ (PH_α) if for all $x \in \mathbb{R}^n$ and $\rho > 0$:

$$p(\rho\, x) = \rho^\alpha p(x) \ . \tag{3.2}$$

Positively homogeneous functions are scaling-invariant with respect to $x^\star = 0$. We also consider that $x \mapsto p(x - x^\star)$ is positively homogeneous w.r.t. $x^\star$ when p is positively homogeneous. Linear functions, norms, and convex quadratic functions are positively homogeneous. We can define PH functions piecewise on cones or half-lines, because a function is PH_α if and only if (3.2) is satisfied within each cone or

half-line (which is not the case with SI functions where x and y in (3.1) can belong to different cones). For example, the function $p : \mathbb{R}^n \to \mathbb{R}$ defined as $p(x) = x_1$ if $x_1 x_2 > 0$ and $p(x) = 0$ otherwise, is PH_1. Positively homogeneous functions and in particular increasing positively homogenous functions are well-studied in the context of Monotonic Analysis [62, 163, 164] or nonsmooth analysis and nonsmooth optimization [74]. Specifically, non-linear programming problems where the objective function and constraints are positively homogeneous are analyzed in [132] whereas saddle representations of continuous positively homogeneous functions by linear functions are established in [75]. The (left) composition of a PH function with a strictly monotonic function is SI while this composite function is in general not PH. One of the questions we investigate in this paper is to which extent SI functions and composites of PH functions with strictly monotonic functions are the same. We prove that a *continuous* SI function is always the composite of a strictly monotonic function with a PH function. We give necessary and sufficient conditions for an SI function to be the composite of a strictly monotonic function with a PH function in the general case.

Only level sets or sublevel sets matter to determine the difficulty of an SI problem optimized with a comparison-based algorithm. We investigate different properties of level sets thereby generalizing properties that are known for PH functions, including a formulation of the Euler homogenous function theorem that holds for PH functions.

Notation

We denote $\mathbb{R}_+$ the interval $[0, +\infty)$, $\mathbb{R}_- = (-\infty, 0]$, $\mathbb{Z}$ the set of all integers, $\mathbb{Z}_+$ the set of all non-negative integers and $\mathbb{Q}$ the set of rational numbers. The Euclidean norm is denoted by $\|.\|$. For $x \in \mathbb{R}^n$ and $\rho > 0$, we denote by $\mathcal{B}(x, \rho) = \{y \in \mathbb{R}^n; \|x - y\| < \rho\}$ the open ball centered at x and of radius ρ, $\overline{\mathcal{B}(x, \rho)}$ its closure and $\mathcal{S}(x, \rho)$ its boundary. When they are centered at 0, we denote $\mathcal{B}_\rho = \mathcal{B}(0, \rho)$, $\overline{\mathcal{B}}_\rho = \overline{\mathcal{B}(0, \rho)}$ and $\mathcal{S}_\rho = \mathcal{S}(0, \rho)$. We refer to a proper interval containing more than a single element as *nontrivial* interval. For a nontrivial interval $I \subset \mathbb{R}$ and a function $\varphi : I \to \mathbb{R}$, we use the terminology of strictly increasing (respectively strictly decreasing) if for all $a, b \in I$ with $a < b$, $\varphi(a) < \varphi(b)$ (respectively $\varphi(a) > \varphi(b)$). For a real number ρ and a subset $A \subset \mathbb{R}^n$, we define $\rho A = \{\rho x; x \in A\}$. For a function f, we denote by $\mathrm{Im}(f)$ the image of f.

3.2 Preliminaries

Given a function $f : \mathbb{R}^n \to \mathbb{R}$ and $x \in \mathbb{R}^n$, we denote the level set going through x as $\mathcal{L}_{f,x} = \{y \in \mathbb{R}^n, f(y) = f(x)\}$ and the sublevel set as $\mathcal{L}^{\leq}_{f,x} = \{y \in \mathbb{R}^n, f(y) \leq f(x)\}$.

If f is SI with respect to $x^\star$, then the function $x \mapsto f(x + x^\star) - f(x^\star)$ is scaling invariant with respect to 0. Hence, if a function f is SI, we assume in the following

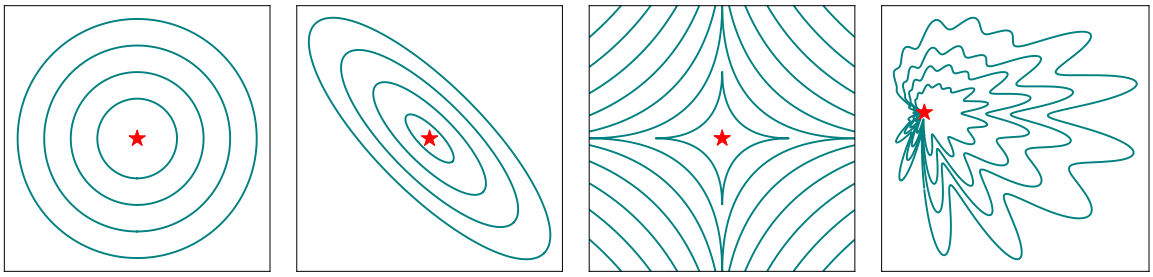


Figure 3.1: Level sets of SI functions with respect to the red star $x^\star$. The four functions are strictly increasing transformations of $x \mapsto p(x - x^\star)$ where p is a PH function. From left to right: $p(x) = \|x\|$; $p(x) = x^\top Ax$ for A symmetric positive and definite; $p(x) = \left(\sum_i \sqrt{|x_i|}\right)^2$ the $\frac{1}{2}$-norm; a randomly generated SI function from a "smoothly" randomly perturbed sphere function. The two first functions from the left have convex sublevel sets, contrary to the last two.

that f is SI with respect to the reference point 0 and that $f(0) = 0$, without loss of generality.

We can immediately imply from (3.1) that if x and y belong to the same level set, then ρx and ρy belong to the same level set. Hence the level set of x and ρx are scaled from one another, i.e. $\mathcal{L}_{f,\rho x} = \rho \mathcal{L}_{f,x}$.

Similarly, since for any $x, y \in \mathbb{R}^n$ and $\rho > 0$, $f(\frac{y}{\rho}) \leq f(x)$ if and only if $f(y) \leq f(\rho x)$,

$$\mathcal{L}^{\leq}_{f,\rho x} = \rho \mathcal{L}^{\leq}_{f,x} \text{ , and } \mathcal{L}_{f,\rho x} = \rho \mathcal{L}_{f,x} \quad . \tag{3.3}$$

These properties are visualized in Figure 4.1.

Given an SI function f, we define surjective restrictions of f to half-lines along a vector $x \in \mathbb{R}^n$ as

$$f_x : t \in [0, \infty) \mapsto f(tx) \quad . \tag{3.4}$$

It is immediate to see that the f_x are also SI[1]. However, f may not be SI even when all f_x are[2].

Scaling invariant functions have at most one isolated local optimum [25] where an isolated local optimum, say, an isolated argmin, x, for a function $g : \mathbb{R}^n \to \mathbb{R}$ is defined in that there exists $\epsilon > 0$ such that for all $y \in \mathcal{B}(x, \epsilon) \setminus \{x\}$, $g(y) > g(x)$. This result is reminded in the following proposition.

[1] This directly follows because for $s, t \in \mathbb{R}_+$ and $\rho > 0$, $f_x(t) \leq f_x(s) \iff f(tx) \leq f(sx) \iff f(\rho tx) \leq f(\rho sx) \iff f_x(\rho t) \leq f_x(\rho s)$.

[2] For example, define $f : \mathbb{R} \to \mathbb{R}$ as $t \mapsto t$ on $\mathbb{R}_+$ and $t \mapsto t^2$ on $\mathbb{R}_-$. Then $f_1(t) = t$ and $f_{-1}(t) = t^2$, for $t \in \mathbb{R}_+$, are both SI and even PH with degree 1 and 2, respectively. But f is not SI, and hence also not PH, because $f(\frac{1}{2}) = \frac{1}{2} > \frac{1}{4} = f(-\frac{1}{2})$ but $f(4 \times \frac{1}{2}) = 2 < 4 = f(4 \times (-\frac{1}{2}))$.

Proposition 1 (see [25, Proposition 3.2])**.** *Let f be an SI function. Then f can admit an isolated local optimum only in $f(0) = 0$ and this local optimum is also the global optimum. In addition, the functions f_x cannot admit a local plateau, i.e., a ball where the function is locally constant, unless the function is equal to 0 everywhere.*

We characterize in the following the functions f_x of an SI function f under different conditions.

Proposition 2. *If f is a continuous SI function on $\mathbb{R}^n$, then for all $x \in \mathbb{R}^n$, f_x is either constant equal to 0 or strictly monotonic.*

More specifically, if $\varphi : \mathbb{R}_+ \to \mathbb{R}$ is a 1-dimensional continuous SI function, then φ is either constant equal to 0 or strictly monotonic.

Proof. Assume that φ is not strictly monotonic on $[0, \infty)$. Then φ is not strictly monotonic on $(0, \infty)$. Since continuous injective functions are strictly monotonic, φ is not injective on $(0, \infty)$. Therefore there exists $0 < s < t$ such that $\varphi(s) = \varphi(t)$. By scaling-invariance, it follows that $\varphi(\frac{s}{t}) = \varphi(1)$. It follows iteratively that for all integer $k > 0$, $\varphi\left((\frac{s}{t})^k\right) = \varphi(1)$. Taking the limit for $k \to \infty$, we obtain that $\varphi(0) = \varphi(1)$. Thereby by scaling-invariance again, it follows for all $\rho > 0$ that $\varphi(0) = \varphi(\rho)$. Hence we have shown that if φ is not strictly monotonic, it is a constant function.

Now if f is a continuous SI function on $\mathbb{R}^n$ and $x \in \mathbb{R}^n$, then f_x is also scaling invariant and continuous. Then it follows that f_x is either constant or strictly monotonic. □ □

We deduce from Proposition 2 the next corollary.

Corollary 1. *Let f be a continuous SI function. If f has a local optimum at x, then for all $t \geq 0$, $f(tx) = f(0)$. In particular, if f has a global argmin (resp. argmax), then 0 is a global argmin (resp. argmax).*

Proof. Assume that there exists a local optimum at x. Then f_x has a local optimum at 1. Therefore f_x is not strictly monotonic, and thanks to Proposition 2, f_x is necessarily a constant function. In other words, $f(tx) = f(0)$ for all $t \geq 0$. □ □

We derive another proposition with the same conclusions as Proposition 2 but under a different assumption. We start by showing the following lemma.

Lemma 1. *Let $\varphi : \mathbb{R}_+ \to \mathbb{R}$ be an SI function continuous at 0 and strictly monotonic on a nontrivial interval $I \subset \mathbb{R}_+$, then φ is strictly monotonic.*

Proof. Assume without loss of generality that φ is strictly increasing on I and that $I = (a, b)$ with $0 < a < b$, up to replacing I with a subset of I. Denote $\rho = \frac{b}{a}$. Then $\left\{\left[\rho^k, \rho^{k+1}\right]\right\}_{k \in \mathbb{Z}}$ covers $(0, \infty)$. To prove that φ is strictly increasing on $(0, \infty)$, it is enough to prove that φ is strictly increasing on $[\rho^k, \rho^{k+1}]$ for all integer k.

Let k be an integer and (x, y) two real numbers such that $\rho^k \leq x < y \leq \rho^{k+1}$. Then $a \leq \frac{ax}{\rho^k} < \frac{ay}{\rho^k} \leq a\rho = b$. Therefore $\varphi(\frac{ax}{\rho^k}) < \varphi(\frac{ay}{\rho^k})$. And by scaling-invariance, $\varphi(x) < \varphi(y)$.

With the continuity at 0, it follows that φ is strictly increasing on $\mathbb{R}_+$. □ □

We derive from Lemma 1 the following proposition.

Proposition 3. *Let f be an SI function continuous at 0. Assume that each f_x is on some nontrivial interval either strictly monotonic or constant. Then for all $x \in \mathbb{R}^n$, f_x is either constant equal to 0 or strictly monotonic.*

Note that the continuity of the function f_x alone does not suffice to conclude that f_x is either constant or strictly monotonic on some nontrivial interval. Indeed there exist 1-D continuous functions (even differentiable functions) that are not monotonic on any nontrivial interval [43, 59, 97].

For the sake of completeness, we construct SI functions in $\mathbb{R}_+$ that are not monotonic on any nontrivial interval. The construction of such functions is based on the nonlinear solutions of the Cauchy functional equation: for all $x, y \in \mathbb{R}$, $g(x + y) = g(x) + g(y)$, called Hamel functions [130]. A Hamel function f also satisfies $f(q_1 x + q_2 y) = q_1 f(x) + q_2 f(y)$ for all real numbers x, y and rational numbers q_1, q_2 [1, Chapter 2]. Since g is nonlinear, there exist real numbers x and y such that the vectors $\{(x, g(x)), (y, g(y))\}$ form a basis of $\mathbb{R}^2$ over the field $\mathbb{R}$. Then the graph of g, which is a vector subspace of $\mathbb{R}^2$ over the field $\mathbb{Q}$, contains $\{q_1 \cdot (x, g(x)) + q_2 \cdot (y, g(y)); (q_1, q_2) \in \mathbb{Q}^2\}$ which is dense in $\mathbb{R}^2$. Therefore a 1-D Hamel function is highly pathological, since its graph is dense in $\mathbb{R}^2$.

Lemma 2. *There exist SI functions on $\mathbb{R}_+$ that are neither monotonic nor continuous on any nontrivial interval.*

Proof. We start by choosing a nonlinear solution of the Cauchy's functional equation denoted by $g : \mathbb{R} \to \mathbb{R}$, knowing that there are uncountably many ways to pick such a g [130]. Then for all real numbers a and b, $g(a + b) = g(a) + g(b)$. And since g is not linear, we also know that g is neither continuous nor monotonic on any nontrivial interval [130]. Let us define $f = \exp \circ g \circ \log$ on $(0, \infty)$ and $f(0) = 0$. Then $f(x) > 0$ for all $x > 0$ and f is still not monotonic on any nontrivial interval. We also have for all $\rho > 0$ and $x > 0$, $f(\rho x) = \exp(g(\log(x) + \log(\rho))) = \exp(g(\log(x))) \exp(g(\log(\rho))) = f(x)f(\rho)$. This last result gives the scaling-invariance property. □ □

Based on Lemma 2, we derive the next proposition.

Proposition 4. *There exist SI functions f on $\mathbb{R}^n$ such that for all non-zero x, f_x is neither monotonic nor continuous on any nontrivial interval.*

Proof. Based on Lemma 2, there exists $\varphi : \mathbb{R}_+ \to \mathbb{R}$ SI on $\mathbb{R}_+$ which is neither monotonic nor continuous on any nontrivial interval. We construct f as follows. For all $x \in \mathbb{R}^n$, $f(x) = \varphi(\|x\|)$. Then f is SI because for $x, y \in \mathbb{R}^n$ and for $\rho > 0$, $f(x) \leq f(y) \iff \varphi(\|x\|) \leq \varphi(\|y\|) \iff \varphi(\rho\|x\|) \leq \varphi(\rho\|y\|) \iff f(\rho x) \leq f(\rho y)$. In addition

for a non-zero x and $t \geq 0$, $f_x(t) = f(tx) = \varphi(t\|x\|)$ and then f_x is neither monotonic nor continuous on any nontrivial interval. □ □

Now assume that f is a continuous scaling invariant function and we can write $f = \varphi \circ g$ where φ is a continuous bijection and g is a positively homogeneous function. As a direct consequence of the bijection theorem given in Appendix 1.1, φ^{-1} is continuous if φ is a continuous bijection defined on an interval. Therefore $g = \varphi^{-1} \circ f$ is also continuous. This result is stated in the following corollary.

Corollary 2. *Let f be a continuous SI function, φ a continuous bijection defined on an interval in $\mathbb{R}$ and p a positively homogeneous function such that $f = \varphi \circ p$. Then p is also continuous.*

3.3 Scaling-invariant Functions as Composite of Strictly Monotonic Functions with Positively Homogeneous Functions

As underlined in the introduction, compositions of strictly monotonic functions with positively homogeneous functions are scaling-invariant (SI) functions. We investigate in this section under which conditions the converse is true, that is, when SI functions are compositions of strictly monotonic functions with PH functions. Section 3.3.1 shows that continuity is a sufficient condition, whereas Section 3.3.2 gives some necessary *and* sufficient condition on f to be decomposable in this way.

3.3.1 Continuous SI Functions

We prove in this section a main result of the paper: any continuous SI function f can be written as $f = \varphi \circ p$ where p is PH_1 and φ is a homeomorphism (and in particular strictly monotonically increasing and continuous). The proof relies on the following proposition where we do not assume yet that f is continuous but only the restrictions of f to the half-lines originating in 0, the f_x functions.

Proposition 5. *Let f be an SI function such that for any $x \in \mathbb{R}^n$, f_x as defined in (3.4) is continuous and strictly monotonic or constant. Then for all $\alpha > 0$, there exist a PH_α function p and a strictly increasing, continuous bijection (thus a homeomorphism) φ such that $f = \varphi \circ p$. For a non-zero f and $\alpha > 0$, the choice of (φ, p) is unique up to a left composition of p with a piece-wise linear function.*

(i) *In addition, if all non-constant f_x have the same monotonicity for all $x \in \mathbb{R}^n$, then for any $x_0 \in \mathbb{R}^n$ such that $f(x_0) \neq 0$, the homeomorphism φ corresponding to a PH_1 function can be chosen as f_{x_0} and is at least as smooth as f.*

(ii) *Otherwise, there exist* $x_1, x_{-1} \in \mathbb{R}^n$ *such that* f_{x_1} *is strictly increasing and* $f_{x_{-1}}$ *is strictly decreasing. And for any such* x_1 *and* x_{-1} *we can choose as homeomorphism* φ *corresponding to a* PH_1 *function the function* f_{x_1} *on* $\mathbb{R}_+$ *and* $t \mapsto f_{x_{-1}}(-t)$ *on* $\mathbb{R}_-$*.*

Proof. Let f be an SI function such that for any $x \in \mathbb{R}^n$, f_x is either a constant or a strictly monotonic continuous function.

In the case where all the f_x are constant for all $x \in \mathbb{R}^n$, then $f = 0$ and therefore we can take $p_\alpha = 0$ as a candidate for a continuous PH_α and $\varphi_\alpha : t \mapsto t$ as the candidate for the corresponding homeomorphism.

From now on, at least one of the $\{f_x\}_{x \in \mathbb{R}^n}$ is non-constant. We now split the proof in two parts, the case where all the non-constant f_x have the same monotonicity and the case where there exist $x_1, x_{-1} \in \mathbb{R}^n$ such that f_{x_1} is strictly increasing and $f_{x_{-1}}$ is strictly decreasing.

Part 1. Assume here that all the non-constant f_x have the same monotonicity for all $x \in \mathbb{R}^n$. And up to a transformation $x \mapsto -f(x)$, we can assume without loss of generality that they are increasing. Therefore 0 is a global argmin and since we have assumed $f(0) = 0 : f(x) \geq 0$ for all $x \in \mathbb{R}^n$. Then there exists $x_0 \in \mathbb{R}^n$ such that $f(x_0) > 0$.

For any $x \in \mathcal{L}_{f,x_0} = \{y \in \mathbb{R}^n, f(y) = f(x_0)\}$, and any $\lambda > 0$ different than 1, $\lambda x \notin \mathcal{L}_{f,x_0}$. Indeed, as $x \in \mathcal{L}_{f,x_0}$, we know from Proposition 2 that f_x is strictly increasing on $\mathbb{R}_+$, since f_x cannot be constant equal to 0.

Moreover, for all $x \in \mathbb{R}^n$ such that $f(x) \neq 0$, there exists $\lambda > 0$ such that $\lambda x \in \mathcal{L}_{f,x_0}$. Indeed, if $f(x) < f(x_0)$, the intermediate value theorem applied to the continuous function f_{x_0} shows that there exists $0 < t < 1$ such that $f(tx_0) = f_{x_0}(t) = f(x)$, and then $f(\frac{1}{t}x) = f(x_0)$. And by interchanging x and x_0, the same argument holds if $f(x) > f(x_0)$.

The two last paragraphs ensure that for all x such that $f(x) \neq 0$, there exists a unique positive number λ_x such that $\lambda_x x \in \mathcal{L}_{f,x_0}$. Let us define the function p for all $x \in \mathbb{R}^n$ as follows: if $f(x) \neq 0$ then $p(x) = \frac{1}{\lambda_x}$, otherwise $p(x) = 0$. We prove in the following that p is PH_1.

Let $x \in \mathbb{R}^n$ and $\rho > 0$. If $f(x) = 0$ (hence $f(\rho x) = 0$), then $p(\rho x) = 0 = \rho p(x)$. Otherwise $f(x) > 0$ (hence $f(\rho x) > 0$), and $p(\rho x) = \frac{\rho}{\lambda_x}$ since $\frac{\lambda_x}{\rho}$ is the (unique) positive number such that $\frac{\lambda_x}{\rho}\rho x = \lambda_x x \in \mathcal{L}_{f,x_0}$. And thereby $p(\rho x) = \rho\, p(x)$.

We prove that $f = f_{x_0} \circ p$, where f_{x_0} is a continuous strictly increasing function and p is PH_1. Let $x \in \mathbb{R}^n$. If $f(x) = 0$, then $p(x) = 0$, and then $f(x) = 0 = f(0) = f_{x_0}(0) = (f_{x_0} \circ p)(x)$. Otherwise, we have by construction that $\frac{x}{p(x)} \in \mathcal{L}_{f,x_0}$. Therefore $f(\frac{x}{p(x)}) = f(x_0)$ and then $f(x) = f(p(x)x_0) = f_{x_0}(p(x))$. By Theorem 14, $\varphi = f_{x_0}$ is a

homeomorphism. Let $\alpha > 0$, define $\tilde{\varphi} = t \mapsto \varphi(t^{1/\alpha})$ and $\tilde{p} = p^\alpha$. Then $\tilde{p}$ is PH_α, $\tilde{\varphi}$ is a homeomorphism and $f = \tilde{\varphi} \circ \tilde{p}$.

Assume that we have two couples of solutions (φ, p) and $(\bar{\varphi}, \bar{p})$ such that $f = \varphi \circ p = \bar{\varphi} \circ \bar{p}$ where $\varphi, \bar{\varphi}$ are homeomorphisms and $p, \bar{p}$ are PH_α. For all $t > 0$ and $x \in \mathbb{R}^n$, we have for instance $p(tx) = t^\alpha p(x)$. Therefore $\mathrm{Im}(p) = \mathbb{R}_+$. Denote $\psi = \bar{\varphi}^{-1} \circ \varphi$. For all $\lambda > 0$ and $x \in \mathbb{R}^n$, $\psi(\lambda^\alpha p(x)) = \psi(p(\lambda x)) = \bar{p}(\lambda x) = \lambda^\alpha \psi(p(x))$. Hence ψ is PH_1 on $\mathbb{R}_+$. For all $t > 0$, $\psi(t) = t\psi(1)$. Therefore ψ is linear.

Part 2. Assume now that there exist $x_1, x_{-1} \in \mathbb{R}^n$ such that f_{x_1} is strictly increasing and $f_{x_{-1}}$ is strictly decreasing. Then $f(x_1) > 0$ and $f(x_{-1}) < 0$. Then thanks to the intermediate value theorem, if $f(x) > 0$, there exists a unique positive number λ_x such that $\lambda_x x \in \mathcal{L}_{f,x_1}$, and if $f(x) < 0$, there exists a unique positive number λ_x such that $\lambda_x x \in \mathcal{L}_{f,x_{-1}}$. We define now p for all $x \in \mathbb{R}^n$ as follows: if $f(x) = 0$ then $p(x) = 0$, if $f(x) > 0$ then $p(x) = \frac{1}{\lambda_x}$, and finally if $f(x) < 0$ then $p(x) = -\frac{1}{\lambda_x}$. Let us show that p is PH_1. Indeed for any $\rho > 0$ and $x \in \mathbb{R}^n$, if $f(x) = 0$ (hence $f(\rho x) = 0$), then $p(\rho x) = 0 = \rho p(x)$. If $f(x) > 0$ (hence $f(\rho x) > 0$), and $p(\rho x) = \frac{\rho}{\lambda_x} = \rho p(x)$ since $\frac{\lambda_x}{\rho}$ is the (unique) positive number such that $\frac{\lambda_x}{\rho}\rho x = \lambda_x x \in \mathcal{L}_{f,x_1}$. And finally if $f(x) < 0$ (hence $f(\rho x) < 0$), then $p(\rho x) = -\frac{\rho}{\lambda_x} = \rho p(x)$ since $\frac{\lambda_x}{\rho}$ is the (unique) positive number such that $\frac{\lambda_x}{\rho}\rho x = \lambda_x x \in \mathcal{L}_{f,x_{-1}}$. Hence p is PH_1.

We define now the function $\varphi : \mathbb{R} \to \mathbb{R}$ such that if $t \geq 0$, $\varphi(t) = f_{x_1}(t)$ and if $t \leq 0$, $\varphi(t) = f_{x_{-1}}(-t)$. Then, φ is well defined ($f_{x_1}(0) = 0 = f_{x_{-1}}(0)$), continuous and strictly increasing.

Let $x \in \mathbb{R}^n$. If $f(x) = 0$, then $p(x) = 0$, and then $f(x) = 0 = (\varphi \circ p)(x)$. If $f(x) > 0$, $\varphi(p(x)) = f_{x_1}(p(x)) = f(p(x)x_1) = f(x)$ since $\frac{x}{p(x)} \in \mathcal{L}_{f,x_1}$. And finally if $f(x) < 0$, $\varphi(p(x)) = f_{x_{-1}}(-p(x)) = f(-p(x)x_{-1}) = f(x)$ since $-\frac{x}{p(x)} = \lambda_x x \in \mathcal{L}_{f,x_{-1}}$. Thereby, $f = \varphi \circ p$. Theorem 14 ensures that φ is a homeomorphism. By defining for all $\alpha > 0$, $\tilde{\varphi}(t) = \varphi(t^{1/\alpha})$ if $t \geq 0$, $\tilde{\varphi}(t) = \varphi(-(-t)^{1/\alpha})$ if $t < 0$, $\tilde{p}(x) = p(x)^\alpha$ if $p(x) \geq 0$ and $\tilde{p}(x) = -(-p(x))^\alpha$ if $p(x) < 0$, it follows that $f = \tilde{\varphi} \circ \tilde{p}$.

Assume here again that we have two couples of solutions (φ, p) and $(\bar{\varphi}, \bar{p})$ such that $f = \varphi \circ p = \bar{\varphi} \circ \bar{p}$ where $\varphi, \bar{\varphi}$ are homeomorphisms and $p, \bar{p}$ are PH_α. For all $t > 0$ and $x \in \mathbb{R}^n$, we have $p(tx) = t^\alpha p(x)$. Therefore $\mathrm{Im}(p) = \mathbb{R}$ since $p(x_1)$ and $p(x_2)$ have opposite signs. Denote $\psi = \bar{\varphi}^{-1} \circ \varphi$. For all $\lambda > 0$ and $x \in \mathbb{R}^n$, $\psi(\lambda^\alpha p(x)) = \psi(p(\lambda x)) = \bar{p}(\lambda x) = \lambda^\alpha \psi(p(x))$. Hence ψ is PH_1 on $\mathbb{R}$. For all $t > 0$, $\psi(t) = t\psi(1)$ and $\psi(-t) = t\psi(-1)$ Therefore depending on the values of $\psi(1)$ and $\psi(-1)$, ψ is either linear or piece-wise linear. □ □

We now use the previous proposition to prove that a continuous SI function is a homeomorphic transformation of a continuous PH_1 function. The proof relies on the result that for a continuous SI function, the f_x are either constant or strictly monotonic and continuous (see Proposition 2). We distinguish the case where f has a global

optimum as in Proposition 5 (i) and the case where f does not have a global optimum as in Proposition 5 (ii). Overall the following result holds.

Theorem 1. *Let f be a continuous SI function. Then for all $\alpha > 0$, there exists a continuous PH$_\alpha$ function p and a strictly increasing and continuous bijection (thus a homeomorphism) φ such that $f = \varphi \circ p$.*

For a non-zero f and $\alpha > 0$, the choice of (φ, p) is unique up to a left composition of p with a piece-wise linear function. If f admits a global optimum, then 0 is also a global optimum. For any $x_0 \in \mathbb{R}^n$ such that $f(x_0) \neq 0$, in the case where p is a PH_1 function, the homeomorphism φ can be chosen as f_{x_0} and is at least as smooth as f.

If f does not admit a global optimum, then there exist $x_1, x_{-1} \in \mathbb{R}^n$ such that $f(x_1) > 0$ and $f(x_{-1}) < 0$. For any such x_1 and x_{-1}, in the case where p is a PH_1 function, the homeomorphism φ can be chosen as the function equal to f_{x_1} on $\mathbb{R}_+$ and equal to $t \mapsto f_{x_{-1}}(-t)$ on $\mathbb{R}_-$.

Proof. Let f be a continuous SI function. Thanks to Proposition 2, for all $x \in \mathbb{R}^n$, f_x is either constant equal to 0 or strictly monotonic.

Part 1. Assume that f has a global optimum. Corollary 1 shows that 0 is also a global optimum. Then we can apply Proposition 5 in the case where the non-constant f_x have the same monotonicity. Let $x_0 \in \mathbb{R}^n$ such that $f(x_0) \neq 0$ and define $\varphi = f_{x_0}$. Then $f = \varphi \circ p$ and φ is a homeomorphism. That settles the continuity of the PH_1 function $\varphi^{-1} \circ f$ thanks to Corollary 2.

Part 2. Assume in this part that f has no global optimum. Since 0 is not a global optimum, we can find x_1 and x_{-1} such that $f(x_1) > 0$ and $f(x_{-1}) < 0$. Therefore f_{x_1} is strictly increasing and $f_{x_{-1}}$ is strictly decreasing. We apply Proposition 5 in the case where the non-constant f_x do not have the same monotonicity. If φ is the function equal to f_{x_1} on $\mathbb{R}_+$ and to $t \mapsto f_{x_{-1}}(-t)$ on $\mathbb{R}_-$, then $f = \varphi \circ p$ where φ is a homeomorphism. That settles the continuity of the PH_1 function $\varphi^{-1} \circ f$ thanks to Corollary 2. For all $\alpha > 0$, the unique construction of (φ, p) up to a piece-wise linear function in both parts is a consequence of Proposition 5. □ □

3.3.2 Sufficient and Necessary Condition to be the Composite of a PH Function

We have seen in the previous section that a *continuous* SI function can be written as $\varphi \circ p$ with φ strictly monotonic and p PH. Relaxing continuity, we prove in the next theorem some necessary *and* sufficient condition under which an SI function is the composite of a PH function with a strictly monotonic function.

Theorem 2. *Let f be an SI function. There exist a PH$_1$ function p and a strictly increasing function φ such that $f = \varphi \circ p$ if and only if for all $x \in \mathbb{R}^n$, the function f_x is either constant or strictly monotonic and the strictly increasing f_x share the same*

image (i.e., if $\lambda \in \mathbb{R}$ is reached for one of these functions, then it is reached for all of them) and the strictly decreasing ones too.

For a non-zero f, up to a left composition of p with a piece-wise linear function, the choice of (φ, p) is unique.

Proof. We prove first the forward implication. Suppose there is a PH_1 function p and a strictly monotonic function φ such that $f = \varphi \circ p$. Consider $x \in \mathbb{R}^n$. Either $p(x) = 0$ and then for any $t \geqslant 0$ we have that $p(tx) = 0$, so that $f_x(t) = f(tx) = \varphi(p(tx)) = \varphi(0)$ and f_x is constant on $\mathbb{R}_+$, or $p(x) \neq 0$, and then $t \in \mathbb{R}_+ \mapsto p(tx) = tp(x)$ is strictly monotonic, and $f_x(t) = \varphi(p(tx))$ is strictly monotonic too on $\mathbb{R}_+$. Moreover, consider $x_1 \neq x_2$ such that f_{x_1} and f_{x_2} are increasing. Then $p(x_1)$ and $p(x_2)$ are of the same sign, so there is some $t_* > 0$ such that $p(x_1) = t_* p(x_2) = p(t_* x_2)$, so the functions $t \mapsto f(tx_1)$ and $t \mapsto f(tt_* x_2)$ are equal, so the functions f_{x_1} and f_{x_2} take the same values. The same applies on the strictly decreasing functions.

We now prove the backward implication. Suppose that the functions f_x are either constant or strictly monotonic and the increasing ones share the same values and the decreasing ones too.

If all the f_x are constant, then for all $x \in \mathbb{R}^n$, $f(x) = f(0) = 0$ and it is enough to write $f = \varphi \circ p$ with $p = t \mapsto t$ on $\mathbb{R}_+$ and $p = 0$. We assume from now on that at least one f_x is not constant.

Consider that all the non-constant f_x have the same monotonicity. Let us choose x_0 such that $f(x_0) \neq 0$. Then for all $x \neq 0$, f_x and f_{x_0} have the same monotonicity. Since they have the same image and are injective, there exists a unique $\lambda_x > 0$ such that $\lambda_x x \in \mathcal{L}_{f,x_0}$. We then define p and φ as in the Part 1 of Theorem 1 to ensure that $f = \varphi \circ p$ where p is PH_1 and φ is strictly monotonic.

Consider finally that all the non-constant f_x do not have the same monotonicity. Then there exist x_1 and x_{-1} such that $f(x_1) > 0$ and $f(x_{-1}) < 0$. Then, thanks to the assumption that all increasing f_x share the same values and the strictly decreasing f_x too, if $f(x) > 0$, then there exists a unique positive number λ_x such that $\lambda_x x \in \mathcal{L}_{f,x_1} = \{y \in \mathbb{R}^n, f(y) = f(x_1)\}$, and if $f(x) < 0$, then there exists a unique (thanks to the assumption of strict monotonicity for the non-constant f_x) positive number λ_x such that $\lambda_x x \in \mathcal{L}_{f,x_{-1}}$. Therefore, we can define p as in Theorem 1. As before, p is PH_1. Define also the function $\varphi : \mathbb{R} \to \mathbb{R}$ as in Theorem 1. It is still increasing, but not necessarily continuous. Then, as in Theorem 1, $f = \varphi \circ p$.

The proof of the unicity of (φ, p) up to a piece-wise real linear function is similar to the proof in Proposition 5. □ □

Complementing Theorem 2, we construct an example of an SI function that can not be decomposed as $f = \varphi \circ p$, because the strictly increasing f_x do not share the same image. Define f such that for all $x \in \mathbb{R}^n$, $f(x) = \tanh(x_1)$ if the first coordinate

$x_1 \geq 0$ and $f(x) = 1 + \exp(-x_1)$ otherwise. Then f is SI and if $x_1 \neq 0$, f_x is strictly increasing. However for all x such that $x_1 > 0$ then $\mathrm{Im}(f_x) = [0, 1)$ and otherwise for x such that $x_1 \leq 0$, $\mathrm{Im}(f_x) = \{0\} \cup (2, \infty)$.

3.4 Level Sets of SI Functions

Scaling-invariant functions appear naturally when studying the convergence of comparison-based optimization algorithms [25]. In this specific context, the difficulty of a problem is entirely determined by its level sets whose properties are studied in this section.

3.4.1 Identical Sublevel Sets

Level sets and sublevel sets of a function f remain unchanged if we compose the function with a strictly increasing function φ since

$$f(x) \leq f(y) \iff \varphi(f(x)) \leq \varphi(f(y)) \ . \tag{3.5}$$

We prove in the next theorem that two arbitrary functions f and p have the same level sets if and only if $f = \varphi \circ p$ where φ is strictly increasing.

Theorem 3. *Two functions f and p have the same sublevel sets if and only if there exists a strictly increasing function φ such that $f = \varphi \circ p$.*

Proof. If $f = \varphi \circ p$ with φ strictly increasing, since sublevel sets are invariant by φ, f and p have the same sublevel sets. Now assume that f and p have the same sublevel sets. Then for all $x \in \mathbb{R}^n$, there exists $T(x) \in \mathbb{R}^n$ such that $\mathcal{L}^{\leq}_{f,x} = \mathcal{L}^{\leq}_{p,T(x)}$. In other words for all $y \in \mathbb{R}^n$, $f(y) \leq f(x) \iff p(y) \leq p(T(x))$. We define the function

$$\phi : \begin{cases} \mathrm{Im}(f) & \longrightarrow & \mathrm{Im}(p) \\ f(x) & \longmapsto & p\left(T(x)\right) \end{cases} .$$

The function ϕ is well-defined because for $x, y \in \mathbb{R}^n$ such that $f(x) = f(y)$, $\mathcal{L}^{\leq}_{f,x} = \mathcal{L}^{\leq}_{f,y}$. And since $\mathcal{L}^{\leq}_{f,x} = \mathcal{L}^{\leq}_{p,T(x)}$ and $\mathcal{L}^{\leq}_{f,y} = \mathcal{L}^{\leq}_{p,T(y)}$, then $\mathcal{L}^{\leq}_{p,T(x)} = \mathcal{L}^{\leq}_{p,T(y)}$, and then $p\left(T(x)\right) = p\left(T(y)\right)$. Therefore $\phi(f(x)) = \phi(f(y))$. By construction we have that $\phi \circ f = p \circ T$.

Let us show that $p \circ T = p$. We have $\mathcal{L}^{\leq}_{p\circ T,x} = \mathcal{L}^{\leq}_{f,x} = \mathcal{L}^{\leq}_{p,T(x)}$. Then $x \in \mathcal{L}^{\leq}_{p,T(x)}$ and then $p(x) \leq p(T(x))$. Therefore $p \leq p \circ T$. In addition for all $y \in \mathbb{R}^n$, there exists x such that $\mathcal{L}^{\leq}_{p,y} = \mathcal{L}^{\leq}_{f,x} = \mathcal{L}^{\leq}_{p,T(x)} = \mathcal{L}^{\leq}_{p\circ T,x}$. Then $y \in \mathcal{L}^{\leq}_{p\circ T,x}$, which induces that $p(T(y)) \leq p(T(x))$. Plus, $\mathcal{L}^{\leq}_{p,y} = \mathcal{L}^{\leq}_{p,T(x)}$, therefore $p(T(x)) = p(y)$. Thereby $p(T(y)) \leq p(y)$, and then $p \circ T \leq p$. Finally $p \circ T = p$. Hence $\phi \circ f = p$.

Let us prove now that ϕ is strictly increasing. Consider $x, y \in \mathbb{R}^n$ such that $f(x) < f(y)$. Then $\mathcal{L}^{\leq}_{f,x} \subset \mathcal{L}^{\leq}_{f,y}$ with a strict inclusion, which means that $\mathcal{L}^{\leq}_{p,T(x)} \subset \mathcal{L}^{\leq}_{p,T(y)}$ with a strict inclusion. Thereby $p(T(x)) < p(T(y))$, i.e. $\phi(f(x)) < \phi(f(y))$. Hence ϕ is strictly increasing. And up to restricting ϕ to its image, we can assume without loss of generality that ϕ is a strictly increasing bijection. We finally denote $\varphi = \phi^{-1}$ and it follows that $f = \varphi \circ p$. □ □

Theorem 3 and Theorem 2 give both equivalence conditions for an SI function f to be equal to $\varphi \circ p$ where φ is strictly increasing and p is positively homogeneous[3]. One condition is that there exists a PH function with the same sublevel sets as f, while the other condition is that the f_x are either constant or strictly monotonic, and the strictly increasing and decreasing ones have the same image, respectively.

3.4.2 Compactness of the Sublevel Sets

Compactness of sublevel sets is relevant for analyzing step-size adaptive randomized search algorithms [24, 143]. We investigate here how compactness properties shown for positively homogeneous functions extend to scaling-invariant functions. For an SI function f, we have $\mathcal{L}^{\leq}_{f,tx} = t\mathcal{L}^{\leq}_{f,x}$. When $\psi : y \longmapsto ty$ is a homeomorphism, we have that $\psi(\mathcal{L}^{\leq}_{f,x})$ equals $t\mathcal{L}^{\leq}_{f,x}$ and is compact if and only if $\mathcal{L}^{\leq}_{f,x}$ is compact. Therefore, for all $t > 0$:

$$\mathcal{L}^{\leq}_{f,tx} \text{ is compact if and only if } \mathcal{L}^{\leq}_{f,x} \text{ is compact.} \tag{3.6}$$

Furthermore, if p is a lower semi-continuous positively homogeneous function such that $p(x) > 0$ for all nonzero x then the sublevel sets of p are compact [24, Lemma 2.7]. We recall it with all the details in the following proposition:

Proposition 6 ([24, Lemma 2.7])**.** *Let p be a positively homogeneous function with degree $\alpha > 0$ and $p(x) > 0$ for all $x \neq 0$ (or equivalently 0 is the unique global argmin of p) and $p(x)$ finite for every $x \in \mathbb{R}^n$. Then for every $x \in \mathbb{R}^n$, the following holds:*

(i) $\lim_{t \to 0} p(tx) = 0$ *and for all $x \neq 0$ the function $p_x : [0, \infty) \ni t \longmapsto p(tx) \in \mathbb{R}_+$ is continuous, strictly increasing and converges to ∞ when t goes to ∞.*

(ii) *If p is lower semi-continuous, the sublevel set $\mathcal{L}^{\leq}_{p,x}$ is compact.*

We prove a similar theorem for lower semi-continuous SI functions f with continuous f_x functions, showing in particular that the unicity of the global argmin is equivalent to the above items. Note that we need to assume the continuity of the functions f_x, while this property is unconditionally satisfied for positively homogeneous functions where $p_x(t) = t^\alpha p_x(1)$ for all $x \in \mathbb{R}^n$ and for all $t > 0$.

[3] Note that in Theorem 2, we can assume without loss of generality that φ is always strictly increasing by replacing if needed φ by $t \to \varphi(-t)$ and p by $-p$.

Theorem 4. *Let f be SI. Then the conditions*

- $f(x) > 0$ *for all* $x \neq 0$ *and*
- 0 *is the unique global argmin*

are equivalent. Let f *be additionally lower semi-continuous and for all* $x \in \mathbb{R}^n$, f_x *is continuous on a neighborhood of* 0*. Then the following are equivalent:*

(i) 0 *is the unique global argmin.*

(ii) *for any* $x \in \mathbb{R}^n \setminus \{0\}$*, the function* f_x *is strictly increasing.*

(iii) *The sublevel sets* $\mathcal{L}^{\leq}_{f,x}$ *for all* x *are compact.*

Proof. Since, w.l.o.g., f is given such that $f(0) = 0$, its unique global argmin is 0 if and only if $f(x) > 0$ for all $x \neq 0$. Now we prove first that (i) $\Rightarrow$ (ii)*:* Let $x \in \mathbb{R}^n \setminus \{0\}$. Assume (by contraposition) that there exists $0 < t_1 < t_2$ such that $f_x(t_1) = f_x(t_2)$. Then by scaling-invariance, $f_x(1) = f_x\left(\frac{t_1}{t_2}\right)$. It follows by multiplying iteratively by $\frac{t_1}{t_2}$ that for all $k \in \mathbb{Z}_+$, $f_x(1) = f_x\left(\left(\frac{t_1}{t_2}\right)^k\right)$. Therefore if we take the limit when $k \to \infty$, it follows thanks to the continuity of f_x at 0 that $f(x) = f_x(1) = f(0)$, which contradicts the assumption (i). Hence, f_x is an injective function. Plus, there exists $\epsilon > 0$ such that f_x is continuous on $[0, \epsilon]$. Therefore f_x is an injective continuous function on $[0, \epsilon]$, which implies that f_x is a strictly monotonic function on $[0, \epsilon]$. Lemma 1 implies therefore that f_x is strictly monotonic. And since 0 is an argmin of f_x, then f_x is strictly increasing.

(ii) $\Rightarrow$ (iii)*:* f is lower semi-continuous on the compact $\mathcal{S}_1$, then it reaches its minimum on that compact: there exists $s \in \mathcal{S}_1$ such that $f(s) = \min_{z \in \mathcal{S}_1} f(z)$. Also, since sublevel sets of lower semi-continuous functions are closed, then $\mathcal{L}^{\leq}_{f,s}$ is closed. Now let us show that it is also bounded.

If $y \in \mathcal{L}^{\leq}_{f,s} \setminus \{0\}$, then $f(y) \leq f(s) \leq f\left(\frac{y}{\|y\|}\right)$. And since f_y is strictly increasing, we obtain that $1 \leq \frac{1}{\|y\|}$, thereby $\|y\| \leq 1$. We have shown that $\mathcal{L}^{\leq}_{f,s} \subset \overline{\mathcal{B}}_1$.

Then $\mathcal{L}^{\leq}_{f,s}$ is a compact set, as it is a closed and bounded subset of $\mathbb{R}^n$. By (3.6), it follows that $\mathcal{L}^{\leq}_{f,ts}$ is compact for all $t > 0$.

For all $x \in \mathbb{R}^n \setminus \{0\}$, $f(x) > f(0)$ thanks to (ii). Then $\mathcal{L}^{\leq}_{f,0} = \{0\}$ and is compact.

Let $x \in \mathbb{R}^n \setminus \{0\}$. Then there exists $\epsilon > 0$ such that $f_{\frac{x}{\|x\|}}$ is continuous on $[0, \epsilon]$. We have that $f_{\frac{x}{\|x\|}}(0) = 0 < f(s) \leq f_{\frac{x}{\|x\|}}(1)$, then by scaling-invariance, $f_{\frac{x}{\|x\|}}(0) < f(\epsilon s) \leq f_{\frac{x}{\|x\|}}(\epsilon)$. Therefore by the intermediate value theorem applied to $f_{\frac{x}{\|x\|}}$ continuous on $[0, \epsilon]$, there exists $t \in (0, \epsilon]$ such that $f_{\frac{x}{\|x\|}}(t) = f(\epsilon s)$. Then $\mathcal{L}^{\leq}_{f,t\frac{x}{\|x\|}} = \mathcal{L}^{\leq}_{f,\epsilon s}$ and is compact. We apply again (3.6) to observe that $\mathcal{L}^{\leq}_{f,x}$ is compact.

(iii) $\Rightarrow$ (i)*:* Let $x \in \mathcal{L}^{\leq}_{f,0}$, then $tx \in \mathcal{L}^{\leq}_{f,0}$ for all $t \geq 0$, and then $\{tx\}_{t\in\mathbb{R}_+} \subset \mathcal{L}^{\leq}_{f,0}$ which is a compact set. This is only possible if $x = 0$, otherwise the set $\{tx\}_{t\in\mathbb{R}_+}$ would not be bounded. Hence $\mathcal{L}^{\leq}_{f,0} = \{0\}$ which implies that 0 is the unique global argmin. □ □

We derive from Theorem 4 the next corollary, stating when for a lower semi-continuous SI function with a unique global argmin the intersection of any half-line of origin 0 and a level set is a singleton.

Corollary 3. *Let f be a lower semi-continuous SI function with 0 as unique global argmin. Assume that for all $x \in \mathbb{R}^n$, f_x is continuous on a neighborhood of 0, and the f_x share the same image. Then for all $x \in \mathbb{R}^n$, any half-line of origin 0 intersects $\mathcal{L}_{f,x}$ at a unique point.*

Proof. For all non-zero x, Theorem 4 ensures that f_x is strictly increasing. Therefore for a non-zero x, f_x is injective. And then the intersection of a level set and a half-line of origin 0 contains at most one point. In addition the f_x share the same image for all non-zero x. Then for two non-zero vectors x, y, there exists $t \geq 0$ such that $f_y(t) = f_x(1)$. In other words, there exists $t \geq 0$ such that $ty \in \mathcal{L}_{f,x}$. We end this proof by noticing that $\mathcal{L}_{f,0} = \{0\}$ and then intersects any half-line of origin 0 only at 0. □ □

3.4.3 Sufficient Condition for Lebesgue Negligible Level Sets

We assume that f is lower semi-continuous SI admitting a unique global argmin and all f_x are continuous and prove that f has Lebesgue negligible level sets.

Proposition 7. *Let f be an SI function with 0 as unique global argmin. Assume that f is lower semi-continuous and for all $x \in \mathbb{R}^n$, f_x is continuous. Then the level sets of f are Lebesgue negligible.*

Proof. Let $x \in \mathbb{R}^n$. Let us denote by μ the Lebesgue measure. For all $t > 0$, $\mu(\mathcal{L}_{f,tx}) = \mu(t\mathcal{L}_{f,x}) = t^n\mu(\mathcal{L}_{f,x})$, thanks to (3.3). Therefore, if $t \geq 1$, $\mu(\mathcal{L}_{f,tx}) \geq \mu(\mathcal{L}_{f,x})$. In addition, for all $k \geq 1$, $\mathcal{L}_{f,(1+\frac{1}{k})x} \subset \{y \in \mathbb{R}^n, f(x) \leq f(y) \leq f(2x)\} \subset \mathcal{L}^{\leq}_{f,2x}$, because if $x \neq 0$, f_x is strictly increasing thanks to Theorem 4. And the same theorem induces that $\mathcal{L}^{\leq}_{f,x}$ is compact and hence $\mu\left(\mathcal{L}^{\leq}_{f,x}\right) < \infty$. It follows that $\sum_{k=1}^{\infty} \mu(\mathcal{L}_{f,x}) \leq \sum_{k=1}^{\infty} \mu\left(\mathcal{L}_{f,(1+\frac{1}{k})x}\right) \leq \mu\left(\mathcal{L}^{\leq}_{f,2x}\right) < \infty$. Hence, $\mu(\mathcal{L}_{f,x}) = 0$. □ □

3.4.4 Balls Containing and Balls Contained in Sublevel Sets

The sublevel sets of continuous PH functions include and are embedded in balls whose construction is scaling-invariant. Given that continuous SI functions are monotonic transformation of PH functions, those properties are naturally transferred to SI functions. This is what we formalize in this section.

From the definition of a PH function with degree α, for all $x \neq 0$ we have $p(x) = \|x\|^\alpha p\left(x/\|x\|\right)$ for all $x \neq 0$. Therefore, p is continuous on $\mathbb{R}^n \setminus \{0\}$ if and only if p is continuous on $\mathcal{S}_1$. For such p, we denote $m_p = \min_{x \in \mathcal{S}_1} p(x)$, and $M_p = \max_{x \in \mathcal{S}_1} p(x)$. We have the following propositions:

Proposition 8 ([24, Lemma 2.8])**.** *Let p be a PH function with degree α such that $p(x) > 0$ for all $x \neq 0$. Assume that p is continuous on $\mathcal{S}_1$, then for all $x \neq 0$, the following holds*

$$\|x\| m_p^{1/\alpha} \leq p(x)^{1/\alpha} \leq \|x\| M_p^{1/\alpha} \quad . \tag{3.7}$$

Proposition 9 ([24, Lemma 2.9])**.** *Let p be a PH function with degree α such that $g(x) > 0$ for all $x \neq 0$. Assume that p is continuous on $\mathcal{S}_1$. Then for all $\rho > 0$, the ball centered in 0 and of radius ρ is included in the sublevel set of degree $\rho^\alpha M_p$, i.e. $\mathcal{B}(0, \rho) \subset \mathcal{L}^{\leq}_{p, \rho x_{M_p}}$, with $p(x_{M_p}) = M_p$. For all $x \neq 0$, the sublevel set of degree $p(x)$ is included into the ball centered in 0 and of radius $(p(x)/m_p)^\alpha$, i.e.*

$$\mathcal{L}^{\leq}_{p,x} \subset \mathcal{B}\left(0, \left(\frac{p(x)}{m_p}\right)^\alpha\right).$$

We can generalize both propositions to continuous scaling-invariant functions using Theorem 1.

Proposition 10. *Let f be a continuous SI function such that $f(x) > 0$ for all $x \neq 0$. Then there exist an increasing homeomorphism φ on $\mathbb{R}_+$ and two positive numbers $0 < m \leq M$ such that*

(i) *for all $x \neq 0$, $\varphi\left(m\|x\|\right) \leq f(x) \leq \varphi\left(M\|x\|\right)$,*

(ii) *for all $\rho > 0$, the ball centered in 0 and of radius ρ is included in the sublevel set of degree $\varphi(\rho\varphi^{-1}(M))$, i.e. $\mathcal{B}(0, \rho) \subset \mathcal{L}^{\leq}_{f, \rho x_M}$ with $f(\rho x_M) = \varphi\left(\rho\varphi^{-1}(M)\right)$.*

(iii) *for all $x \neq 0$, the sublevel set of degree $f(x)$ is included into the ball centered in 0 and of radius $\dfrac{\varphi^{-1}\left(f(x)\right)}{\varphi^{-1}(m)}$, i.e.*

$$\mathcal{L}^{\leq}_{f,x} \subset \mathcal{B}\left(0, \frac{\varphi^{-1}\left(f(x)\right)}{\varphi^{-1}(m)}\right). \tag{3.8}$$

Proof. Thanks to Theorem 1, we can write $f = f_{x_0} \circ p$ where $x_0 \neq 0$, p PH_1 φ defined as f_{x_0} is an increasing homeomorphism. Then $p = \varphi^{-1} \circ f$ and hence verifies: for all $x \neq 0$, $p(x) > 0$. Define m and M as $m = \varphi(m_p)$ and $M = \varphi(M_p)$, where $m_p = \min_{x \in \mathcal{S}_1} p(x)$ and $M_p = \max_{x \in \mathcal{S}_1} p(x)$.

For $x \neq 0$, Proposition 8 ensures that $m_p\|x\| \leq p(x) \leq M_p\|x\|$. Taking the image of this equation with respect to φ proves (i).

For all $\rho > 0$, $\mathcal{B}(0,\rho) \subset \mathcal{L}^{\leq}_{p,\rho x_{M_p}}$, with $p(x_{M_p}) = M_p$. Since sublevel sets are invariant with respect to an increasing bijection, it follows that $\mathcal{L}^{\leq}_{p,\rho x_{M_p}} = \mathcal{L}^{\leq}_{f,\rho x_{M_p}}$. In addition, $f(\rho x_{M_p}) = \varphi(p(\rho x_{M_p})) = \varphi(\rho p(x_{M_p})) = \varphi(\rho\varphi^{-1}(M))$ such that we have proven (ii).

Let $x \neq 0$. Again by invariance of the sublevel set, $\mathcal{L}^{\leq}_{p,x} = \mathcal{L}^{\leq}_{f,x}$. And Proposition 9 says that $\mathcal{L}^{\leq}_{p,x} \subset \mathcal{B}\left(0, \frac{p(x)}{m_p}\right)$. We obtain the results with the facts that $p = \varphi^{-1} \circ f$ and $m_p = \varphi^{-1}(m)$. □ □

3.4.5 A Generalization of a Weak Formulation of Euler's Homogeneous Function Theorem

For a function $p : \mathbb{R}^n \to \mathbb{R}$ continuously differentiable on $\mathbb{R}^n \setminus \{0\}$, Euler's homogeneous function theorem states that there is equivalence between p is PH with degree α and for all $x \neq 0$

$$\alpha p(x) = \nabla p(x) \cdot x \quad . \tag{3.9}$$

If in addition p is continuously differentiable in zero, then $\alpha p(0) = 0 = \nabla p(0) \cdot 0$. Along with (3.9), this latter equation implies that at each point y of a level set $\mathcal{L}_{p,x}$, the scalar product between $\nabla p(y)$ and y is constant equal to $\nabla p(x) \cdot x$ or that the level sets of p and of the function $x \mapsto \nabla p(x) \cdot x$ are the same, that is, the level sets of a continuously differentiable PH function satisfy

$$\mathcal{L}_{p,x} = \mathcal{L}_{z \mapsto \nabla p(z) \cdot z, x} = \{y \in \mathbb{R}^n, \nabla p(y) \cdot y = \nabla p(x) \cdot x\} \quad . \tag{3.10}$$

We call this a weak formulation of Euler's homogeneous function theorem.

If f is a continuous SI function, we can write f as $\varphi \circ p$ where p is PH and φ is a homeomorphism, according to Theorem 1. We have the following proposition in the case where φ and p are also continuously differentiable.

Proposition 11. *Let f be a continuously differentiable SI function that can be written as $\varphi \circ p$ where p is PH_α, φ is a homeomorphism, and φ and φ^{-1} are continuously differentiable (and thus p is continuously differentiable). Then for all $x \in \mathbb{R}^n$,*

$$\nabla f(x) \cdot x = \alpha\, \varphi'(p(x))\, p(x) \quad . \tag{3.11}$$

Proof. Since $p = \varphi^{-1} \circ f$, it is continuously differentiable. From the chain rule, for all $x \in \mathbb{R}^n$: $\nabla f(x) \cdot x = \varphi'(p(x)) \nabla p(x) \cdot x = \alpha \varphi'(p(x)) p(x)$. The last equality results from the Euler's homogeneous theorem applied to p. □ □

Yet, the assumptions of the previous proposition are not necessarily satisfied when f is a continuously differentiable SI function. Indeed, we exhibit in the next proposition an example of a SI and continuously differentiable function f such that $f = \varphi \circ p$ but either p or φ is non-differentiable.

Proposition 12. *Define* $\varphi : t \mapsto \int_0^t \frac{1}{1+\log^2(u)} \mathrm{d}u$ *on* $\mathbb{R}_+$ *and* $p : x \mapsto |x_1|$. *Then* $f = \varphi \circ p$ *is continuously differentiable and SI. Yet, for any* $\tilde{\varphi}$ *strictly increasing and* $\tilde{p}$ *PH such that* $f = \tilde{\varphi} \circ \tilde{p}$ *(including* φ *and* p *above), either* $\tilde{p}$ *is not differentiable on any point of the set* $\{x;\, x_1 = 0\}$ *or* $\tilde{\varphi}$ *is not differentiable at* 0.

Proof. Let us prove that f is continuously differentiable. For $x \neq 0$, $\nabla f(x) = \frac{1}{1+\log^2(|x_1|)} \frac{x_1}{|x_1|} e_1$ where e_1 is the unit vector $(1, 0, \ldots, 0)$. Then $\lim_{x \to 0} \nabla f(x)$ exists and is equal to 0, hence f is continuously differentiable.

Assume that $(\tilde{\varphi}, \tilde{p})$ is such that $\varphi \circ p = \tilde{\varphi} \circ \tilde{p}$, with $\tilde{\varphi}$ strictly increasing and $\tilde{p}$ PH_α. Denote $\psi = \tilde{\varphi}^{-1} \circ \varphi$. For all $\lambda > 0$ and $x \in \mathbb{R}^n$, $\psi(\lambda p(x)) = \psi(p(\lambda x)) = \tilde{p}(\lambda x) = \lambda^\alpha \psi(p(x))$. Therefore ψ is PH_α on $\mathrm{Im}(p) = \mathbb{R}_+$, hence for all $t > 0$, $\psi(t) = t^\alpha \psi(1)$. Then up to a positive constant multiplicative factor, $\tilde{p}(x) = |x_1|^\alpha$ and $\tilde{\varphi}(t) = \varphi(t^{1/\alpha})$. And then if $\tilde{p}$ is differentiable, we necessarily have that $\alpha > 1$.

In the case where $\alpha > 1$, for all $t > 0$, $\tilde{\varphi}'(t) = \frac{1}{\alpha} \frac{t^{\frac{1}{\alpha}-1}}{1+\log^2(t^{1/\alpha})}$ and then $\tilde{\varphi}$ is not differentiable at 0. □ □

Yet we can prove that for all continuously differentiable SI functions, the level set of f going through x, i.e. $\mathcal{L}_{f,x}$ is included in the level set of $z \mapsto \nabla f(z) \cdot z$ going through x.

Lemma 3. *For a continuously differentiable SI function* f *and for* $x \in \mathbb{R}^n$,

$$\mathcal{L}_{f,x} \subset \mathcal{L}_{z \mapsto \nabla f(z) \cdot z, x} = \{y \in \mathbb{R}^n, \nabla f(y) \cdot y = \nabla f(x) \cdot x\}. \tag{3.12}$$

That is, each level set of f *has a single value of* $\nabla f(x) \cdot x$ *while also different level sets of* f *can have the same value of* $\nabla f(x) \cdot x$.

Proof. Let $y \in \mathcal{L}_{f,x}$. Since $f(y) = f(x)$, then for all $t \geq 0$, $f(ty) = f(tx)$. We define the function h on $\mathbb{R}_+$ such that for all $t \geq 0$, $h(t) = f(tx) - f(ty)$. Then h is the zero function, so is its derivative: $h'(t) = \nabla f(tx) \cdot x - \nabla f(ty) \cdot y = 0$ for all $t \geq 0$. In particular we have the result for $t = 1$. □ □

We exhibit in the next proposition a continuously differentiable SI function where the inclusion in the above lemma is strict (another example is Lemma 4).

Proposition 13. *Let* p *be the PH*$_2$ *function* $x \in \mathbb{R}^n \mapsto \|x\|^2$ *and* φ *the strictly monotonic function* $\varphi(t) = \exp(-t)$ *for all* $t \geq 0$. *Then* $f : x \mapsto \varphi(p(x)) = \exp(-\|x\|^2)$ *is continuously differentiable. For any* $0 < r < 1$, *there is a unique* $s > 1$ *such that for any* $x \in \mathcal{S}_r$:

$$\mathcal{L}_{z \mapsto \nabla f(z) \cdot z, x} = \mathcal{L}_{f,x} \cup \mathcal{L}_{f, \frac{s}{r} x}$$

where $\mathcal{L}_{f,x}$ *and* $\mathcal{L}_{f, \frac{s}{r} x}$ *are disjoint.*

Proof. Remark that the transformation could be chosen to obtain a degree equal to 1 as in Theorem 1, but the differentiability of p would not be guaranteed.

We notice that $t \to t\varphi'(t)$ is not injective on $\mathbb{R}_+$. It is injective on $[0,1)$ and on $[1,\infty)$ and for any $0 < r < 1$ there is a unique $s > 1$ such that

$$r^2\varphi'(r^2) = s^2\varphi'(s^2). \tag{3.13}$$

We will prove that for any such r, s and any $x \in \mathcal{S}_r$,

$$\{y \in \mathbb{R}^n, \nabla f(y)\cdot y = \nabla f(x)\cdot x\} = \mathcal{L}_{f,x} \cup \mathcal{L}_{f,\frac{s}{r}x} \tag{3.14}$$

Let $x \in \mathbb{R}^n$ such that $\|x\| = r$. By the chain rule, for all $y \in \mathbb{R}^n$ we have

$$\nabla f(y)\cdot y = \varphi'(p(y))\nabla p(y)\cdot y = 2\varphi'(p(y))p(y).$$

Therefore $y \in \{y \in \mathbb{R}^n, \nabla f(y)\cdot y = \nabla f(x)\cdot x\}$ if and only if $\|y\|^2\varphi'(\|y\|^2) = r^2\varphi'(r^2)$. From (3.13), we know that this is possible only if $\|y\| = r$ or $\|y\| = s$, i.e. only if $f(y) = f(x)$ or $f(y) = f(\frac{s}{r}x)$. Hence the equality in (3.14).

We remark that $\mathcal{L}_{f,x}$ and $\mathcal{L}_{f,\frac{s}{r}x}$ are disjoint whenever $f(x) \neq f(\frac{s}{r}x)$. If $x \in \mathcal{S}_r$, $f(x) = e^{-r^2} \neq e^{-s^2} = f\left(\frac{s}{r}x\right)$ which implies that $\mathcal{L}_{f,x}$ and $\mathcal{L}_{f,\frac{s}{r}x}$ are disjoint. □ □

The non-injectivity of $t \to t\varphi'(t)$ is essential in the above example to obtain a non-strict inclusion in (3.12) for some SI functions. We obtain a weak formulation of Euler's homogeneous function theorem for some SI functions in the following proposition.

Proposition 14. *Let f be a continuously differentiable SI function that can be written as $\varphi \circ p$ where φ is a homeomorphism, p is PH$_1$ and φ and φ^{-1} are continuously differentiable. Assume that the function $\mathbb{R}_+ \ni t \longmapsto t\varphi'(t) \in \mathbb{R}$ is injective. Then for $x \in \mathbb{R}^n$,*

$$\mathcal{L}_{f,x} = \mathcal{L}_{z\mapsto\nabla f(z)\cdot z,x} \quad . \tag{3.15}$$

Proof. It follows from Proposition 11 that for all $x \in \mathbb{R}^n$, $\nabla f(x)\cdot x = \varphi'(p(x))p(x)$. Thanks to the bijectivity of φ along with the injectivity of $t \to t\varphi'(t)$, we have:

$$\varphi'(p(y))p(y) = \varphi'(p(x))p(x) \iff p(x) = p(y) \iff f(x) = f(y).$$

In other words, $\mathcal{L}_{f,x} = \{y \in \mathbb{R}^n, \nabla f(y)\cdot y = \nabla f(x)\cdot x\}$. □ □

3.4.6 Compact Neighborhoods of Level Sets with Non-vanishing Gradient

We prove in this section that any continuously differentiable SI function f with a unique global argmin has level sets, for example $\mathcal{L}_{f,z_0}$, such that for some compact neighborhood of the level set, $\mathcal{N} \supset \mathcal{L}_{f,z_0}$, the gradient does not vanish and $\nabla f(z)\cdot z > 0$ for all $z \in \mathcal{N}$.

For a continuously differentiable PH function p such that $p(x) > 0$ for all $x \neq 0$, *i.e.* such that 0 is the unique global argmin of p, this result is a consequence of Euler's homogeneous function theorem which implies that

$$\nabla p(x) \cdot x > 0 \text{ for all } x \neq 0 \ . \tag{3.16}$$

In particular, (3.16) is true on any compact neighborhood of any level set of p, if that compact does not contain 0.

We now remark that the property that $\nabla f \neq 0$ for all $x \neq 0$ is not necessarily true if f is a continuously differentiable SI function with a unique global argmin. Namely, f can have level sets that contain only saddle points.

Lemma 4. *Let $p(z) = \|z\|^2$ and $\varphi(t) = \int_0^t \sin^2(u)\mathrm{d}u$ for $t \geq 0$. Then $f = \varphi \circ p$ is a continuously differentiable SI function with a unique global argmin and an infinite number of z belonging to different level sets of f, such that $\nabla f(z) = 0$.*

Proof. The function φ is strictly increasing since $\sin^2$ is non-negative and has zeros on isolated points. Also, for all $t \geq 0$, $\varphi(t) = \frac{t}{2} - \frac{\sin(2t)}{4}$, where we use that $\cos(2t) = 1 - 2\sin^2(t)$.

For any natural integer n, $n\pi$ is a stationary point of inflection of $\varphi : \varphi'(n\pi) = 0$ and $\varphi''(t) = \sin(2t)$ has opposite signs in the neighborhood of $n\pi$. For all z with $\|z\|^2 \in \pi\mathbb{Z}_+$, $\nabla f(z) = \varphi'(g(z))\nabla g(z) = 2\varphi'(\|z\|^2)\ z = 0$.

Hence there exists an infinite number of level sets $\mathcal{L}_{f,z}$ for which $\nabla f(z) = 0$. □ □

Yet, a consequence of Theorem 4 and Lemma 3 is the existence of a level set of f such that $\nabla f(z) \cdot z > 0$ for all z in that level set as shown in the next proposition

Proposition 15. *Let f be a continuously differentiable SI function with 0 as unique global argmin. There exists $z_0 \in \overline{\mathcal{B}}_1$ with $\mathcal{L}_{f,z_0} \subset \overline{\mathcal{B}}_1$, such that for all $z \in \mathcal{L}_{f,z_0}$, $\nabla f(z) \cdot z > 0$.*

Proof. Since f is a continuous SI function, we have all the equivalences in Theorem 4.

Inside the proof of Theorem 4, we have shown that there exists $s \in \mathcal{S}_1$ such that $\mathcal{L}^{\leq}_{f,s} \subset \overline{\mathcal{B}}_1$, with $f(s) = \min_{z \in \mathcal{S}_1} f(z)$. Since f_s is strictly increasing and differentiable, there exists $t \in (0, 1]$ such that $f_s'(t) > 0$. Let us denote $z_0 = ts$. We have that $\mathcal{L}_{f,z_0} \subset \mathcal{L}^{\leq}_{f,s} \subset \overline{\mathcal{B}}_1$. And with the chain rule, $0 < f_s'(t) = \nabla f(z_0) \cdot \frac{z_0}{t}$. Therefore along with Lemma 3, it follows that for all $z \in \mathcal{L}_{f,z_0}$, $\nabla f(z) \cdot z = \nabla f(z_0) \cdot z_0 > 0$. □ □

From the uniform continuity of $z \mapsto \nabla f(z) \cdot z$ on a compact we deduce the announced result.

Proposition 16. *Let f be a continuously differentiable SI function with 0 as unique global argmin. There exists $\delta > 0$, $z_0 \in \overline{\mathcal{B}}_1$ with $\mathcal{L}_{f,z_0} \subset \overline{\mathcal{B}}_1$ such that for all $z \in \mathcal{L}_{f,z_0} + \overline{\mathcal{B}(0,\delta)}$, $\nabla f(z) \cdot z > 0$.*

Proof. Since $\nabla f(z) \cdot z > 0$ for all z in the compact $\mathcal{L}_{f,z_0}$, then $z \mapsto \nabla f(z) \cdot z$ has a positive minimum (that is reached) denoted by $\epsilon = \min_{z \in \mathcal{L}_{f,z_0}} \nabla f(z) \cdot z > 0$. The continuous function $z \mapsto \nabla f(z) \cdot z$ is uniformly continuous on the compact $\mathcal{L}_{f,z_0} + \overline{\mathcal{B}(0,1)}$, therefore there exists a positive number $\delta < 1$ such that if $y, z \in \mathcal{L}_{f,z_0} + \overline{\mathcal{B}(0,1)}$ with $\|y - z\| \leq \delta$ then $|\nabla f(z) \cdot z - \nabla f(y) \cdot y| < \frac{\epsilon}{2}$. Then for all $z \in \mathcal{L}_{f,z_0} + \overline{\mathcal{B}(0,\delta)}$, there exists $y \in \mathcal{L}_{f,z_0}$ such that $|\nabla f(z) \cdot z - \nabla f(y) \cdot y| < \frac{\epsilon}{2}$. Then $\nabla f(z) \cdot z > \nabla f(y) \cdot y - \frac{\epsilon}{2} \geq \frac{\epsilon}{2} > 0$. Hence $z \mapsto \nabla f(z) \cdot z$ is positive on the compact set $\mathcal{L}_{f,z_0} + \overline{\mathcal{B}(0,\delta)}$. □ □

3.5 Summary and Conclusion

This paper reveals that *continuous* scaling-invariant functions are strictly monotonic transformations of *continuous* positively homogeneous functions. Moreover, we present necessary and sufficient conditions for any scaling-invariant function to be a strictly monotonic transformation of a positively homogeneous function. The derivation is solely based on analyzing restrictions to the half-lines starting from zero that need to be strictly monotonic on a nontrivial interval (or entirely flat). We also highlight counter-intuitive examples of scaling-invariant functions that are not monotonic on any nontrivial interval.

We then present different properties of the level sets of a scaling-invariant function. In particular, Proposition 16 shows that continuously differentiable scaling-invariant functions with a unique argmin have a compact level set in a compact neighborhood with non-vanishing gradient. The level set intersects any half-line with origin zero at a single point—forming a "star-shaped" manifold.

Scaling-invariant functions play a central role in the analysis of the convergence of some comparison-based stochastic optimization algorithms [25]. On this function class, for some translation and scale invariant comparison-based algorithms, linear convergence can be deduced when analyzing the stability of a normalized process[4]. When linear convergence occurs, the step-size and the distance of the current solution to the optimum decrease geometrically fast to zero at the same (linear) rate.

[4] In the case of step-size adaptive algorithms where the state of the algorithm equals a current solution and a step-size, the normalized process equals to the solution minus the reference point $x^\star$ of the scaling-invariant function, divided by the step-size. Stability of the normalized process is key to imply linear convergence of the adaptive algorithm.

A stability analysis leading to linear convergence can be carried out for composites of strictly increasing functions with continuously differentiable scaling-invariant functions. To obtain basic stability properties deduced from a connection to a deterministic control model [47], one can exploit that these functions have Lebesgue negligible level sets as a consequence of Proposition 7. In addition, the stability study relies on proving that when the normalized process diverges, the step-size multiplicative factor converges in distribution to the factor on nontrivial linear functions. The proof exploits level set properties shown in Proposition 16 and Corollary 3.

Chapter 4

Global linear convergence of Evolution Strategies with recombination on scaling-invariant functions

Contents

Note: the content of this chapter is submitted in the Journal of Global Optimization, by the authors Cheikh Toure, Anne Auger and Nikolaus Hansen.

Evolution Strategies (ES) are stochastic derivative-free optimization algorithms whose most prominent representative, the CMA-ES algorithm, is widely used to solve difficult numerical optimization problems. We provide the first rigorous investigation of the linear convergence of step-size adaptive ES involving a population and recombination, two ingredients crucially important in practice to be robust to local irregularities or multimodality. Our methodology relies on investigating the stability of a Markov chain associated to the algorithm. Our stability study is crucially based on recent developments connecting the stability of deterministic control models to the stability of associated Markov chains.

We investigate convergence on composites of strictly increasing functions with continuously differentiable scaling-invariant functions with a global optimum. This function class includes functions with non-convex sublevel sets and discontinuous functions. We prove the existence of a constant r such that the logarithm of the distance to the optimum divided by the number of iterations of step-size adaptive ES with weighted recombination converges to r. The constant is given as an expectation with respect to the stationary distribution of a Markov chain—its sign allows to infer linear convergence or divergence of the ES and is found numerically.

Our main condition for convergence is the increase of the expected log step-size on linear functions. In contrast to previous results, our condition is equivalent to the almost sure geometric divergence of the step-size.

4.1 Introduction

Evolution Strategies (ES) are stochastic numerical optimization algorithms introduced in the 70's [159,160,172,173]. They aim at optimizing an objective function $f : \mathbb{R}^n \to \mathbb{R}$ in a so-called zero-order black-box scenario where gradients are not available and only *comparisons* between f-values of candidate solutions are used to update the state of the algorithm. Evolution Strategies sample candidate solutions from a multivariate normal distribution parametrized by a mean vector and a covariance matrix. The mean vector represents the incumbent or current favorite solution while the covariance matrix determines the geometric shape of the sampling probability distribution. In adaptive ES, not only the mean vector but also the covariance matrix is adapted in each iteration. Covariance matrices can be seen as encoding a metric such that Evolution Strategies that adapt a full covariance matrix are variable metric algorithms [180].

Among ESs, the covariance-matrix-adaptation ES (CMA-ES) [88, 95] is nowadays recognized as state-of-the-art to solve difficult numerical optimization problems that can typically be non-convex, non-linear, ill-conditioned, non-separable, rugged or multi-modal. Adaptation of the full covariance matrix is crucial to solve ill-conditioned, non-separable problems. Up to a multiplicative factor that converges to zero, the covariance matrix becomes on strictly convex-quadratic objective functions close to the inverse Hessian of the function [85].

The CMA-ES algorithm follows a $(\mu/\mu_w, \lambda)$-ES algorithmic scheme where from the offspring population of λ candidate solutions sampled at each iteration the $\mu \approx \lambda/2$ best solutions—the new parent population—are recombined as a weighted sum to define the new mean vector of the multivariate normal distribution. On a unimodal spherical function, the optimal step-size, $i.e.$ the standard deviation that should be used to sample each coordinate of the candidate solutions, depends monotonously on μ [160]. Hence, increasing the population size makes the search less local while preserving a close-to-optimal convergence rate per function evaluation as long as λ remains moderately large [12, 13, 90]. This remarkable theoretical property implies robustness and partly explains why on many multi-modal test functions increasing λ empirically increases the probability to converge to the global optimum [93]. The robustness when increasing λ and the inherent parallel nature of Evolution Strategies are two key features behind their success for tackling difficult black-box optimization problems.

Convergence is a central question in optimization. For comparison-based algorithms like Evolution Strategies, linear convergence (where the distance to the optimum decreases geometrically) is the fastest possible convergence [114, 182]. Gradient methods also converge linearly on strongly convex functions [147, Theorem 2.1.15].

We have ample empirical evidence that adaptive Evolution Strategies converge

linearly on wide classes of functions [90, 96, 105, 162]. Yet, proving linear convergence is known to be difficult. So far, linear convergence could be proven only for step-size adaptive algorithms where the covariance matrix equals a scalar times the identity [19, 24, 106–108, 110] or a scalar times a covariance matrix with eigenvalues upper bounded and bounded away from zero [3]. In addition, these proofs require the parent population size to be one.

In this context, we analyze here for the first time the linear convergence of a step-size adaptive ES with a parent population size greater than one and recombination, following a $(\mu/\mu_w, \lambda)$-ES framework. As a second novelty, we model the step-size update by a generic function and thereby also encompass the step-size updates in the CMA-ES algorithm [88] (however with a specific parameter setting which leads to a reduced state-space) and in the xNES algorithm [70].

Our proofs hold on composites of strictly increasing functions with either continuously differentiable scaling-invariant functions with a unique argmin or nontrivial linear functions. This class of function includes discontinuous functions, functions with infinite many critical points, and functions with non-convex sublevel sets. It does not include functions with more than one (local or global) optimum.

In this paper, we use a methodology formalized in [25] and previously used in [19, 24]. The methodology is based on analyzing the stochastic process defined as the difference between the mean vector and a reference point (often the optimum of the problem), normalized by the step-size. This construct is a viable model of the underlying (translation and scale invariant) algorithm when optimizing scaling-invariant functions, in which case the stochastic process is also a Markov chain and here referred to as *σ-normalized Markov chain*. This chain is *homogeneous* as a consequence of three crucial invariance properties of the ES algorithms: translation invariance, scale invariance, and invariance to strictly increasing transformations of the objective function. Proving *stability* of the σ-normalized Markov chain (φ-irreducibility, Harris recurrence, positivity) is key to obtain almost sure *linear behavior* of the algorithm [25]. The sign and value of the convergence or divergence rate can however only be obtained from elementary Monte Carlo simulations.

One main difficulty is to prove φ-irreducibility, because the updates of mean and step-size in the analyzed algorithms are coupled [150]. We solve this difficulty by using recent tools that connect the stability of a deterministic control model to the stability of an associated Markov chain [47]. Other stability properties are established using standard Lyapunov drift conditions [140].

This paper is organized as follows. We present in Section 4.2 the algorithm framework, the assumptions on the algorithm and the class of objective functions where the convergence analysis is carried out. In Section 4.3 we present the main results of the paper. In Section 4.4, we present the methodology and the Markov chain notions needed for obtaining the proofs. In Section 4.5 we establish different stability properties on the σ-normalized Markov chain. We prove the main results in Section 4.6. In

Section 4.7, we highlight previous works related to this paper.

Notation

We denote by $\mathbb{R}_+$ the set of non-negative real numbers and by $\mathbb{N}$ the set of non-negative integers. The Euclidean norm is denoted by $\|\cdot\|$. For $x \in \mathbb{R}^n$ and $\rho > 0$, $\mathbf{B}(x,\rho) = \{y \in \mathbb{R}^n; \|x-y\| < \rho\}$ and $\overline{\mathbf{B}(x,\rho)}$ is its closure.

For a set A, we denote by A^c the complement of A. For a topological space $\mathcal{Z}$, we denote its Borel sigma-field by $\mathcal{B}(\mathcal{Z})$. For a signed measure ν, we denote for any real-valued function g, $\mathbb{E}_\nu(g) = \int g(z)\,\nu(\mathrm{d}z)$. For a positive function h, we denote by $\|\cdot\|_h$ the norm on signed measures on $\mathcal{B}(\mathcal{Z})$ defined for all signed measure ν as $\|\nu\|_h = \sup_{|g|\le h} |\mathbb{E}_\nu(g)|$. If $(\mathcal{Z}_1, \mathcal{B}(\mathcal{Z}_1), \pi_1)$ and $(\mathcal{Z}_2, \mathcal{B}(\mathcal{Z}_2), \pi_2)$ are two measure spaces, we denote by $\pi_1 \times \pi_2$ the product measure on the product measurable space $(\mathcal{Z}_1 \times \mathcal{Z}_2,\, \mathcal{B}(\mathcal{Z}_1) \otimes \mathcal{B}(\mathcal{Z}_2))$ where $\otimes$ is the tensor product.

We denote by $\mathcal{N}$ the standard normal distribution. If $x \in \mathbb{R}^m$ and C is a covariance matrix of dimension $m \times m$, we denote by $\mathcal{N}(x, C)$ the multivariate normal distribution with mean x and covariance matrix C. If C is the identity matrix, $\mathcal{N}_m = \mathcal{N}(0, C)$ denotes the standard multivariate normal distribution in dimension m. We denote by $p_{\mathcal{N}_m}$ its probability density function.

We denote by $\|\cdot\|_\infty$ the infinity norm on a space of bounded functions. For a matrix T, we denote by $T^\intercal$ the transpose of T. For $p \in \mathbb{N} \setminus \{0\}$, we denote an element u of $\mathbb{R}^{pm}$ as $u = (u^1, \dots, u^m)$ where $u^i \in \mathbb{R}^p$ for $i = 1, \dots, m$. If $m = 1$, we write that $u = (u^1) = u^1$. For $w \in \mathbb{R}^m$ and $u \in \mathbb{R}^{pm}$, we denote $\sum_{i=1}^m w_i u^i$ as $w^\intercal u$. For an objective function $f : \mathbb{R}^n \to \mathbb{R}$ and an element $z \in \mathbb{R}^n$, we denote by $\mathcal{L}_{f,z}$ the level set $\{y \in \mathbb{R}^n ; f(y) = f(z)\}$.

We refer to a non-zero linear function as a nontrivial linear function.

4.2 Algorithm framework and class of functions studied

We present in this section our step-size adaptive algorithm framework, the assumptions on the algorithm and the function class considered. We also present preliminary results.

In the following, we consider an abstract measurable space $(\Omega, \mathcal{F})$ and a probability measure P so that $(\Omega, \mathcal{F}, P)$ is a measure space.

4.2.1 The $(\mu/\mu_w, \lambda)$-ES algorithm framework

We introduce the algorithm framework studied in this work, specifically, step-size adaptive evolution strategies using weighted multi-recombination, referred to as step-size adaptive $(\mu/\mu_w, \lambda)$-ES. Given a positive integer n and a function $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ to be minimized, the sequence of states of the algorithm is represented by $\{(X_k, \sigma_k)\,; k \in \mathbb{N}\}$ where at iteration k, $X_k \in \mathbb{R}^n$ is the incumbent (the favorite solution) and the positive scalar σ_k is the step-size. The incumbent is also considered as a current estimate of the optimum. We fix positive integers λ and μ such that $\mu \leq \lambda$.

Let $(X_0, \sigma_0) \in \mathbb{R}^n \times (0, \infty)$ and $U = \{U_{k+1} = (U_{k+1}^1, \ldots, U_{k+1}^\lambda)\,; k \in \mathbb{N}\}$ be a sequence of independent and identically distributed (i.i.d.) random inputs independent from (X_0, σ_0), where for all $k \in \mathbb{N}$, $U_{k+1} = (U_{k+1}^1, \ldots, U_{k+1}^\lambda)$ is composed of λ independent random vectors following a standard multivariate normal distribution $\mathcal{N}_n$. Given (X_k, σ_k) for $k \in \mathbb{N}$, we consider the following iterative update. First, we define λ candidate solutions as

$$X_{k+1}^i = X_k + \sigma_k\, U_{k+1}^i \quad \text{for } i = 1, \ldots, \lambda. \tag{4.1}$$

Second, we evaluate the candidate solutions on the objective function f. We then denote an f-sorted permutation of $\left(X_{k+1}^1, \ldots, X_{k+1}^\lambda\right)$ as $\left(X_{k+1}^{1:\lambda}, \ldots, X_{k+1}^{\lambda:\lambda}\right)$ such that

$$f(X_{k+1}^{1:\lambda}) \leq \cdots \leq f(X_{k+1}^{\lambda:\lambda}) \tag{4.2}$$

and thereby define the indices $i:\lambda$. To break possible ties, we require that $i:\lambda < j:\lambda$ if $f(X_{k+1}^i) = f(X_{k+1}^j)$ and $i < j$. The sorting indices $i:\lambda$ are also used for the σ-normalized difference vectors U_{k+1}^i in that

$$U_{k+1}^{i:\lambda} = \frac{X_{k+1}^{i:\lambda} - X_k}{\sigma_k} \quad .$$

Accordingly, we define the *selection function* α_f of $z \in \mathbb{R}^n$ and $u = (u^1, \ldots, u^\lambda) \in \mathbb{R}^{n\lambda}$ to yield the sorted sequence of the difference vectors as

$$\alpha_f(z, u) = (u^{1:\lambda}, \ldots, u^{\mu:\lambda}) \in \mathbb{R}^{n\mu}, \tag{4.3}$$

with $f(z+u^{1:\lambda}) \leq \cdots \leq f(z+u^{\lambda:\lambda})$ and the above tie breaking. For $\lambda = 2$ and $\mu = 1$, the selection function has the simple expression $\alpha_f(z, (u^1, u^2)) = (u^1 - u^2)\mathbb{1}_{\{f(z+u^1) \leq f(z+u^2)\}} + u^2$.

By definition, we have for $k \in \mathbb{N}$, $\alpha_f(X_k, \sigma_k U_{k+1}) = \left(\sigma_k U_{k+1}^{1:\lambda}, \ldots, \sigma_k U_{k+1}^{\mu:\lambda}\right)$ so that

$$\frac{\alpha_f(X_k, \sigma_k U_{k+1})}{\sigma_k} = \left(U_{k+1}^{1:\lambda}, \ldots, U_{k+1}^{\mu:\lambda}\right). \tag{4.4}$$

However, α_f is not a homogeneous function in general, because the indices $i:\lambda$ in (4.4) depend on f and hence on α_f and hence on σ_k.

The update of the state of the algorithm uses the objective function only through the above selection function. This selection function is invariant to strictly increasing transformations of the objective function as formalized in the next lemma. Indeed, the selection is determined through the ranking of candidate solutions in (4.2) which is the same when we optimize $g \circ f$ instead of f given g is strictly increasing. We talk about comparison-based algorithms in this case.

Lemma 5. *Let g be a function. Define f as $f = \varphi \circ g$ where φ is strictly increasing. Then $\alpha_f = \alpha_g$.*

To update the mean vector X_k, we consider a weighted average of the $\mu \leq \lambda$ best solutions $\sum_{i=1}^{\mu} w_i X_{k+1}^{i:\lambda}$ where $w = (w_1, \ldots, w_\mu)$ is a non-zero vector. With only positive weights summing to one, this weighted average is situated in the convex hull of the μ best points.

The next incumbent X_{k+1} is constructed by combining X_k and $\sum_{i=1}^{\mu} w_i X_{k+1}^{i:\lambda}$ as

$$X_{k+1} = \left(1 - \sum_{i=1}^{\mu} w_i\right) X_k + \sum_{i=1}^{\mu} w_i X_{k+1}^{i:\lambda} \tag{4.5}$$

$$= X_k + \sum_{i=1}^{\mu} w_i \left(X_{k+1}^{i:\lambda} - X_k\right) = X_k + \sigma_k \sum_{i=1}^{\mu} w_i U_{k+1}^{i:\lambda} \quad . \tag{4.6}$$

Positive weights with small indices move the new mean vector towards the better solutions, hence these weights should generally be large. In evolution strategies, the weights are always non-increasing in i. With the notable exception of Natural Evolution Strategies ([70] and related works), all weights are positive. In practice, $\sum_{i=1}^{\mu} w_i$ is often set to 1 such that the new mean vector is the weighted average of the μ best solutions. Proposition 19 describes (generally weak) explicit conditions for the weights under which our results hold.

We write the step-size update in an abstract manner as

$$\sigma_{k+1} = \sigma_k \, \Gamma\left(U_{k+1}^{1:\lambda}, \ldots, U_{k+1}^{\mu:\lambda}\right) \tag{4.7}$$

where $\Gamma : \mathbb{R}^{n\mu} \to \mathbb{R}_+ \setminus \{0\}$ is a measurable function. This generic step-size update is by construction scale-invariant, which is key for our analysis [25]. The update of the mean vector and of the step-size are both functions of the f-sorted sampled vectors $(U_{k+1}^{1:\lambda}, \ldots, U_{k+1}^{\mu:\lambda})$.

Using (4.4), we rewrite the algorithm framework (4.6) and (4.7) for all k as:

$$X_{k+1} = X_k + \sum_{i=1}^{\mu} w_i \left[\alpha_f(X_k, \sigma_k U_{k+1})\right]_i = X_k + w^\top \alpha_f(X_k, \sigma_k U_{k+1}) \tag{4.8}$$

$$\sigma_{k+1} = \sigma_k \, \Gamma\left(\frac{\alpha_f(X_k, \sigma_k U_{k+1})}{\sigma_k}\right) \tag{4.9}$$

with $U = \{U_{k+1}\,; k \in \mathbb{N}\}$ the sequence of identically distributed random inputs and $w \in \mathbb{R}^\mu \setminus \{0\}$.

4.2.2 Algorithms encompassed

The generic update in (4.7) or equivalently (4.9) encompasses the step-size update of the cumulative step-size adaptation evolution strategy ($(\mu/\mu_w, \lambda)$-CSA-ES) [25, 95] with cumulation factor set to 1 where for $d_\sigma > 0$, $w \in \mathbb{R}^\mu \setminus \{0\}$ and $u = (u^1, \ldots, u^\mu) \in \mathbb{R}^{n\mu}$,

$$\Gamma^0_{\mathrm{CSA1}}(u^1, \ldots, u^\mu) = \exp\left(\frac{1}{d_\sigma}\left(\frac{\|\sum_{i=1}^\mu w_i u^i\|}{\|w\|\,\mathbb{E}\left[\|\mathcal{N}_n\|\right]} - 1\right)\right) \quad . \tag{4.10}$$

The acronym CSA1 emphasizes that we only consider a particular case here: in the original CSA algorithm, the sum in (4.10) is an exponentially fading average of these sums from the past iterations with a smoothing factor of $1 - c_\sigma$. Equation (4.10) only holds when the cumulation factor c_σ is equal to 1, whereas in practice, $1/c_\sigma$ is between $\sqrt{n}/2$ and $n + 2$ (see [88] for more details). The damping parameter $d_\sigma \approx 1$ scales the change magnitude of $\log(\sigma_k)$.

Equation (4.10) increases the step-size if and only if the length of $\sum_{i=1}^\mu w_i U^{i:\lambda}_{k+1}$ is larger than the expected length of $\sum_{i=1}^\mu w_i U^i_{k+1}$ which is equal to $\|w\|\,\mathbb{E}\left[\|\mathcal{N}_n\|\right]$. Since the function Γ^0_{CSA1} is not continuously differentiable (an assumption needed in our analysis) we consider a version of the $(\mu/\mu_w, \lambda)$-CSA1-ES [15] that compares the square length of $\sum_{i=1}^\mu w_i U^{i:\lambda}_{k+1}$ to the expected square length of $\sum_{i=1}^\mu w_i U^i_{k+1}$ which is $n\|w\|^2$. Hence, we analyze for $d_\sigma > 0$, $w \in \mathbb{R}^\mu \setminus \{0\}$ and $u = (u^1, \ldots, u^\mu) \in \mathbb{R}^{n\mu}$:

$$\Gamma_{\mathrm{CSA1}}(u^1, \ldots, u^\mu) = \exp\left(\frac{1}{2d_\sigma n}\left(\frac{\|\sum_{i=1}^\mu w_i u^i\|^2}{\|w\|^2} - n\right)\right). \tag{4.11}$$

Another step-size update encompassed with (4.4) is given by the Exponential Natural Evolution Strategy (xNES) [25, 70, 149, 170] and defined for $d_\sigma > 0$, $w \in \mathbb{R}^\mu \setminus \{0\}$ and $u = (u^1, \ldots, u^\mu) \in \mathbb{R}^{n\mu}$ as

$$\Gamma_{\mathrm{xNES}}(u^1, \ldots, u^\mu) = \exp\left(\frac{1}{2d_\sigma n}\left(\sum_{i=1}^\mu \frac{w_i}{\sum_{j=1}^\mu |w_j|}\left(\|u^i\|^2 - n\right)\right)\right). \tag{4.12}$$

Both equations (4.11) and (4.12) correlate the step-size increment with the vector lengths of the μ best solutions. While (4.11) takes the squared norm of the weighted sum of the vectors, (4.12) takes the weighted sum of squared norms. Hence, correlations between the directions u^i affect only (4.11). Both equations are offset to become unbiased such that $\log \circ \Gamma$ is zero in expectation when $u^i \sim \mathcal{N}_n$ for all $1 \le i \le \lambda$, are i.i.d. random vectors.

4.2.3 Assumptions on the algorithm framework

We pose some assumptions on the algorithm (4.8) and (4.9) starting with assumptions on the step-size update function Γ.

A1. The function $\Gamma : \mathbb{R}^{n\mu} \to \mathbb{R}_+ \setminus \{0\}$ is continuously differentiable (C^1).

A2. Γ is *invariant under rotation* in the following sense: for all $n \times n$ orthogonal matrices T, for all $u = (u_1, \ldots, u_\mu) \in \mathbb{R}^{n\mu}$, $\Gamma(Tu_1, \ldots, Tu_\mu) = \Gamma(u)$.

A3. The function Γ is lower-bounded by a constant $m_\Gamma > 0$, that is for all $x \in \mathbb{R}^{n\mu}$, $\Gamma(x) \geq m_\Gamma$.

A4. $\log \circ \Gamma$ is $\mathcal{N}_{n\mu}$-integrable, that is, $\int |\log(\Gamma(u))| \, p_{\mathcal{N}_{n\mu}}(u) \mathrm{d}u < \infty$.

We can easily verify that Assumptions A1–A4 are satisfied for the $(\mu/\mu_w, \lambda)$-CSA1 and $(\mu/\mu_w, \lambda)$-xNES updates given in (4.11) and (4.12). More precisely, the following lemma holds.

Lemma 6. *The step-size update function Γ_{CSA1} defined in (4.11) satisfies Assumptions A1–A4. Endowed with non-negative weights $w_i \geq 0$ for all $i = 1, \ldots, \mu$, the step-size update function Γ_{xNES} defined in (4.12) satisfies Assumptions A1–A4.*

Proof. A1 and A4 are immediate to verify. For A2, the invariance under rotation comes from the norm-preserving property of orthogonal matrices. For all $u = (u_1, \ldots, u_\mu) \in \mathbb{R}^{n\mu}$, $\Gamma_{\mathrm{CSA1}}(u) \geq \exp\left(-\frac{1}{2d_\sigma}\right)$ such that Γ_{CSA1} satisfies A3. Similarly

$$\Gamma_{\mathrm{xNES}}(u) = \exp\left(-\frac{1}{2d_\sigma} \frac{\sum_{i=1}^{\mu} w_i}{\sum_{j=1}^{\mu} |w_j|} + \frac{1}{2d_\sigma n} \sum_{i=1}^{\mu} \frac{w_i}{\sum_{j=1}^{\mu} |w_j|} \|u^i\|^2\right).$$

Since all the weights are non-negative, $\frac{1}{2d_\sigma n} \sum_{i=1}^{\mu} w_i \|u^i\|^2 \geq 0$. And then $-\frac{1}{2d_\sigma} \sum_{i=1}^{\mu} w_i + \frac{1}{2d_\sigma n} \sum_{i=1}^{\mu} w_i \|u^i\|^2 \geq -\frac{1}{2d_\sigma} \sum_{i=1}^{\mu} w_i$. Therefore $\Gamma_{\mathrm{xNES}}(u) \geq \exp\left(-\frac{1}{2d_\sigma}\right)$ which does not depend on u, such that Γ_{xNES} satisfies A3. □

Assumptions A1–A4 are also satisfied for a constant function Γ equal to a positive number. When the positive number is greater than 1, our main condition for a linear behavior is satisfied, as we will see later on. Yet, the step-size of this algorithm clearly diverges geometrically.

We formalize now the assumption on the source distribution used to sample candidate solutions, as it was already specified when defining the algorithm framework.

A5. $U = \{U_{k+1} = \left(U_{k+1}^1, \ldots, U_{k+1}^\lambda\right) \in \mathbb{R}^{n\lambda} ; k \in \mathbb{N}\}$, see e.g. (4.1), is an i.i.d. sequence that is also independent from (X_0, σ_0), and for all natural integer k, U_{k+1} is an independent sample of λ standard multivariate normal distributions on $\mathbb{R}^n$ at time $k+1$.

The last assumption is natural as evolution strategies use predominantly Gaussian distributions[1]. Yet, we can replace the multivariate normal distribution by a distribution

[1] In Evolution Strategies, Gaussian distributions are mainly used for convenience: they are the natural choice to generate rotationally invariant random vectors. Several attempts have

with finite first and second moments and a probability density function of the form $x \mapsto \frac{1}{\sigma^n} g\left(\frac{\|x\|^2}{\sigma^2}\right)$ where $\sigma > 0$ and $g : \mathbb{R}_+ \to \mathbb{R}_+$ is C^1, non-increasing and submultiplicative in that there exists $K > 0$ such that for $t \in \mathbb{R}_+$ and $s \in \mathbb{R}_+$, $g(t+s) \leq K g(t) g(s)$ (such that Proposition 28 holds).

4.2.4 Assumptions on the objective function

We introduce in this section the assumptions needed on the objective function to prove the linear behavior of step-size adaptive $(\mu/\mu_w, \lambda)$-ES. Our main assumption is that the function is scaling-invariant. We remind that a function f is scaling-invariant [25] with respect to a reference point $x^\star$ if for all $\rho > 0$, $x, y \in \mathbb{R}^n$

$$f(x^\star + x) \leq f(x^\star + y) \iff f(x^\star + \rho x) \leq f(x^\star + \rho y) . \tag{4.13}$$

We pose one of the following assumptions on f:

F1. The function f satisfies $f = \varphi \circ g$ where φ is a strictly increasing function and g is a C^1 scaling-invariant function with respect to $x^\star$ and has a unique global argmin (that is $x^\star$).

F2. The function f satisfies $f = \varphi \circ g$ where φ is a strictly increasing function and g is a nontrivial linear function.

Assumption F1 is our core assumption for studying convergence: we assume scaling invariance and continuous differentiability not on f but on g where $f = \varphi \circ g$ such that the function f can be discontinuous (we can include jumps in the function via the function φ). Because ES are comparison-based algorithms and thus the selection function is identical on f or $g \circ f$ (see Lemma 5), our analysis is invariant if we carry it out on f or $g \circ f$. Strictly increasing transformations of strictly convex quadratic functions satisfy F1. Functions with non-convex sublevel sets can satisfy F1 (see Figure 4.1). More generally, strictly increasing transformations of C^1 positively homogeneous functions with a unique global argmin satisfy F1. Recall that a function p is positively homogeneous with degree $\alpha > 0$ and with respect to $x^\star$ if for all $x, y \in \mathbb{R}^n$, for all $\rho > 0$,

$$p(\rho(x - x^\star)) = \rho^\alpha p(x - x^\star) \ . \tag{4.14}$$

been made to replace Gaussian distributions by Cauchy distributions [119, 171, 199]. Yet, their implementations are typically not rotational invariant and steep performance gains are observed either in low dimensions or crucially based on the implicit exploitation of separability [92].

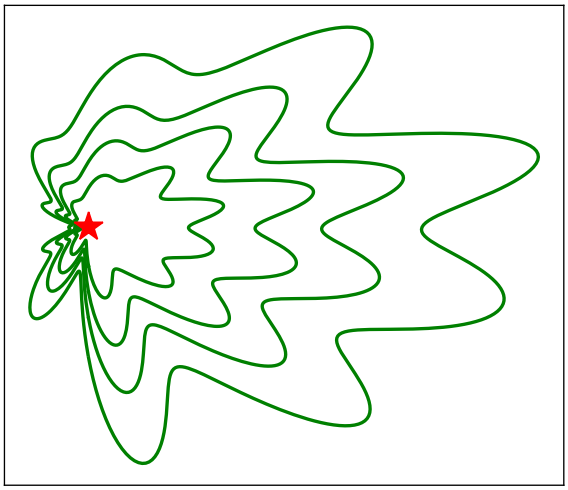


Figure 4.1: Level sets of scaling-invariant functions with respect to the red star $x^\star$. A randomly generated scaling-invariant function from a "smoothly" randomly perturbed sphere function.

4.2.5 Preliminary results

If f is scaling-invariant with respect to $x^\star$, the composite of the selection function α_f with the translation $(z,u) \mapsto (x^\star + z, u)$ is positively homogeneous with degree 1. If in addition f is a measurable function with Lebesgue negligible level sets, then [47, Proposition 5.2] gives the explicit expression of the probability density function of $\alpha_f(x^\star + z, U_1)$ where U_1 follows the distribution of $\mathcal{N}_{n\lambda}$. These results are formalized in the next lemma.

Lemma 7. *If f is a scaling-invariant function with respect to $x^\star$, then the function $(z,u) \mapsto \alpha_f(x^\star + z, u)$ is positively homogeneous with degree 1. In other words, for all $z \in \mathbb{R}^n$, $\sigma > 0$ and $u = \left(u^1, \ldots, u^\lambda\right) \in \mathbb{R}^{n\lambda}$*

$$\alpha_f\left(x^\star + \sigma z, \sigma u\right) = \sigma \alpha_f\left(x^\star + z, u\right).$$

If in addition f is a measurable function with Lebesgue negligible level sets and $U_1 = \left(U_1^1, \ldots, U_1^\lambda\right)$ is distributed according to $\mathcal{N}_{n\lambda}$, then for all $z \in \mathbb{R}^n$, the probability density function p_z^f of $\alpha_f(x^\star + z, U_1)$ exists and for all $u = (u^1, \ldots, u^\mu) \in \mathbb{R}^{n\mu}$,

$$p_z^f(u) = \frac{\lambda!}{(\lambda-\mu)!}(1 - Q_z^f(u^\mu))^{\lambda-\mu} \prod_{i=1}^{\mu-1} \mathbb{1}_{f(x^\star+z+u^i)<f(x^\star+z+u^{i+1})} \prod_{i=1}^{\mu} p_{\mathcal{N}_n}(u^i) \tag{4.15}$$

where $Q_z^f(w) = P\left(f\left(x^\star + z + \mathcal{N}_n\right) \leq f\left(x^\star + z + w\right)\right)$.

Proof. We have that $f(x^\star+z+u^{1:\lambda}) \leq \cdots \leq f(x^\star+z+u^{\lambda:\lambda})$ if and only if $f(x^\star+\sigma(z+u^{1:\lambda})) \leq \cdots \leq f(x^\star+\sigma(z+u^{\lambda:\lambda}))$. Therefore $\alpha_f\left(x^\star + \sigma z, \sigma u\right) = \sigma\left(u^{1:\lambda}, \ldots, u^{\mu:\lambda}\right) = \sigma\alpha_f\left(x^\star + z, u\right)$.

Equation (4.15) is given by [47, Proposition 5.2] whenever f has Lebesgue negligible level sets. □

On a linear function f, the selection function α_f defined in (4.3) is independent of the current state of the algorithm and is positively homogeneous with degree 1. This result is underlying previous results [19, 48]. We provide here a simple formalism and proof.

Lemma 8. *If f is an increasing transformation of a linear function, then for all $x \in \mathbb{R}^n$ the function $\alpha_f(x, \cdot)$ does not depend on x and is positively homogeneous with degree 1. In other words, for $x \in \mathbb{R}^n$, $\sigma > 0$ and $u = \left(u^1, \ldots, u^\lambda\right) \in \mathbb{R}^{n\lambda}$*

$$\alpha_f(x, \sigma u) = \sigma \alpha_f(0, u).$$

Proof. By linearity $f(x + \sigma u^{1:\lambda}) \leq \cdots \leq f(x + \sigma u^{\lambda:\lambda})$ if and only if $f(u^{1:\lambda}) \leq \cdots \leq f(u^{\lambda:\lambda})$. Therefore $\alpha_f(x, \sigma u) = \sigma\left(u^{1:\lambda}, \ldots, u^{\mu:\lambda}\right) = \sigma \alpha_f(0, u)$. □

Let $l^\star$ be the linear function defined for all $x \in \mathbb{R}^n$ as $l^\star(x) = x_1$ and $U_1 = \left(U_1^1, \ldots, U_1^\lambda\right)$ where $U_1^1, \ldots, U_1^\lambda$ are i.i.d. with law $\mathcal{N}_n$. Define the step-size change $\Gamma^\star_{\text{linear}}$ as

$$\Gamma^\star_{\text{linear}} = \Gamma\left(\alpha_{l^\star}(0, U_1)\right). \tag{4.16}$$

We prove in the next proposition that for all nontrivial linear functions, the step-size multiplicative factor of the algorithm (4.8) and (4.9) has at all iterations the distribution of $\Gamma^\star_{\text{linear}}$. This result derives from the rotation invariance of the function Γ (see Assumption A2) and of the probability density function $p_{\mathcal{N}_{n\mu}} : u \mapsto \frac{1}{(2\pi)^{n\mu/2}} \exp\left(-\|u\|^2/2\right)$. The details of the proof are in Appendix 1.2.

Proposition 17. *(Invariance of the step-size multiplicative factor on linear functions) Let f be an increasing transformation of a nontrivial linear function, i.e. satisfy F2. Assume that $\{U_{k+1}\,; k \in \mathbb{N}\}$ satisfies Assumption A5 and that Γ satisfies Assumption A2, $i.e.$ Γ is invariant under rotation. Then for all $z \in \mathbb{R}^n$ and all natural integer k, the step-size multiplicative factor $\Gamma\left(\alpha_f(z, U_{k+1})\right)$ has the law of the step-size change $\Gamma^\star_{\text{linear}}$ defined in (4.16).*

The proposition shows that on any (nontrivial) linear function the step-size change factor is independent of X_k, Z_k and even σ_k. We can now state the result which is at the origin of the methodology used in this paper, namely that on scaling-invariant functions, $\{Z_k = (X_k - x^\star)/\sigma_k\,; k \in \mathbb{N}\}$ is a homogeneous Markov chain. (We specify later on why the stability of this chain is key for the linear convergence of $\{(X_k, \sigma_k)\,; k \in \mathbb{N}\}$.) For this, we introduce the following function

$$F_w(z, v) = \frac{z + \sum_{i=1}^\mu w_i v_i}{\Gamma(v)} \text{ for all } (z, v) \in \mathbb{R}^n \times \mathbb{R}^{n\mu}, \tag{4.17}$$

which allows to write Z_{k+1} as a deterministic function of Z_k and U_{k+1}. The following proposition establishes conditions under which $\{Z_k; k \in \mathbb{N}\}$ is a homogeneous Markov chain that is defined with (4.17), independently of $\{(X_k, \sigma_k); k \in \mathbb{N}\}$. We refer to $\{Z_k; k \in \mathbb{N}\}$ as the σ-normalized chain. This is a particular case of [25, Proposition 4.1] where a more abstract algorithm framework is assumed.

Proposition 18. *Let f be a scaling invariant function with respect to $x^\star$ and $\{(X_k, \sigma_k); k \in \mathbb{N}\}$ be the sequences defined in (4.6) and (4.7). Then $\{Z_k = (X_k - x^\star)/\sigma_k; k \in \mathbb{N}\}$ is a homogeneous Markov chain and for all natural integer k, the following equation holds*

$$Z_{k+1} = F_w\left(Z_k, \alpha_f\left(x^\star + Z_k, U_{k+1}\right)\right), \tag{4.18}$$

where α_f is defined in (4.3), F_w is defined in (4.17) and $\{U_{k+1}; k \in \mathbb{N}\}$ is the sequence of random inputs used to sample the candidate solutions in (4.1) corresponding to the random input in (4.8) and (4.9).

Proof. The definition of the selection function α_f allows to write (4.6) and (4.7) as (4.8) and (4.9). We then divide (4.8) by (4.9), it follows:

$$\begin{aligned} Z_{k+1} = \frac{X_{k+1} - x^\star}{\sigma_{k+1}} &= \frac{X_k - x^\star + \sum_{i=1}^{\mu} w_i \left[\alpha_f(X_k, \sigma_k U_{k+1})\right]_i}{\sigma_k \, \Gamma\left(\frac{\alpha_f(X_k, \sigma_k U_{k+1})}{\sigma_k}\right)} \\ &= \frac{Z_k + \sum_{i=1}^{\mu} w_i \frac{\left[\alpha_f(X_k, \sigma_k U_{k+1})\right]_i}{\sigma_k}}{\Gamma\left(\frac{\alpha_f(X_k, \sigma_k U_{k+1})}{\sigma_k}\right)} \quad . \end{aligned}$$

By Lemma 7, $\frac{\alpha_f(X_k, \sigma_k U_{k+1})}{\sigma_k} = \frac{\alpha_f(x^\star + X_k - x^\star, \sigma_k U_{k+1})}{\sigma_k} = \alpha_f(x^\star + \frac{X_k - x^\star}{\sigma_k}, U_{k+1})$. Then $Z_{k+1} = F_w\left(Z_k, \alpha_f\left(x^\star + Z_k, U_{k+1}\right)\right)$ and $\{Z_k; k \in \mathbb{N}\}$ is a homogeneous Markov chain. □

Three invariances are key to obtain that $\{Z_k = (X_k - x^\star)/\sigma_k; k \in \mathbb{N}\}$ is a homogeneous Markov chain: invariance to strictly increasing transformations (stemming from the comparison-based property of ES), translation invariance, and scale invariance [25, Proposition 4.1]. The last two invariances are satisfied with the update we assume for mean and step-size.

4.3 Main results

We present our main results that express the global linear convergence of the algorithm presented in Section 4.2. Linear convergence can be visualized by looking at the distance to the optimum: after an adaptation phase, we observe that the log distance to the optimum diverges to minus infinity with a graph that resembles a straight line with random perturbations. The step-size converges to zero at the same linear

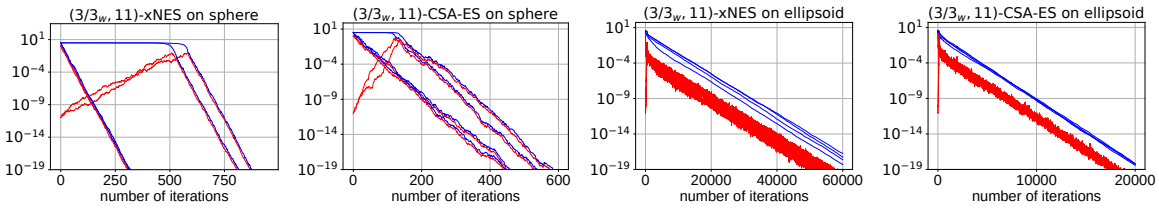


Figure 4.2: Four independent runs of $(\mu/\mu_w, \lambda)$-xNES and $(\mu/\mu_w, \lambda)$-CSA1-ES (without cumulation) as presented in Section 4.2.1 on the functions $x \mapsto \|x\|^2$ (first two figures) and $x \mapsto \sum_{i=1}^{n} 10^3 \frac{i-1}{n-1} x_i^2$ (last two figures). Illustration of $\|X_k\|$ in blue and σ_k in red where k is the number of iterations, $\mu = 3$, $\lambda = 11$ and $w_i = 1/\mu$. Initializations: σ_0 equals to 10^{-11} in two runs and 1 in the two other runs, X_0 is the all-ones vector in dimension 10.

rate (see Figure 4.2). We call this constant the convergence rate of the algorithm. Formally, in case of convergence, there exists $\mathrm{r} > 0$ such that

$$\lim_{k\to\infty} \frac{1}{k} \log \frac{\|X_k - x^\star\|}{\|X_0 - x^\star\|} = \lim_{k\to\infty} \frac{1}{k} \log \frac{\sigma_k}{\sigma_0} = -\mathrm{r} \tag{4.19}$$

where $x^\star$ is the optimum of the function. We prove (4.19) for f satisfying F1 while our approach does not allow to prove the sign of the rate r. When (4.19) holds and r is strictly negative, then the algorithm diverges linearly. We prove this linear divergence for functions satisfying F2. In the sequel we talk about linear *behavior* when (4.19) holds but the sign of the rate r is not specified. We can in a straightforward manner simulate the convergence rate (and obtain its sign, see for example Figure 4.2) as our theory shows consistency of the estimators, as is discussed later.

4.3.1 Linear behavior

Our condition for the linear behavior is that the expected logarithm of the step-size change function Γ on a nontrivial linear function is positive. More precisely, let us denote the expected change of the logarithm of the step-size for any state $z \in \mathbb{R}^n$ of the σ-normalized chain as

$$\mathcal{R}_f(z) = \mathbb{E}_{U_1 \sim \mathcal{N}_{n\lambda}} \left[\log \left(\Gamma \left(\alpha_f(x^\star + z, U_1) \right) \right) \right] \quad . \tag{4.20}$$

By Proposition 17, when f satisfies F2, the expected change of the logarithm of the step-size is constant and for all z,

$$\mathcal{R}_f(z) = \mathcal{R}_f(-x^\star) = \mathbb{E}\left[\log \left(\Gamma^\star_{\text{linear}} \right) \right]$$

where $\Gamma^\star_{\text{linear}}$ is defined in (4.16). Our main result states that if the expected logarithm of the step-size increases on nontrivial linear functions, in other words if $\mathbb{E}\left[\log \left(\Gamma^\star_{\text{linear}} \right) \right] > 0$, then almost sure linear behavior holds on functions satisfying F1 or F2. If f satisfies F2, then almost sure linear divergence holds with a divergence rate of $\mathbb{E}\left[\log \left(\Gamma^\star_{\text{linear}} \right) \right]$.

Theorem 5. *Let f be a scaling-invariant function with respect to $x^\star$. Assume that f satisfies F1 (in which case $x^\star$ is the global optimum) or F2. Let $\{(X_k, \sigma_k)\,; k \in \mathbb{N}\}$ be the sequence defined in (4.6) and (4.7) such that Assumptions A1–A5 are satisfied. Let $\{Z_k = (X_k - x^\star)/\sigma_k\,; k \in \mathbb{N}\}$ be the homogeneous Markov chain defined in Proposition 18. Define $\mathcal{R}_f$ as in (4.20). If the expected logarithm of the step-size increases on nontrivial linear functions, $i.e.$ if $\mathbb{E}\left[\log\left(\Gamma^\star_{\text{linear}}\right)\right] > 0$ where $\Gamma^\star_{\text{linear}}$ is defined in (4.16), then $\{Z_k\,; k \in \mathbb{N}\}$ admits an invariant probability measure π such that $\mathcal{R}_f$ is π-integrable. And for all $(X_0, \sigma_0) \in (\mathbb{R}^n \setminus \{x^\star\}) \times (0, \infty)$, linear behavior of X_k and σ_k as in (4.19) holds almost surely with*

$$\lim_{k\to\infty} \frac{1}{k} \log \frac{\|X_k - x^\star\|}{\|X_0 - x^\star\|} = \lim_{k\to\infty} \frac{1}{k} \log \frac{\sigma_k}{\sigma_0} = \mathbb{E}_\pi(\mathcal{R}_f) \quad . \tag{4.21}$$

In addition, for all initial conditions $(X_0, \sigma_0) = (x, \sigma) \in \mathbb{R}^n \times (0, \infty)$, we have linear behavior of the expected log-progress, with

$$\lim_{k\to\infty} \mathbb{E}_{\frac{x-x^\star}{\sigma}} \left[\log \frac{\|X_{k+1} - x^\star\|}{\|X_k - x^\star\|}\right] = \lim_{k\to\infty} \mathbb{E}_{\frac{x-x^\star}{\sigma}} \left[\log \frac{\sigma_{k+1}}{\sigma_k}\right] = \mathbb{E}_\pi(\mathcal{R}_f) \quad . \tag{4.22}$$

If f satisfies F2, then $\mathcal{R}_f$ is constant equal to $\mathbb{E}_\pi(\mathcal{R}_f) = \mathbb{E}\left[\log\left(\Gamma^\star_{\text{linear}}\right)\right] > 0$, and then both X_k and σ_k diverge to infinity with a divergence rate of $\mathbb{E}\left[\log\left(\Gamma^\star_{\text{linear}}\right)\right]$.

If $\mathbb{E}_\pi(\mathcal{R}_f) < 0$, then X_k converges (linearly) to the global optimum $x^\star$ with a convergence rate of $-\mathbb{E}_\pi(\mathcal{R}_f)$ and the step-size converges to zero.

The result that both the step-size and log distance converge (resp. diverge) to the optimum (resp. to ∞) at the same rate is noteworthy and directly follows from our theory. In addition, we provide the exact expression of the rate. Yet it is expressed using the stationary distribution of the Markov chain $\{Z_k\,; k \in \mathbb{N}\}$ for which we know little information. Indeed, in contrast to the analysis of many Markov Chain Monte Carlo (MCMC) algorithms like the Metropolis algorithm [98, 138] where the stationary distribution is known *a priori* (it is the distribution we wish to simulate), here we do not know π and have to show its existence. This explains the difficulties in actually proving the sign of the convergence or divergence rate.

From a practical perspective, while we never know the optimum of a function on a real problem, (4.19) suggests that we can track the evolution of the step-size to define a termination criterion based on the tolerance of the x-values.

The almost sure linear behavior result in (4.21) derives from applying a Law of Large Numbers (LLN) to $\{Z_k\,; k \in \mathbb{N}\}$ and a generalized law of large numbers to the chain $\{(Z_k, U_{k+2})\,; k \in \mathbb{N}\}$. Our proof techniques are mostly about showing that $\{Z_k\,; k \in \mathbb{N}\}$ satisfies the right properties for a LLN to hold. Yet we actually prove stronger properties than what is needed for a LLN and imply from there a central limit theorem related to the expected logarithm of the step-size change.

4.3.2 Central limit theorem

The rate of convergence (or divergence) of a step-size adaptive $(\mu/\mu_w, \lambda)$-ES given in (4.21) is expressed as $|\mathbb{E}_\pi(\mathcal{R}_f)|$ where π is the invariant probability measure of the σ-normalized Markov chain and $\mathcal{R}_f$ is defined in (4.20). Yet we do not have an explicit expression for π and thus of $\mathbb{E}_\pi(\mathcal{R}_f)$. However we can approximate $\mathbb{E}_\pi(\mathcal{R}_f)$ with Monte Carlo simulations. We present a central limit theorem for the approximation of $\mathbb{E}_\pi(\mathcal{R}_f)$ as $\frac{1}{t}\sum_{k=0}^{t-1}\mathcal{R}_f(Z_k)$ where $\{Z_k; k \in \mathbb{N}\}$ is the homogeneous Markov chain defined in Proposition 18.

Theorem 6. *(Central limit theorem for the expected* $\log$ *step-size) Let* f *be a scaling-invariant function with respect to* $x^\star$ *that satisfies F1 or F2. Let* $\{(X_k, \sigma_k)\,; k \in \mathbb{N}\}$ *be the sequence defined in* (4.6) *and* (4.7) *such that Assumptions A1–A5 are satisfied. If the expected logarithm of the step-size increases on nontrivial linear functions, i.e. if* $\mathbb{E}\left[\log\left(\Gamma^\star_{\text{linear}}\right)\right] > 0$ *where* $\Gamma^\star_{\text{linear}}$ *is defined in* (4.16)*, then the Markov chain* $\{Z_k = (X_k - x^\star)/\sigma_k\,; k \in \mathbb{N}\}$ *admits an invariant probability measure* π*. Define* $\mathcal{R}_f$ *as in* (4.20) *and for all positive integer* t*, define* $S_t(\mathcal{R}_f) = \sum_{k=0}^{t-1}\mathcal{R}_f(Z_k)$*. Then the constant* γ^2 *defined as*

$$\mathbb{E}_\pi\left[\left(\mathcal{R}_f(Z_0) - \mathbb{E}_\pi(\mathcal{R}_f)\right)^2\right] + 2\sum_{k=1}^{\infty}\mathbb{E}_\pi\left[\left(\mathcal{R}_f(Z_0) - \mathbb{E}_\pi(\mathcal{R}_f)\right)\left(\mathcal{R}_f(Z_k) - \mathbb{E}_\pi(\mathcal{R}_f)\right)\right]$$

is well defined, non-negative, finite and $\lim_{t\to\infty}\frac{1}{t}\mathbb{E}_\pi\left[\left(S_t(\mathcal{R}_f) - t\,\mathbb{E}_\pi(\mathcal{R}_f)\right)^2\right] = \gamma^2$.

If $\gamma^2 > 0$*, then the central limit theorem holds in the sense that for any initial condition* z_0*,* $\sqrt{\frac{t}{\gamma^2}}\left(\frac{1}{t}S_t(\mathcal{R}_f) - \mathbb{E}_\pi(\mathcal{R}_f)\right)$ *converges in distribution to* $\mathcal{N}(0,1)$*. If* $\gamma^2 = 0$*, then* $\lim_{t\to\infty}\frac{S_t(\mathcal{R}_f) - t\mathbb{E}_\pi(\mathcal{R}_f)}{\sqrt{t}} = 0$ ***a.s.***

4.3.3 Sufficient conditions for the linear behavior of $(\mu/\mu_w, \lambda)$-CSA1-ES and $(\mu/\mu_w, \lambda)$-xNES

Theorems 5 and 6 hold for an abstract step-size update function Γ that satisfies Assumptions A1–A4. For the step-size update functions of the $(\mu/\mu_w, \lambda)$-CSA1-ES and the $(\mu/\mu_w, \lambda)$-xNES defined in (4.11) and (4.12), sufficient and necessary conditions to obtain a step-size increase on linear functions are presented in the next proposition. They are expressed using the weights and the μ best order statistics $\mathcal{N}^{1:\lambda}, \ldots, \mathcal{N}^{\mu:\lambda}$ of a sample of λ standard normal distributions $\mathcal{N}^1, \ldots, \mathcal{N}^\lambda$ defined such as $\mathcal{N}^{1:\lambda} \leq \mathcal{N}^{2:\lambda} \leq \cdots \leq \mathcal{N}^{\lambda:\lambda}$.

Proposition 19 (**Necessary and sufficient condition for step-size increase on nontrivial linear functions**)**.** *For the* $(\mu/\mu_w, \lambda)$*-CSA-ES without cumulation,*

$\mathbb{E}\left[\log\left((\Gamma_{\mathrm{CSA1}})^{\star}_{linear}\right)\right] = \frac{1}{2d_\sigma n}\left(\mathbb{E}\left[\left(\sum_{i=1}^{\mu}\frac{w_i}{\|w\|}\mathcal{N}^{i:\lambda}\right)^2\right]-1\right)$. *Therefore, the expected logarithm of the step-size increases on nontrivial linear functions if and only if*

$$\mathbb{E}\left[\left(\sum_{i=1}^{\mu}\frac{w_i}{\|w\|}\mathcal{N}^{i:\lambda}\right)^2\right] > 1 \quad . \tag{4.23}$$

For the $(\mu/\mu_w,\lambda)$-xNES without covariance matrix adaptation, if $w_i \geq 0$ for all $i = 1,\dots,\mu$, $\mathbb{E}\left[\log\left((\Gamma_{\mathrm{xNES}})^{\star}_{linear}\right)\right] = \frac{1}{2d_\sigma n}\left(\sum_{i=1}^{\mu}\frac{w_i}{\sum_{j=1}^{\mu}w_j}\mathbb{E}\left[\left(\mathcal{N}^{i:\lambda}\right)^2\right]-1\right)$. Therefore the expected logarithm of the step-size increases on nontrivial linear functions if and only if

$$\sum_{i=1}^{\mu}\frac{w_i}{\sum_{j=1}^{\mu}w_j}\mathbb{E}\left[\left(\mathcal{N}^{i:\lambda}\right)^2\right] > 1 \quad . \tag{4.24}$$

In addition, this latter equation is satisfied if λ, μ and w are set such that $\lambda \geq 3$, $\mu < \frac{\lambda}{2}$ and $w_1 \geq w_2 \geq \cdots \geq w_\mu \geq 0$.

The positivity of $\mathbb{E}\left[\log\left(\Gamma^{\star}_{\mathrm{linear}}\right)\right]$ is the main assumption for our main results. In this context, Proposition 19 gives more practical and concrete ways to obtain the conclusion of Theorems 5 and 6 for the $(\mu/\mu_w,\lambda)$-CSA1-ES and $(\mu/\mu_w,\lambda)$-xNES.

In the case where $\mu = 1$, (4.23) and (4.24) are equivalent and yield the equation $\mathbb{E}\left[\left(\mathcal{N}^{1:\lambda}\right)^2\right] > 1$. The latter is satisfied if $\lambda \geq 3$ and $\mu = 1$, which is the linear divergence condition on linear functions of the $(1,\lambda)$-CSA1-ES in [48].

Conditions similar to (4.23) had already been derived for the so-called mutative self-adaptation of the step-size [84].

4.4 Introduction of the methodology and reminders on Markov chains

We sketch in this section the main steps of our methodology and introduce definitions and tools stemming from Markov chain theory that are needed in the rest of the paper.

If f is scaling-invariant with respect to $x^\star$, $\{Z_k = (X_k - x^\star)/\sigma_k\,; k \in \mathbb{N}\}$ is a homogeneous Markov chain where $\{(X_k,\sigma_k)\,; k \in \mathbb{N}\}$ is the sequence of states of the step-size adaptive $(\mu/\mu_w,\lambda)$-ES defined in (4.6) and (4.9) (see Proposition 18). Since $X_k - x^\star = \sigma_k Z_k$, it follows that

$$\begin{aligned}\log\frac{\|X_{k+1}-x^\star\|}{\|X_k-x^\star\|} &= \log\frac{\|Z_{k+1}\|}{\|Z_k\|} + \log\frac{\sigma_{k+1}}{\sigma_k} \\ &= \log\frac{\|Z_{k+1}\|}{\|Z_k\|} + \log\left(\Gamma\left(\alpha_f\left(x^\star + Z_k, U_{k+1}\right)\right)\right)\end{aligned} \tag{4.25}$$

where Γ and α_f are defined in (4.7) and in (4.3). We deduce from the previous equation the following one

$$\frac{1}{k}\log\frac{\|X_k - x^\star\|}{\|X_0 - x^\star\|} = \frac{1}{k}\sum_{t=0}^{k-1}\log\frac{\|X_{t+1} - x^\star\|}{\|X_t - x^\star\|} \tag{4.26}$$

$$= \frac{1}{k}\sum_{t=0}^{k-1}\log\frac{\|Z_{t+1}\|}{\|Z_t\|} + \frac{1}{k}\sum_{t=0}^{k-1}\log(\Gamma(\alpha_f(x^\star + Z_t, U_{t+1}))) \ . \tag{4.27}$$

This latter equation suggests that if we can apply a law of large numbers to $\{Z_k\,; k \in \mathbb{N}\}$ and $\{(Z_k, U_{k+1})\,; k \in \mathbb{N}\}$, the right-hand side converges to $\int \mathbb{E}_{U_1\sim\mathcal{N}_{n\lambda}}\left[\log\left(\Gamma\left(\alpha_f(x^\star + z, U_1)\right)\right)\right]\pi(dz) = \mathbb{E}_\pi(\mathcal{R}_f)$ where $\mathcal{R}_f$ is defined in (4.20) and π is the invariant measure of $\{Z_k\,; k \in \mathbb{N}\}$. From there, we obtain the almost sure convergence of $\frac{1}{k}\log\frac{\|X_k - x^\star\|}{\|X_0 - x^\star\|}$ towards $\mathbb{E}_\pi(\mathcal{R}_f)$ expressed in (4.21) translating the asymptotic linear behavior of the algorithm.

This is the main idea behind the asymptotic linear behavior proof we provide in the paper. This idea was introduced in [33] in the context of a self-adaptation evolution strategy on the sphere function, exploited in [19] and generalized to a wider class of algorithms and functions in [25]. We therefore see that we need to investigate under which conditions $\{Z_k\,; k \in \mathbb{N}\}$ and $\{(Z_k, U_{k+1})\,; k \in \mathbb{N}\}$ satisfy a LLN.

Similarly the proof idea for (4.22) goes as follows. Let us first define for a Markov chain $\{Z_k\,; k \in \mathbb{N}\}$ on a measure space $(\mathcal{Z}, \mathcal{B}(\mathcal{Z}), P)$ where $\mathcal{Z}$ in an open subset of $\mathbb{R}^n$, for all $k \in \mathbb{N}$ its k-step transition kernel as

$$P^k(z, A) = P\left(Z_k \in A | Z_0 = z\right)$$

for $z \in \mathcal{Z}$, $A \in \mathcal{B}(\mathcal{Z})$. We also denote $P(z, A)$ and $P_z(A)$ as $P^1(z, A)$.

If we take the expectation under $Z_0 = z$ in (4.9), then

$$\mathbb{E}_z\left[\log\frac{\sigma_{k+1}}{\sigma_k}\right] = \mathbb{E}_z\left[\log\left(\Gamma\left(\alpha_f\left(x^\star + Z_k, U_{k+1}\right)\right)\right)\right] = \int P^k(z, \mathrm{d}y)\mathcal{R}_f(y).$$

With (4.25) we have that

$$\mathbb{E}_z\left[\log\frac{\|X_{k+1} - x^\star\|}{\|X_k - x^\star\|}\right] = \mathbb{E}_z\left[\log\frac{\|Z_{k+1}\|}{\|Z_k\|}\right] + \mathbb{E}_z\left[\log\frac{\sigma_{k+1}}{\sigma_k}\right] \tag{4.28}$$

$$= \int P^{k+1}(z, \mathrm{d}y)\log(\|y\|) - \int P^k(z, \mathrm{d}y)\log(\|y\|)$$

$$+ \int P^k(z, \mathrm{d}y)\mathcal{R}_f(y). \tag{4.29}$$

If $P^k(z, .)$ converges to π assuming all the limits can be taken, then the right-hand side converges to $\mathbb{E}_\pi(\mathcal{R}_f)$ as the two first integrals cancel each other such that

$$\lim_{k\to\infty}\mathbb{E}_{\frac{x-x^\star}{\sigma}}\left[\log\frac{\|X_{k+1} - x^\star\|}{\|X_k - x^\star\|}\right] = \lim_{k\to\infty}\mathbb{E}_{\frac{x-x^\star}{\sigma}}\left[\log\frac{\sigma_{k+1}}{\sigma_k}\right] = \mathbb{E}_\pi(\mathcal{R}_f),$$

$i.e.$ (4.22) is satisfied. To prove that $P^k(z,.)$ converges to π we prove the h-ergodicity of $\{Z_k\,; k \in \mathbb{N}\}$, a notion formally defined later on.

We remind in the rest of this part different notions on Markov chains that we investigate later on to prove in particular that $\{Z_k\,; k \in \mathbb{N}\}$ satisfies a LLN, a central limit theorem and that for some $z \in \mathcal{Z}$, $P^k(z,\cdot)$ converges to a stationary distribution. Following the terminology of [140], we refer in an informal way to those properties as stability properties.

4.4.1 Stability notions and practical drift conditions

The first stability notion to be verified is the so-called φ-irreducibility. If there exists a nontrivial measure φ on $(\mathcal{Z},\mathcal{B}(\mathcal{Z}))$ such that for all $A \in \mathcal{B}(\mathcal{Z})$, $\varphi(A) > 0$ implies $\sum_{k=1}^{\infty} P^k(z,A) > 0$ for all $z \in \mathcal{Z}$, then the chain is called φ-irreducible. A φ-irreducible Markov chain is Harris recurrent if for all $A \in \mathcal{B}(\mathcal{Z})$ with $\varphi(A) > 0$ and for all $z \in \mathcal{Z}$, $P_z(\eta_A = \infty) = 1$, where $\eta_A = \sum_{k=1}^{\infty} \mathbb{1}_{Z_k \in A}$ is the *occupation time* of A.

A σ-finite measure π on $(\mathcal{Z},\mathcal{B}(\mathcal{Z}))$ is an invariant measure for $\{Z_k; k \in \mathbb{N}\}$ if for all $A \in \mathcal{B}(\mathcal{Z})$, $\pi(A) = \int_{\mathcal{Z}} \pi(dz)P(z,A)$. A Harris recurrent chain admits a unique (up to constant multiples) invariant measure π (see [140, Theorem 10.0.1]). A φ-irreducible Markov chain admitting an invariant probability measure π is said positive. A positive Harris-recurrent chain satisfies a LLN as reminded below.

Theorem 7. *[140, Theorem 17.0.1] If $\{Z_k; k \in \mathbb{N}\}$ is a positive and Harris recurrent chain with invariant probability measure π, then the LLN holds for any π-integrable function g, $i.e.$ for any g with $\mathbb{E}_\pi(|g|) < \infty$, $\lim_{k\to\infty} \frac{1}{k} \sum_{t=0}^{k-1} g(Z_t) = \mathbb{E}_\pi(g)$.*

We prove positivity and Harris-recurrence using Foster-Lyapunov drift conditions. Before introducing those conditions we need the notion of aperiodicity. Assume that d is a positive integer and $\{Z_k\,; k \in \mathbb{N}\}$ is a φ-irreducible Markov chain defined on $(\mathcal{Z},\mathcal{B}(\mathcal{Z}))$. Let $(D_i)_{i=1,\ldots,d} \in \mathcal{B}(\mathcal{Z})^d$ be a sequence of disjoint sets. Then $(D_i)_{i=1,\ldots,d}$ is called a d-cycle if

(i) $P(z, D_{i+1}) = 1$ for all $z \in D_i$ and $i = 0,\ldots,d-1$ (mod d),

(ii) $\Lambda\left(\left(\bigcup_{i=1}^{d} D_i\right)^c\right) = 0$ for all irreducibility measure Λ of $\{Z_k\,; k \in \mathbb{N}\}$.

If $\{Z_k; k \in \mathbb{N}\}$ is φ-irreducible, there exists a d-cycle where d is a positive integer [140, Theorem 5.4.4]. The largest d for which there exists a d-cycle is called the period of $\{Z_k\,; k \in \mathbb{N}\}$. We then say that a φ-irreducible Markov chain $\{Z_k\,; k \in \mathbb{N}\}$ on $(\mathcal{Z},\mathcal{B}(\mathcal{Z}))$ is aperiodic if it has a period of 1.

A set $C \in \mathcal{B}(\mathcal{Z})$ is called *small* if there exists a positive integer k and a nontrivial measure ν_k on $\mathcal{B}(\mathcal{Z})$ such that $P^k(z,A) \geq \nu_k(A)$ for all $z \in C$, $A \in \mathcal{B}(\mathcal{Z})$. We then say that C is a ν_k-small set [140].

Given an extended-valued, non-negative and measurable function $V : \mathcal{Z} \to \mathbb{R}_+ \cup \{\infty\}$ (called potential function), the drift operator is defined for all $z \in \mathcal{Z}$ as

$$\Delta V(z) = \mathbb{E}[V(Z_1)|Z_0 = z] - V(z) = \int_{\mathcal{Z}} V(y)P(z, \mathrm{d}y) - V(z) \ .$$

A φ-irreducible, aperiodic Markov chain $\{Z_k\,; k \in \mathbb{N}\}$ defined on $(\mathcal{Z}, \mathcal{B}(\mathcal{Z}))$ satisfies a geometric drift condition if there exist $0 < \gamma < 1$, $b \in \mathbb{R}$, a small set C and a potential function V greater than 1, finite at some $z_0 \in \mathcal{Z}$ such that for all $z \in \mathcal{Z}$:

$$\Delta V(z) \leq (\gamma - 1)V(z) + b\mathbb{1}_C(z) \ ,$$

or equivalently if $\mathbb{E}[V(Z_1)|Z_0 = z] \leq \gamma V(z) + b\mathbb{1}_C(z)$. The function V is called a geometric drift function and if $\{y \in \mathcal{Z}\,; V(y) < \infty\} = \mathcal{Z}$, we say that $\{Z_k\,; k \in \mathbb{N}\}$ is V-geometrically ergodic.

If a φ-irreducible and aperiodic Markov chain is V-geometrically ergodic, then it is positive and Harris recurrent [140, Theorem 13.0.1 and Theorem 9.1.8]. We prove a geometric drift condition in Section 4.5.3, this in turn implies positivity and Harris-recurrence property.

From a geometric drift condition follows a stronger result than a LLN, namely a central limit theorem.

Theorem 8. *[140, Theorem 17.0.1 and Theorem 16.0.1] Let $\{Z_k\,; k \in \mathbb{N}\}$ be a φ-irreducible aperiodic Markov chain on $(\mathcal{Z}, \mathcal{B}(\mathcal{Z}))$ that is V-geometrically ergodic, with invariant probability measure π. For any function g on $\mathcal{Z}$ that satisfies $g^2 \leq V$, the central limit theorem holds for $\{Z_k\,; k \in \mathbb{N}\}$ in the following sense. Define $\bar{g} = g - \mathbb{E}_\pi(g)$ and for all positive integer t, define $S_t(\bar{g}) = \sum_{k=0}^{t-1} \bar{g}(Z_k)$. Then the constant $\gamma^2 = \mathbb{E}_\pi[(\bar{g}(Z_0))^2] + 2\sum_{k=1}^{\infty} \mathbb{E}_\pi[\bar{g}(Z_0)\bar{g}(Z_k)]$ is well defined, non-negative, finite and $\lim_{t\to\infty} \frac{1}{t}\mathbb{E}_\pi[(S_t(\bar{g}))^2] = \gamma^2$. Moreover if $\gamma^2 > 0$ then $\frac{1}{\sqrt{t\gamma^2}} S_t(\bar{g})$ converges in distribution to $\mathcal{N}(0, 1)$ when t goes to ∞, else if $\gamma^2 = 0$ then $\frac{1}{\sqrt{t}} S_t(\bar{g}) = 0$ $a.s.$*

For a measurable function $h \geq 1$ on $\mathcal{Z}$, [140, Theorem 14.0.1] states that a φ-irreducible aperiodic Markov chain $\{Z_k\,; k \in \mathbb{N}\}$ defined on $(\mathcal{Z}, \mathcal{B}(\mathcal{Z}))$ is positive Harris recurrent with invariant probability measure π such that h is π-integrable if and only if there exist $b \in \mathbb{R}$, a small set C and an extended-valued non-negative function $V \neq \infty$ such that for all $z \in \mathcal{Z}$:

$$\Delta V(z) \leq -h(z) + b\mathbb{1}_C(z). \tag{4.30}$$

Recall that for a measurable function $h \geq 1$, we say that a general Markov chain $\{Z_k\,; k \in \mathbb{N}\}$ is h-ergodic if there exists a probability measure π such that $\lim_{k\to\infty} \|P^k(z, \cdot) - \pi\|_h = 0$ for any initial condition z. The probability measure π is then called the invariant probability measure of $\{Z_k\,; k \in \mathbb{N}\}$. If $h = 1$, we say that $\{Z_k\,; k \in \mathbb{N}\}$ is ergodic.

With [140, Theorem 14.0.1], a φ-irreducible aperiodic Markov chain on $\mathcal{Z}$ that satisfies (4.30) is h-ergodic if in addition $\{y \in \mathcal{Z}\,; V(y) < \infty\} = \mathcal{Z}$.

Prior to establishing a drift condition, we need to identify small sets. Using the notion of T-chain defined below, compact sets are small sets as a direct consequence of [140, Theorem 5.5.7 and Theorem 6.2.5] stating that for a φ-irreducible aperiodic T-chain, every compact set is a small set.

The T-chain property calls for the notion of kernel: a kernel K is a function on $(\mathcal{Z}, \mathcal{B}(\mathcal{Z}))$ such that for all $A \in \mathcal{B}(\mathcal{Z})$, $K(., A)$ is a measurable function and for all $z \in \mathcal{Z}$, $K(z, .)$ is a signed measure. A non-negative kernel K satisfying $K(z, \mathcal{Z}) \leq 1$ for all $z \in \mathcal{Z}$ is called substochastic. A substochastic kernel K satisfying $K(z, \mathcal{Z}) = 1$ for all $z \in \mathcal{Z}$ is a transition probability kernel. Let b be a probability distribution on $\mathbb{N}$ and denote by K_b the probability transition kernel defined as

$$K_b(z, A) = \sum_{k=0}^{\infty} b(k) P^k(z, A) \text{ for all } z \in \mathcal{Z},\ A \in \mathcal{B}(\mathcal{Z}).$$

If T is a substochastic transition kernel such that $T(., A)$ is lower semi-continuous for all $A \in \mathcal{B}(\mathcal{Z})$ and $K_b(z, A) \geq T(z, A)$ for all $z \in \mathcal{Z},\ A \in \mathcal{B}(\mathcal{Z})$, then T is called a continuous component of K_b. If a Markov chain $\{Z_k\,; k \in \mathbb{N}\}$ admits a probability distribution b on $\mathbb{N}$ such that K_b has a continuous component T that satisfies $T(z, \mathcal{Z}) > 0$ for all $z \in \mathcal{Z}$, then $\{Z_k\,; k \in \mathbb{N}\}$ is called a T-chain.

4.4.2 Generalized law of large numbers

To apply a LLN for the convergence of the term $\frac{1}{k}\sum_{t=0}^{k-1} \log(\Gamma(\alpha_f(x^\star + Z_t, U_{t+1})))$ in (4.27), we proceed in two steps. First we prove that if $\{Z_k\,; k \in \mathbb{N}\}$ is defined as $Z_{k+1} = G(Z_k, U_{k+1})$ where $G : \mathcal{Z} \times \mathbb{R}^m \to \mathcal{Z}$ is a measurable function and $\{U_{k+1}\,; k \in \mathbb{N}\}$ is a sequence of i.i.d. random vectors, then the ergodic properties of $\{Z_k\,; k \in \mathbb{N}\}$ are transferred to $\{W_k = (Z_k, U_{k+2})\,; k \in \mathbb{N}\}$. Afterwards we apply a generalized LLN introduced in [118] and recalled in the following theorem.

Theorem 9. *[118, Theorem 1] Assume that $\{Z_k\,; k \in \mathbb{N}\}$ is a homogeneous Markov chain on an abstract measurable space $(\mathbf{E}, \mathcal{E})$ that is ergodic with invariant probability measure π. For all measurable function $g : \mathbf{E}^\infty \to \mathbb{R}$ such that for all $s \in \mathbb{N}$, $\mathbb{E}_\pi(|g(Z_s, Z_{s+1}, \dots)|) < \infty$ and for any initial distribution Λ, the generalized LLN holds as follows*

$$\lim_{k\to\infty} \frac{1}{k} \sum_{t=0}^{k-1} g\left(Z_t, Z_{t+1}, \dots\right) = \mathbb{E}_\pi(g(Z_s, Z_{s+1}, \dots))\ \ P_\Lambda\ \ a.s.$$

where P_Λ is the distribution of the process $\{Z_k\,; k \in \mathbb{N}\}$ on $(\mathbf{E}^\infty, \mathcal{E}^\infty)$.

Theorem 9 generalizes [179, Theorems 3.5.7 and 3.5.8]. Indeed in [179, Theorems 3.5.7 and 3.5.8], the generalized LLN holds only if the initial state is distributed under the invariant measure, whereas in [118, Theorem 1], the initial distribution considered can be anything.

If we have the generalized LLN for a chain $\{(Z_k, U_{k+2})\,; k \in \mathbb{N}\}$ on $\mathbb{R}^n \times \mathbb{R}^m$, then a LLN for the chain $\{(Z_k, U_{k+1})\,; k \in \mathbb{N}\}$ is directly implied. We formalize this statement in the next corollary.

Corollary 4. *Assume that $\{W_k = (Z_k, U_{k+2})\,; k \in \mathbb{N}\}$ is a homogeneous Markov chain on $\mathbb{R}^n \times \mathbb{R}^m$ that is ergodic with invariant probability measure π. Then the LLN holds for $\{(Z_k, U_{k+1})\,; k \in \mathbb{N}\}$ in the following sense. Define the function $T : (\mathbb{R}^n \times \mathbb{R}^m)^2 \to \mathbb{R}^n \times \mathbb{R}^m$ as $T((z_1, u_3), (z_2, u_4)) = (z_2, u_3)$. If $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is such that for all $s \in \mathbb{N}$, $\mathbb{E}_\pi(|g \circ T|\,(W_s, W_{s+1})) < \infty$, then $\lim_{k\to\infty} \frac{1}{k} \sum_{t=0}^{k-1} g(Z_t, U_{t+1}) = \mathbb{E}_\pi\left[(g \circ T)\,(W_s, W_{s+1})\right]$.*

Proof. We have $\lim_{k\to\infty} \frac{1}{k} \sum_{t=0}^{k-1} (g \circ T)\,(W_t, W_{t+1}) = \mathbb{E}_\pi\left[(g \circ T)\,(W_s, W_{s+1})\right]$ thanks to Theorem 9. For $t \in \mathbb{N}$, $(g \circ T)\,(W_t, W_{t+1}) = g\,(Z_{t+1}, U_{t+2})$. Therefore

$$\mathbb{E}_\pi\left[(g \circ T)\,(W_s, W_{s+1})\right] = \lim_{k\to\infty} \frac{1}{k} \sum_{t=0}^{k-1} g\,(Z_{t+1}, U_{t+2}) = \lim_{k\to\infty} \frac{1}{k} \sum_{t=0}^{k-1} g(Z_t, U_{t+1}) \quad .$$

□

We formulate now that for a Markov chain following a non-linear state space model of the form $Z_{k+1} = G(Z_k, U_{k+1})$ with G measurable and $\{U_{k+1}\,; k \in \mathbb{N}\}$ i.i.d., then φ-irreducibility, aperiodicity and V-geometric ergodicity of Z_k are transferred to $\{W_k = (Z_k, U_{k+2})\,; k \in \mathbb{N}\}$. We provide a proof of this result in Appendix 1.3 for the sake of completeness.

Proposition 20. *Let $\{Z_k\,; k \in \mathbb{N}\}$ be a Markov chain on $(\mathcal{Z}, \mathcal{B}(\mathcal{Z}))$ defined as $Z_{k+1} = G(Z_k, U_{k+1})$ where $G : \mathcal{Z} \times \mathbb{R}^m \to \mathcal{Z}$ is a measurable function and $\{U_{k+1}\,; k \in \mathbb{N}\}$ is a sequence of i.i.d. random vectors with probability measure Ψ. Consider $\{W_k = (Z_k, U_{k+2})\,; k \geq 0\}$, then it is a Markov chain on $\mathcal{B}(\mathcal{Z}) \otimes \mathcal{B}(\mathbb{R}^m)$ which inherits properties of $\{Z_k\,; k \in \mathbb{N}\}$ in the following sense:*

- *If φ (resp. π) is an irreducibility (resp. invariant) measure of $\{Z_k\,; k \in \mathbb{N}\}$, then $\varphi \times \Psi$ (resp. $\pi \times \Psi$) is an irreducibility (resp. invariant) measure of $\{W_k\,; k \in \mathbb{N}\}$.*
- *The set of integers d such that there exists a d-cycle for $\{Z_k\,; k \in \mathbb{N}\}$ is equal to the set of integers d such that there exists a d–cycle for $\{W_k\,; k \in \mathbb{N}\}$. In particular $\{Z_k\,; k \in \mathbb{N}\}$ and $\{W_k\,; k \in \mathbb{N}\}$ have the same period. Therefore $\{Z_k\,; k \in \mathbb{N}\}$ is aperiodic if and only if $\{W_k\,; k \in \mathbb{N}\}$ is aperiodic.*
- *If C is a small set for $\{Z_k\,; k \in \mathbb{N}\}$, then $C \times \mathbb{R}^m$ is a small set for $\{W_k\,; k \in \mathbb{N}\}$.*
- *If $\{Z_k\,; k \in \mathbb{N}\}$ satisfies a drift condition*

$$\Delta V(z) \leq -\beta h(z) + b\mathbb{1}_C(z) \quad \text{for all } z \in \mathcal{Z}, \tag{4.31}$$

where V is a potential function, $0 < \beta < 1$, $h \geq 0$ is a measurable function and $C \subset \mathcal{Z}$ is a measurable set, then $\{W_k\,; k \in \mathbb{N}\}$ satisfies the following drift condition for all $(z, u) \in \mathcal{Z} \times \mathbb{R}^m$:

$$\Delta \widetilde{V}(z, u) \leq -\beta \widetilde{h}(z, u) + b\mathbb{1}_{C \times \mathbb{R}^m}(z, u) \quad , \tag{4.32}$$

where $\widetilde{V} : (z, u) \mapsto V(z)$ and $\widetilde{h} : (z, u) \mapsto h(z)$.

Remark that the drift condition in (4.31) includes the geometric drift condition by taking $h = V$, the drift condition for h-ergodicity by dividing the equation by β and assuming that $h \geq 1$, for positivity and Harris recurrence by taking $h = 1/\beta$, and for Harris recurrence by taking $h = 0$. This is obtained assuming that V and C satisfy the proper assumptions for the drift to hold.

4.4.3 φ-irreducibility, aperiodicity and T-chain property via deterministic control models

Proving that a Markov chain is a φ-irreducible aperiodic T-chain can sometimes be immediate. In our case however, it is difficult to establish those properties "by hand". We thus resort to tools connecting those properties to stability notions of the underlying control model [140, Chapter 13] [47]. We remind here the different notions needed and refer to [47] for more details. Assume that $\mathcal{Z}$ is an open subset of $\mathbb{R}^n$. We consider a Markov chain that takes the following form

$$Z_{k+1} = F\left(Z_k, \alpha\left(Z_k, U_{k+1}\right)\right), \tag{4.33}$$

where $Z_0 \in \mathcal{Z}$ and for all natural integer k, $F : \mathcal{Z} \times \mathbb{R}^{n\mu} \to \mathcal{Z}$ and $\alpha : \mathcal{Z} \times \mathbb{R}^{n\lambda} \to \mathbb{R}^{n\mu}$ are measurable functions, $U = \left\{U_{k+1} \in \mathbb{R}^{n\lambda}\,;\, k \in \mathbb{N}\right\}$ is a sequence of i.i.d. random vectors. We consider the following assumptions on the model:

- B1. (Z_0, U) are random variables on a probability space $(\Omega, \mathcal{F}, P_{Z_0})$.
- B2. Z_0 is independent of U.
- B3. U is an independent and identically distributed process.
- B4. For all $z \in \mathcal{Z}$, the random variable $\alpha(z, U_1)$ admits a probability density function denoted by p_z, such that the function $(z, u) \longmapsto p_z(u)$ is lower semi-continuous.
- B5. The function $F : \mathcal{Z} \times \mathbb{R}^{n\mu} \to \mathcal{Z}$ is C^1.

We recall the deterministic control model related to (4.33) denoted by CM(F) [47]. It is based on the notion of extended transition map function [139], defined recursively for all $z \in \mathcal{Z}$ as $S_z^0 = z$, and for all $k \in \mathbb{N}\backslash\{0\}$, $S_z^k : \mathbb{R}^{n\mu k} \to \mathcal{Z}$ such that for all $\mathbf{w} = (w_1, \ldots, w_k) \in \mathbb{R}^{n\mu k}$,

$$S_z^k(\mathbf{w}) = F\left(S_z^{k-1}\left(w_1, \ldots, w_{k-1}\right), w_k\right) \quad .$$

Assume in the following that Assumptions B1–B4 are satisfied and that F is continuous.

Let us define the process W for all $k \in \mathbb{N}\backslash\{0\}$ and $z \in \mathcal{Z}$ as $W_1 = \alpha\left(z, U_1\right)$ and $W_k = \alpha\left(S_z^{k-1}(W_1, \ldots, W_{k-1}), U_k\right)$. Then the probability density function of $(W_1, W_2, \ldots, W_k)$ denoted by p_z^k is what is called the extended probability function. It is defined inductively for all $k \in \mathbb{N}\backslash\{0\}$, $\mathbf{w} = (w_1, \ldots, w_k) \in \mathbb{R}^{n\mu k}$ by $p_z^1(w_1) = p_z(w_1)$ and $p_z^k(\mathbf{w}) =$

$p_z^{k-1}(w_1,\dots,w_{k-1})\, p_{S_z^{k-1}(w_1,\dots,w_{k-1})}(w_k)$. For all $k \in \mathbb{N}\backslash\{0\}$ and for all $z \in \mathcal{Z}$, the control sets are finally defined as $\mathcal{O}_z^k = \left\{\mathbf{w} \in \mathbb{R}^{n\mu k} \; ; \; p_z^k(\mathbf{w}) > 0\right\}$. The control sets are open sets since F is continuous and the functions $(z,\mathbf{w}) \mapsto p_z^k(\mathbf{w})$ are lower semi-continuous (see [47] for more details).

The deterministic control model CM(F) is defined recursively for all $k \in \mathbb{N}\backslash\{0\}$, $z \in \mathcal{Z}$ and $(w_1,\dots,w_{k+1}) \in \mathcal{O}_z^{k+1}$ as

$$S_z^{k+1}(w_1,\dots,w_{k+1}) = F\left(S_z^k(w_1,\dots,w_k), w_{k+1}\right) \quad .$$

For $z \in \mathcal{Z}$, $A \in \mathcal{B}(\mathcal{Z})$ and $k \in \mathbb{N}\backslash\{0\}$, we say that $\mathbf{w} \in \mathbb{R}^{n\mu k}$ is a k-steps path from z to A if $\mathbf{w} \in \mathcal{O}_z^k$ and $S_z^k(\mathbf{w}) \in A$. We introduce for $z \in \mathcal{Z}$ and $k \in \mathbb{N}$ the set of all states reachable from z in k steps by CM(F), denoted by $A_+^k(z)$ and defined as $A_+^0(z) = \{z\}$ and $A_+^k(z) = \left\{S_z^k(\mathbf{w}) \; ; \; \mathbf{w} \in \mathcal{O}_z^k\right\}$.

A point $z \in \mathcal{Z}$ is a steadily attracting state if for all $y \in \mathcal{Z}$, there exists a sequence $\left\{y_k \in A_+^k(y) | \, k \in \mathbb{N} \setminus \{0\}\right\}$ that converges to z.

The controllability matrix is defined for $k \in \mathbb{N}\backslash\{0\}$, $z \in \mathcal{Z}$ and $\mathbf{w} \in \mathbb{R}^{n\mu k}$ as the Jacobian matrix of $(w_1,\dots,w_k) \longmapsto S_z^k(w_1,\dots,w_k)$ and denoted by $C_z^k(\mathbf{w})$. Namely, $C_z^k(\mathbf{w}) = \left[\frac{\partial S_z^k}{\partial w_1}(\mathbf{w}) | \dots | \frac{\partial S_z^k}{\partial w_k}(\mathbf{w})\right]$.

If F is C^1, the existence of a steadily attracting state z and a full-rank condition on a controllability matrix of z imply that a Markov chain following (4.33) is a φ-irreducible aperiodic T-chain, as reminded in the next theorem.

Theorem 10. *[47, Theorem 4.4: Practical condition to be a φ-irreducible aperiodic T-chain.] Consider a Markov chain $\{Z_k \, ; k \in \mathbb{N}\}$ following the model (4.33) for which the conditions B1−B5 are satisfied. If there exist a steadily attracting state $z \in \mathcal{Z}$, $k \in \mathbb{N}\backslash\{0\}$ and $\mathbf{w} \in \mathcal{O}_z^k$ such that $\operatorname{rank}\left(C_z^k(\mathbf{w})\right) = n$, then $\{Z_k \, ; k \in \mathbb{N}\}$ is a φ-irreducible aperiodic T-chain, and every compact set is a small set.*

The next lemma allows to loosen the full-rank condition stated above if the control set $\mathcal{O}_z^k$ is dense in $\mathbb{R}^{n\mu k}$.

Lemma 9. *Consider a Markov chain $\{Z_k \, ; k \in \mathbb{N}\}$ following the model (4.33) for which the conditions B1−B5 are satisfied. Assume that there exist a positive integer k and $z \in \mathcal{Z}$ such that the control set $\mathcal{O}_z^k$ is dense in $\mathbb{R}^{n\mu k}$. If there exists $\mathbf{w} \in \mathbb{R}^{n\mu k}$ such that $\operatorname{rank}(C_z^k(\mathbf{w})) = n$, then the rank condition in Theorem 10 is satisfied, $i.e.$ there exists $\mathbf{w} \in \mathcal{O}_z^k$ such that $\operatorname{rank}(C_z^k(\mathbf{w})) = n$.*

Proof. The function $w \longmapsto S_z^k(w)$ is C^1 thanks to [47, Lemma 6.1]. Then by openness of the set of matrices of full rank, there exists an open neighborhood $\mathcal{V}_\mathbf{w}$ of $\mathbf{w}$ such that for all $w \in \mathcal{V}_\mathbf{w}$, $\operatorname{rank}(C_z^k(w)) = n$. By density of $\mathcal{O}_z^k$, the non-empty set $\mathcal{V}_\mathbf{w} \cap \mathcal{O}_z^k$ contains an element $\mathbf{w}$. □

If z is steadily attracting, then there exists under mild assumptions an open set outside of a ball centered at z, with positive measure with respect to the invariant

probability measure of a chain following the model (4.33). We state this result in the next lemma.

Lemma 10. *Consider a Markov chain* $\{Z_k\,; k \in \mathbb{N}\}$ *on* $\mathbb{R}^n$ *following the model* (4.33) *for which the conditions B1–B5 are satisfied. Assume that there exist a steadily attracting state* $z \in \mathbb{R}^n$ *such that* $\mathcal{O}_z^1$ *is dense in* $\mathbb{R}^n$ *and* $w \in \mathcal{O}_z^1$ *with* $\operatorname{rank}\left(C_z^1(w)\right) = n$. *Assume also that* $\{Z_k\,; k \in \mathbb{N}\}$ *is a positive Harris recurrent chain with invariant probability measure* π. *Then there exists* $0 < \epsilon < 1$ *such that* $\pi(\mathbb{R}^n \setminus \overline{\mathbf{B}\left(z, \epsilon\right)}) > 0$.

Proof. A φ-irreducible Markov chain admits a maximal irreducibility measure ψ dominating any other irreducibility measure [140, Theorem 4.0.1]. In other words, for a measurable set A, $\psi(A) = 0$ induces that $\varphi(A) = 0$ for any irreducibility measure φ. Thanks to [140, Theorem 10.4.9], π is equivalent to the maximal irreducibility measure ψ. Since z is steadily attracting, then thanks to [47, Proposition 3.3] and [47, Proposition 4.2], $\operatorname{supp}\ \psi = \overline{A_+(z)} = \overline{\bigcup_{k\in\mathbb{N}} \left\{S_z^k(\mathbf{w})\,; \mathbf{w} \in \mathcal{O}_z^k\right\}}$. We have $\operatorname{rank}\left(C_z^1(w)\right) = n$, therefore the function $F(z, \cdot)$ is not constant. Along with the density of $\mathcal{O}_z^1$, we obtain that there exists $\epsilon > 0$ and a vector $v \in \operatorname{supp}\ \psi$ such that $\|z - v\| = 2\,\epsilon$. By definition of the support, it follows that every open neighborhood of v has a positive measure with respect to π. Since $\mathbb{R}^n \setminus \overline{\mathbf{B}\left(z, \epsilon\right)}$ is an open neighborhood of v, the result of the lemma follows. □

4.5 Stability of the σ-normalized Markov chain $\{Z_k\,; k \in \mathbb{N}\}$

The goal of this section is to prove stability properties of the σ-normalized Markov chain associated to the step-size adaptative $(\mu/\mu_w, \lambda)$-ES defined in Proposition 18, on a function f that is the strictly increasing transformation of either a C^1 scaling-invariant function with a unique global argmin or a nontrivial linear function. We prove that if Assumptions A1–A5 are satisfied and the expected logarithm of the step-size increases on nontrivial linear functions, then the σ-normalized Markov chain is a φ-irreducible aperiodic T-chain that is geometrically ergodic. In particular, it is positive and Harris recurrent.

4.5.1 Irreducibility, aperiodicity and T-chain property of the σ-normalized Markov chain

Prior to establishing Harris recurrence and positivity of the chain $\{Z_k\,; k \in \mathbb{N}\}$, we establish the φ-irreducibility and aperiodicity as well as identify some small sets such that drift conditions can be used. Establishing those properties can be relatively immediate for some ES algorithms (see [19,24]). Yet for the algorithms considered here,

establishing φ-irreducibility turns out to be tricky because the step-size change is a deterministic function of the random input used to update the mean. We hence use the tools developed in [47] and reminded in Section 4.4.3 using the underlying deterministic control model. The chain investigated satisfies $Z_{k+1} = F_w\left(Z_k, \alpha_f\left(x^\star + Z_k, U_{k+1}\right)\right)$ and therefore fits the model (4.33). We prove next that the necessary assumptions needed to use the tools developed in [47] are satisfied if f satisfies F1 or F2. This result relies on [47, Proposition 5.2] ensuring that if f is a continuous scaling-invariant function with Lebesgue negligible level sets, then for all $z \in \mathbb{R}^n$, the random variable $\alpha_f(x^\star + z, U_1)$ admits a probability density function p_z^f such that $(z,u) \mapsto p_z^f(u)$ is lower semi-continuous, $i.e.$ B4 is satisfied.

Proposition 21. *Let f be scaling-invariant with respect to $x^\star$ defined as $\varphi \circ g$ where φ is strictly increasing and g is a continuous scaling-invariant function with Lebesgue negligible level sets. Let $\{Z_k\,; k \in \mathbb{N}\}$ be a σ-normalized Markov chain associated to the step-size adaptive $(\mu/\mu_w, \lambda)$-ES defined as in Proposition 18 satisfying*

$$Z_{k+1} = F_w\left(Z_k, \alpha_f\left(x^\star + Z_k, U_{k+1}\right)\right) \quad .$$

Then model (4.33) follows. In addition, if Assumption A1 is satisfied, then F_w is C^1 and thus B5 is satisfied. If Assumption A5 is satisfied, then Assumptions B1–B4 are satisfied and the probability density function of the random variable $\alpha_f(x^\star + z, U_{k+1})$ denoted by p_z^f and defined in (4.15) satisfies $(z,u) \mapsto p_z^f(u)$ is lower semi-continuous

In particular, if f satisfies F1 or F2, the assumption above on f holds such that the conclusions above are valid.

Proof. It follows from (4.18) that $\{Z_k\,; k \in \mathbb{N}\}$ is a homogeneous Markov chain following model (4.33). By (4.17), F_w is of class C^1 (B5 is satisfied) if A1 is satisfied ($\Gamma : \mathbb{R}^{n\mu} \to \mathbb{R}_+ \backslash \{0\}$ is C^1). If A5 is satisfied, then B1–B3 are also satisfied.

With [47, Proposition 5.2], for all $z \in \mathbb{R}^n$, $\alpha_g(x^\star + z, U_{k+1})$ has a probability density function p_z^g such that $(z,u) \mapsto p_z^g(u)$ is lower semi-continuous, and defined for all $z \in \mathbb{R}^n$ and $u \in \mathbb{R}^{n\mu}$ as in (4.15). With Lemma 5, $\alpha_f = \alpha_g$ and then B4 holds.

A nontrivial linear function is a continuous scaling-invariant function with Lebesgue negligible level sets. Also [184, Proposition 4.2] implies that f still has Lebesgue negligible level sets in the case where it is a C^1 scaling-invariant function with a unique global argmin. □

We show in the following lemma the density of a control set for strictly increasing transformations of continuous scaling-invariant functions with Lebesgue negligible level sets, especially for functions f that satisfy F1 or F2. This is useful for Proposition 22 and for the application of Lemma 9.

Lemma 11. *Let f be a scaling-invariant function defined as $\varphi \circ g$ where φ is strictly increasing and g is a continuous scaling-invariant function with Lebesgue negligible level sets. Assume that $\{Z_k\,; k \in \mathbb{N}\}$ is the σ-normalized Markov chain associated to a*

step-size adaptive $(\mu/\mu_w, \lambda)$-ES as defined in Proposition 18 such that A5 is satisfied. Then for all $z \in \mathbb{R}^n$, the control set $\mathcal{O}_z^1 = \left\{v \in \mathbb{R}^{n\mu} \,;\, p_z^f(v) > 0\right\}$ is dense in $\mathbb{R}^{n\mu}$.

In particular, if f satisfies F1 or F2, the assumption above on f holds and thus the conclusions above are valid.

Proof. By Proposition 21, we obtain that for all $z \in \mathbb{R}^n$, p_z^f is defined as in (4.15). In addition, f has Lebesgue negligible level sets (see [184, Proposition 4.2] and Lemma 5). Therefore $p_z^f > 0$ almost everywhere. Hence we have that $\mathcal{O}_z^1$ is dense in $\mathbb{R}^{n\mu}$. □

According to Theorem 10, to ensure that $\{Z_k \,;\, k \in \mathbb{N}\}$ is a φ-irreducible aperiodic T-chain, we prove in the following that 0 is a steadily attracting state and that there exists $w \in \mathcal{O}_0^1$ such that $\operatorname{rank}\left(C_0^1(w)\right) = n$. We start with the steady attractivity in the next proposition.

Proposition 22. *Let f be a scaling-invariant function defined as $\varphi \circ g$ where φ is strictly increasing and g is a continuous scaling-invariant function with Lebesgue negligible level sets. Assume that $\{Z_k \,;\, k \in \mathbb{N}\}$ is the σ-normalized Markov chain associated to a step-size adaptive $(\mu/\mu_w, \lambda)$-ES as defined in Proposition 18 such that Assumptions A1 and A5 are satisfied. Then 0 is a steadily attracting state of CM(F_w).*

Especially, if f satisfies F1 or F2, the assumption above on f holds and thus the conclusions above are valid.

Proof. We fix $z \in \mathbb{R}^n$ and prove that there exists a sequence $\left\{z_k \in A_+^k(z) \,;\, k \in \mathbb{N}\right\}$ that converges to 0. We construct the sequence recursively as follows.

We define $z_0 = z$ and fix a natural integer k. We define z_{k+1} iteratively as follows. We set $\tilde{v}_k = -\frac{1}{\|w\|^2}\left(w_1 z_k, \ldots, w_\mu z_k\right)$, then $z_k + w^\top \tilde{v}_k = z_k - \frac{1}{\|w\|^2}\sum_{i=1}^{\mu} w_i^2 z_k = 0$. By continuity of F_w and density of $\mathcal{O}_{z_k}^1$ thanks to Lemma 11, there exists $v_k \in \mathcal{O}_{z_k}^1$ such that $\|F_w(z_k, v_k)\| = \|F_w(z_k, v_k) - F_w(z_k, \tilde{v}_k)\| \leq \frac{1}{2^{k+1}}$. Define $z_{k+1} = F_w(z_k, v_k)$. Then the sequence $(z_k)_{k\in\mathbb{N}}$ converges to 0. Now let us show that $z_k \in A_+^k(z)$ for all $k \in \mathbb{N}$.

Since $A_+^0(z) = \{z\}$, then $z_0 = z \in A_+^0(z)$. We fix again a natural integer k and assume that $z_k \in A_+^k(z)$. It is then enough to prove that $z_{k+1} \in A_+^{k+1}(z)$. Recall in the following order that for all $\mathbf{u} \in \mathbb{R}^{n\mu(k+1)}$,

$$
\begin{aligned}
A_+^{k+1}(z) &= \left\{S_z^{k+1}(\mathbf{u}) \,;\, \mathbf{u} \in \mathcal{O}_z^{k+1}\right\} \\
S_z^{k+1}(\mathbf{u}) &= F_w\left(S_z^k\left(u_1, \ldots, u_k\right), u_{k+1}\right) \\
p_z^{f,k+1}(\mathbf{u}) &= p_z^{f,k}\left(u_1, \ldots, u_k\right) p^f_{S_z^k(u_1,\ldots,u_k)}(u_{k+1}) \\
\mathcal{O}_z^{k+1} &= \left\{\mathbf{u} \in \mathbb{R}^{n\mu(k+1)} \;;\; p_z^{f,k+1}(\mathbf{u}) > 0\right\}.
\end{aligned}
$$

Therefore by construction, $p_z^{f,k+1}(v_0, \ldots, v_k) = p_z^{f,k}(v_0, \ldots, v_{k-1}) p^f_{z_k}(v_k) > 0$, hence $(v_0, \ldots, v_k) \in \mathcal{O}_z^{k+1}$. Finally, $z_{k+1} = F_w(z_k, v_k) = S_z^{k+1}(v_0, \ldots, v_k) \in A_+^{k+1}(z)$.

□

The next proposition ensures that the steadily attracting state 0 satisfies also the adequate full-rank condition on a controllability matrix of 0.

Proposition 23. *Let f be a scaling-invariant function defined as $\varphi \circ g$ where φ is strictly increasing and g is a continuous scaling-invariant function with Lebesgue negligible level sets. Assume that $\{Z_k\,; k \in \mathbb{N}\}$ is the σ-normalized Markov chain associated to a step-size adaptive $(\mu/\mu_w, \lambda)$-ES as defined in Proposition 18 such that Assumptions A1 and A5 are satisfied. Then there exists $w \in \mathcal{O}_0^1$ such that $\operatorname{rank}\left(C_0^1(w)\right) = n$.*

In particular, if f satisfies F1 or F2, the assumption above on f holds and thus the conclusions above are valid.

Proof. Lemma 9 along with the density of the control set $\mathcal{O}_0^1$ in Lemma 11 ensure that it is enough to prove the existence of $v \in \mathbb{R}^{n\mu}$ such that $\operatorname{rank}\left(C_0^1(v)\right) = n$. Let us show that the matrix $C_0^1(0) = \frac{\partial S_0^1}{\partial v_1}(0)$ has a full rank, with $S_0^1 : \mathbb{R}^{n\mu} \ni v \longmapsto F_w(0, v) \in \mathbb{R}^n$. This is equivalent to showing that the differential $DS_0^1(0) : \mathbb{R}^{n\mu} \to \mathbb{R}^n$ of S_0^1 at 0 is surjective. Denote by l the linear function $\mathbb{R}^{n\mu} \ni h \longmapsto \sum_{i=1}^{\mu} w_i h_i \in \mathbb{R}^n$. Then $S_0^1 = l/\Gamma$ and then $DS_0^1(h) = Dl(h)\frac{1}{\Gamma(h)} + l(h)D(\frac{1}{\Gamma})(h)$. Since $l(0) = 0$, it follows that $DS_0^1(0) = \frac{l}{\Gamma(0)}$ and finally we obtain that $DS_0^1(0)$ is surjective. □

By applying Propositions 21, 22 and 23 along with Theorem 10, we directly deduce that the σ-normalized Markov chain associated to a step-size adaptive $(\mu/\mu_w, \lambda)$-ES is a φ-irreducible aperiodic T-chain. More formally, the next proposition holds.

Proposition 24. *Let f be a scaling-invariant function defined as $\varphi \circ g$ where φ is strictly increasing and g is a continuous scaling-invariant function with Lebesgue negligible level sets. Assume that $\{Z_k\,; k \in \mathbb{N}\}$ is the σ-normalized Markov chain associated to a step-size adaptive $(\mu/\mu_w, \lambda)$-ES as defined in Proposition 18 such that Assumptions A1 and A5 are satisfied. Then $\{Z_k\,; k \in \mathbb{N}\}$ is a φ-irreducible aperiodic T-chain, and every compact set is a small set.*

In particular, if f satisfies F1 or F2, the assumption above on f holds and thus the conclusions above are valid.

4.5.2 Convergence in distribution of the step-size multiplicative factor

In order to prove that $\{Z_k\,; k \in \mathbb{N}\}$ satisfies a geometric drift condition, we investigate the distribution of $\{Z_k\,; k \in \mathbb{N}\}$ outside of a compact set (small set). Intuitively, when Z_k is very large, *i.e.* $X_k - x^\star$ large compared to the step-size σ_k, the algorithm sees the function f in a small neighborhood from $X_k - x^\star$ where f resembles a linear function (this holds under regularity conditions on the level sets of f). Formally we prove that for all $z \in \mathbb{R}^n$, the step-size multiplicative factor $\Gamma\left(\alpha_f(x^\star + z, U_1)\right)$ converges in

distribution[2] towards the step-size change on nontrivial linear functions $\Gamma^\star_{\text{linear}}$ defined in (4.16).

To do so we derive in Proposition 25 an intermediate result that requires to introduce a specific nontrivial linear function l^f_z defined as follows.

We consider a scaling-invariant function f with respect to its unique global argmin $x^\star$. Then the function $\tilde{f} : x \mapsto f(x^\star + x) - f(x^\star)$ is C^1 scaling-invariant with respect to 0 which is the unique global argmin. Thanks to [184, Corollary 4.1 and Proposition 4.10], there exists a vector in the closed unit ball $z^f_0 \in \overline{\mathbf{B}(0,1)}$ whose $\tilde{f}$-level set is included in the closed unit ball, that is $\mathcal{L}_{\tilde{f},z^f_0} \subset \overline{\mathbf{B}(0,1)}$ and such that for all $z \in \mathcal{L}_{\tilde{f},z^f_0}$, the scalar product between z and the gradient of f at $x^\star + z$ satisfies $z^\top \nabla f(x^\star + z) > 0$. In addition, any half-line of origin 0 intersects the level set $\mathcal{L}_{\tilde{f},z^f_0}$ at a unique point. We denote for all $z \neq 0$ by t^f_z the unique scalar of $(0,1]$ such that $t^f_z \frac{z}{\|z\|}$ belongs to the level set $\mathcal{L}_{\tilde{f},z^f_0} \subset \overline{\mathbf{B}(0,1)}$. We finally define for all $z \neq 0$, the nontrivial linear function l^f_z for all $w \in \mathbb{R}^n$ as

$$l^f_z(w) = w^\top \nabla f\left(x^\star + t^f_z \frac{z}{\|z\|}\right). \tag{4.35}$$

We state below the intermediate result that when $\|z\|$ goes to ∞, the selection random vector $\alpha_f(x^\star + z, U_1)$ has asymptotically the distribution of the selection random vector on the linear function l^f_z. According to Lemma 8, the latter does not depend on the current location and is equal to the distribution of $\alpha_{l^f_z}(0, U_1)$.

Proposition 25. *Let f be a C^1 scaling-invariant function with a unique global argmin. Then for all $\varphi : \mathbb{R}^{n\mu} \to \mathbb{R}$ continuous and bounded,*

$$\lim_{\|z\|\to\infty} \int \varphi(u)\left(p^f_z(u) - p^{l^f_z}_z(u)\right) \mathrm{d}u = 0$$

where l^f_z is defined as in (4.35). In other words, the selection random vectors $\alpha_f(x^\star + z, U_1)$ and $\alpha_{l^f_z}(0, U_1)$ have asymptotically the same distribution when $\|z\|$ goes to ∞.

Proof idea. We sketch the proof idea and refer to Appendix 1.4 for the full proof. Note beforehand that $\alpha_f(x^\star + z, U_1) = \alpha_{\tilde{f}}(z, U_1)$ so that we assume without loss of generality

[2] Recall that a sequence of real-valued random variables $\{Y_k\}_{k\in\mathbb{N}}$ converges in distribution to a random variable Y if $\lim_{k\to\infty} F_{Y_k}(x) = F_Y(x)$ for all continuity point x of F_Y, where F_{Y_k} and F_Y are respectively the cumulative distribution functions of Y_k and Y.

The Portmanteau lemma [34] ensures that $\{Y_k\}_{k\in\mathbb{N}}$ converges in distribution to Y if and only if for all bounded and continuous function φ,

$$\lim_{k\to\infty} \mathbb{E}\left[\varphi(Y_k)\right] = \mathbb{E}\left[\varphi(Y)\right]. \tag{4.34}$$

that $x^\star = 0$ and $f(0) = 0$. If f is a C^1 scaling-invariant function with a unique global argmin, we can construct thanks to [184, Proposition 4.11] a positive number δ_f such that for all element z of the compact set $\mathcal{L}_{f,z_0^f} + \overline{\mathbb{B}(0, 2\delta_f)}$, $z^\top \nabla f(z) > 0$. In particular this result produces a compact neighborhood of the level set $\mathcal{L}_{f,z_0^f}$ where ∇f does not vanish. This helps to establish the limit of $\mathbb{E}\left[\varphi(\alpha_f(z, U_1))\right]$ when $\|z\|$ goes to ∞. We do it in a similar fashion than in [24], by exploiting the uniform continuity of a function that we obtain thanks to its continuity on the compact set $\left(\mathcal{L}_{f,z_0^f} + \overline{\mathbb{B}(0, \delta_f)}\right) \times [0, \delta_f]$. □

Thanks to Proposition 25 and Proposition 17, we can finally state in the next theorem the convergence in distribution of the step-size multiplicative factor for f satisfying F1 towards $\Gamma^\star_{\text{linear}}$ defined in (4.16).

Theorem 11. *Let f be a scaling-invariant function satisfying F1. Assume that $\{U_{k+1}\,; k \in \mathbb{N}\}$ satisfies Assumption A5, Γ is continuous and satisfies Assumption A2, i.e. Γ is invariant under rotation. Then for all natural integer k, $\Gamma\left(\alpha_f(x^\star + z, U_{k+1})\right)$ converges in distribution to $\Gamma^\star_{\text{linear}}$ defined in (4.16), when $\|z\| \to \infty$.*

Proof. Let $\varphi : \Gamma(\mathbb{R}^{n\mu}) \to \mathbb{R}$ be a continuous and bounded function. It is enough to prove that $\lim_{\|z\|\to\infty} \mathbb{E}_{U_1\sim\mathcal{N}_{n\lambda}}\left[\varphi\left(\Gamma\left(\alpha_f(x^\star + z, U_1)\right)\right)\right] = \mathbb{E}\left[\varphi\left(\Gamma^\star_{\text{linear}}\right)\right]$ and apply the Portmanteau lemma.

By Propositions 25, $\lim\limits_{\|z\|\to\infty} \int \varphi\left(\Gamma(u)\right)\left(p_z^f(u) - p_z^{l_z^f}(u)\right) \mathrm{d}u = 0$. Therefore

$$\lim_{\|z\|\to\infty} \mathbb{E}_{U_1\sim\mathcal{N}_{n\lambda}}\left[\varphi\left(\Gamma\left(\alpha_f(x^\star + z, U_1)\right)\right)\right] - \mathbb{E}_{U_1\sim\mathcal{N}_{n\lambda}}\left[\varphi\left(\Gamma\left(\alpha_{l_z^f}(x^\star + z, U_1)\right)\right)\right] = 0.$$

With Propostiion 17, $\mathbb{E}_{U_1\sim\mathcal{N}_{n\lambda}}\left[\varphi\left(\Gamma\left(\alpha_{l_z^f}(x^\star + z, U_1)\right)\right)\right] = \mathbb{E}\left[\varphi\left(\Gamma^\star_{\text{linear}}\right)\right]$. □

4.5.3 Geometric ergodicity of the σ-normalized Markov chain

The convergence in distribution of the step-size multiplicative factor while optimizing a function f that satisfies F1, proven in Theorem 11, allows us to control the behavior of the σ-normalized chain when its norm goes to ∞. More specifically, we use it to show the geometric ergodicity of $\{Z_k\,; k \in \mathbb{N}\}$ defined as in Proposition 18 for f satisfying F1 or F2. Beforehand, let us show the following proposition, which is a first step towards the construction of a geometric drift function.

Proposition 26. *Let f be a scaling-invariant function that satisfies F1 or F2 and $\{Z_k\,; k \in \mathbb{N}\}$ be the σ-normalized Markov chain associated to a step-size adaptive $(\mu/\mu_w, \lambda)$-ES defined as in Proposition 18. We assume that Γ is continuous and Assumptions A2, A3 and A5 are satisfied. Then for all $\alpha > 0$,*

$$\lim_{\|z\|\to\infty} \frac{\mathbb{E}\left[\|Z_1\|^\alpha | Z_0 = z\right]}{\|z\|^\alpha} = \mathbb{E}\left[\frac{1}{[\Gamma^\star_{\text{linear}}]^\alpha}\right]$$

where $\Gamma^\star_{\text{linear}}$ *is the random variable defined in* (4.16) *that represents the step-size change on any nontrivial linear function.*

Proof. Let $z \neq 0$. Since $Z_1 = F_w\left(Z_0, \alpha_f\left(x^\star + Z_0, U_1\right)\right) = \frac{Z_0 + w^\top \alpha_f(x^\star + Z_0, U_1)}{\Gamma(\alpha_f(x^\star + Z_0, U_1))}$, then

$$\frac{\mathbb{E}\left[\|Z_1\|^\alpha | Z_0 = z\right]}{\|z\|^\alpha} - \mathbb{E}\left[\frac{1}{\Gamma\left(\alpha_f(x^\star + z, U_1)\right)^\alpha}\right] = \mathbb{E}\left[\frac{\left\|\frac{z}{\|z\|} + \frac{w^\top \alpha_f(x^\star + z, U_1)}{\|z\|}\right\|^\alpha - 1}{\Gamma\left(\alpha_f(x^\star + z, U_1)\right)^\alpha}\right].$$

The function Γ is lower bounded by $m_\Gamma > 0$ thanks to Assumption A3. In addition, $\|w^\top \alpha_f\left(x^\star + z, U_1\right)\| \leq \|w\|\,\|U_1\|$. Then the term

$$\frac{\left|\left\|\frac{z}{\|z\|} + \frac{1}{\|z\|} w^\top \alpha_f(x^\star + z, U_1)\right\|^\alpha - 1\right|}{\Gamma\left(\alpha_f(x^\star + z, U_1)\right)^\alpha} \tag{4.36}$$

converges almost surely towards 0 when $\|z\|$ goes to ∞, and is bounded (when $\|z\| \geq 1$) by the integrable random variable $\frac{1 + (1 + \|w\|\,\|U_1\|)^\alpha}{m_\Gamma^\alpha}$. Then it follows by the dominated convergence theorem that

$$\lim_{\|z\| \to \infty} \frac{\mathbb{E}\left[\|Z_1\|^\alpha | Z_0 = z\right]}{\|z\|^\alpha} - \mathbb{E}\left[\frac{1}{\Gamma\left(\alpha_f(x^\star + z, U_1)\right)^\alpha}\right] = 0. \tag{4.37}$$

Since $x \mapsto \dfrac{1}{x^\alpha}$ is continuous and bounded on $\Gamma\left(\mathbb{R}^{n\mu}\right) \subset [m_\Gamma, \infty)$, then for f satisfying F1, Theorem 11 implies that $\lim\limits_{\|z\| \to \infty} \mathbb{E}\left[\dfrac{1}{\Gamma\left(\alpha_f(x^\star + z, U_1)\right)^\alpha}\right]$ exists and is equal to $\mathbb{E}\left[\frac{1}{[\Gamma^\star_{\text{linear}}]^\alpha}\right]$. Starting from (4.37) and using Proposition 17 to replace $\mathbb{E}\left[\frac{1}{\Gamma(\alpha_f(x^\star + z, U_1))^\alpha}\right]$ by $\mathbb{E}\left[\frac{1}{[\Gamma^\star_{\text{linear}}]^\alpha}\right]$, the same conclusion holds for f satisfying F2. Thereby

$$\lim_{\|z\| \to \infty} \mathbb{E}\left[\|Z_1\|^\alpha | Z_0 = z\right] / \|z\|^\alpha = \mathbb{E}\left[\frac{1}{[\Gamma^\star_{\text{linear}}]^\alpha}\right].$$

□

We introduce the next two lemmas, that allow to go from Proposition 26 to a formulation with the multiplicative $\log$-step-size factor.

Lemma 12. *Let* f *be a continuous scaling-invariant function with respect to* $x^\star$ *with Lebesgue negligible level sets, let* $z \in \mathbb{R}^n$. *Assume that* Γ *satisfies Assumption A4. Then* $u \mapsto \log\left(\Gamma\left(\alpha_f(x^\star + z, u)\right)\right)$ *is* $\mathcal{N}_{n\lambda}$*-integrable with*

$$\mathbb{E}_{U_1 \sim \mathcal{N}_{n\lambda}}\left[\left|\log\left(\Gamma\left(\alpha_f(x^\star + z, U_1)\right)\right)\right|\right] \leq \frac{\lambda!\, \mathbb{E}_{W_1 \sim \mathcal{N}_{n\mu}}\left[\left|\log \circ \Gamma\right|(W_1)\right]}{(\lambda - \mu)!}. \tag{4.38}$$

Proof. With (4.15), we have $\frac{(\lambda-\mu)!}{\lambda!}\mathbb{E}\left[\left|\log\left(\Gamma\left(\alpha_f(x^\star+z,U_1)\right)\right)\right|\right] \leq \int_{\mathbb{R}^n} |\log\circ\Gamma|(v) \prod_{i=1}^{\mu} p_{\mathcal{N}_n}(v^i)\mathrm{d}v = \mathbb{E}\left[|\log\circ\Gamma|(\mathcal{N}_{n\mu})\right]$, and A4 says that $\log\circ\Gamma$ is $\mathcal{N}_{n\mu}$-integrable. □

The next lemma states that if the expected logarithm of the step-size change is positive, then we can find $\alpha > 0$ such that the limit in Proposition 26 is strictly smaller than 1. This is the key lemma to have the condition in the main results expressed as $\mathbb{E}\left[\log\left(\Gamma^\star_{\text{linear}}\right)\right] > 0$, instead of $\mathbb{E}\left[\frac{1}{[\Gamma^\star_{\text{linear}}]^\alpha}\right] < 1$ for a positive α as was done in [24].

Lemma 13. *Assume that Γ satisfies Assumptions A3 and A4. If $\mathbb{E}\left[\log\left(\Gamma^\star_{\text{linear}}\right)\right] > 0$, then there exists $0 < \alpha < 1$ such that $\mathbb{E}\left[\frac{1}{[\Gamma^\star_{\text{linear}}]^\alpha}\right] < 1$, where $\Gamma^\star_{\text{linear}}$ is defined in (4.16).*

Proof. Lemma 12 ensures that $\log\left(\Gamma^\star_{\text{linear}}\right)$ is integrable. For $\alpha > 0$,

$$\frac{1}{[\Gamma^\star_{\text{linear}}]^\alpha} = \exp\left[-\alpha\log\left(\Gamma^\star_{\text{linear}}\right)\right] = 1 - \alpha\log\left(\Gamma^\star_{\text{linear}}\right) + o(\alpha).$$

Then the random variable $A(\alpha) = \left(\frac{1}{[\Gamma^\star_{\text{linear}}]^\alpha} - 1 + \alpha\log\left(\Gamma^\star_{\text{linear}}\right)\right)/\alpha$ depending on the parameter α converges almost surely towards 0 when α goes to 0.

Let $u \in \mathbb{R}^{n\mu}$ and $\alpha \in (0,1)$. Define the function $\varphi_u : c \mapsto \frac{1}{\Gamma(u)^c} = \exp(-c\log(\Gamma(u)))$ on $[0,\alpha]$. By the mean value theorem, there exists $c_{u,\alpha} \in (0,\alpha)$ such that $\left(\frac{1}{\Gamma(u)^\alpha} - 1\right)/\alpha = \varphi'_u(c_{u,\alpha}) = -\log(\Gamma(u))\frac{1}{\Gamma(u)^{c_{u,\alpha}}}$. In addition, $\frac{1}{\Gamma(u)^{c_{u,\alpha}}} \leq \frac{1}{m_\Gamma^{c_{u,\alpha}}}$ thanks to Assumption A3, and $\frac{1}{m_\Gamma^{c_{u,\alpha}}} = \exp\left(-c_{u,\alpha}\log(m_\Gamma)\right) \leq \exp\left(|\log(m_\Gamma)|\right)$. Therefore

$$|A(\alpha)| \leq \left(1 + \exp\left(|\log(m_\Gamma)|\right)\right)\left|\log\left(\Gamma^\star_{\text{linear}}\right)\right|.$$

The latter is integrable thanks to Assumption A4, and does not depend on α. Then by the dominated convergence theorem, $\mathbb{E}\left[A(\alpha)\right]$ converges to 0 when α goes to 0 or equivalently $\mathbb{E}\left[\frac{1}{[\Gamma^\star_{\text{linear}}]^\alpha}\right] = 1 - \alpha\mathbb{E}\left[\log\left(\Gamma^\star_{\text{linear}}\right)\right] + o(\alpha)$. Hence there exists $0 < \alpha < 1$ small enough such that $\mathbb{E}\left[\frac{1}{[\Gamma^\star_{\text{linear}}]^\alpha}\right] < 1$. □

We now have enough material to state and prove the desired geometric ergodicity of the σ-normalized Markov chain in the following theorem.

Theorem 12. *(Geometric ergodicity) Let f be a scaling-invariant function that satisfies F1 or F2. Let $\{Z_k\,; k \in \mathbb{N}\}$ be the σ-normalized Markov chain associated to a step-size adaptive $(\mu/\mu_w,\lambda)$-ES defined as in Proposition 18 such that Assumptions A1–A5 are satisfied. Assume that $\mathbb{E}\left[\log\left(\Gamma^\star_{\text{linear}}\right)\right] > 0$ where $\Gamma^\star_{\text{linear}}$ is defined in (4.16).*

Then there exists $0 < \alpha < 1$ such that the function $V : z \mapsto 1 + \|z\|^\alpha$ is a geometric drift function for the Markov chain $\{Z_k\,; k \in \mathbb{N}\}$. Therefore $\{Z_k\,; k \in \mathbb{N}\}$ is V-geometrically ergodic, admits an invariant probability measure π and is Harris recurrent.

Proof. Propositions 24 shows that $\{Z_k\,; k \in \mathbb{N}\}$ is a φ-irreducible aperiodic T-chain. Therefore using [140, Theorem 5.5.7 and Theorem 6.2.5], every compact set is a small set. Since we assume that $\mathbb{E}\left[\log\left(\Gamma^\star_{\text{linear}}\right)\right] > 0$, by Lemma 13, there exists $0 < \alpha < 1$ such that $\mathbb{E}\left[\frac{1}{[\Gamma^\star_{\text{linear}}]^\alpha}\right] < 1$. Consider the function $V : z \mapsto 1 + \|z\|^\alpha$. By Proposition 26, $\lim_{\|z\|\to\infty} \mathbb{E}\left[\|Z_1\|^\alpha | Z_0 = z\right] / \|z\|^\alpha = \mathbb{E}\left[\frac{1}{[\Gamma^\star_{\text{linear}}]^\alpha}\right]$. Since $\mathbb{E}\left[V(Z_1)|Z_0 = z\right]/V(z) = \left(1 + \mathbb{E}\left[\|Z_1\|^\alpha | Z_0 = z\right]\right) / \left(1 + \|z\|^\alpha\right)$, then it follows that $\lim_{\|z\|\to\infty} \mathbb{E}\left[V(Z_1)|Z_0 = z\right]/V(z) = \mathbb{E}\left[\frac{1}{[\Gamma^\star_{\text{linear}}]^\alpha}\right]$.

Denote $\gamma = \frac{1}{2}\left(1 + \mathbb{E}\left[\frac{1}{[\Gamma^\star_{\text{linear}}]^\alpha}\right]\right) < 1$. There exists $r > 0$ such that for all $\|z\| > r$

$$\mathbb{E}\left[V(Z_1)|Z_0 = z\right]/V(z) < \gamma. \tag{4.39}$$

In addition, since $\|z + w^\intercal \alpha_f\left(x^\star + z, U_1\right)\| \leq \|z\| + \|w\|\|U_1\|$ then $\mathbb{E}\left[V(Z_1)|Z_0 = z\right] \leq \mathbb{E}\left[\left(\|z\| + \|w\|\|U_1\|\right)^\alpha\right]/m_\Gamma^\alpha$. Since the function $z \mapsto \dfrac{\mathbb{E}\left[\left(\|z\| + \|w\|\|U_1\|\right)^\alpha\right]}{m_\Gamma^\alpha} - \gamma V(z)$ is continuous on the compact $\overline{\mathbf{B}\left(0, r\right)}$, it is bounded on that compact. Denote by $b \in \mathbb{R}_+$ an upper bound. We have proven that for all $z \in \overline{\mathbf{B}\left(0, r\right)}$, $\mathbb{E}\left[V(Z_1)|Z_0 = z\right] \leq \gamma V(z) + b$. This result, along with (4.39), show that for all $z \in \mathbb{R}^n$, $\mathbb{E}\left[V(Z_1)|Z_0 = z\right] \leq \gamma V(z) + b\mathbb{1}_{\overline{\mathbf{B}(0,r)}}(z)$. Therefore $\{Z_k\,; k \in \mathbb{N}\}$ is V-geometrically ergodic. Then thanks to [140, Theorem 15.0.1], $\{Z_k; k \in \mathbb{N}\}$ is positive and Harris recurrent with invariant probability measure π. □

4.6 Proofs of the main results

We present in this section the proofs of the main results stated in Section 4.3 namely Theorems 5, 6 and Proposition 19. Before establishing those main results, we need to prove the integrability of $z \mapsto \log\|z\|$ and $\mathcal{R}_f$ defined in (4.20), with respect to the invariant probability measure of the Markov chain $\{Z_k\,; k \in \mathbb{N}\}$ whose existence is established in Theorem 12.

4.6.1 Integrabilities with respect to the invariant probability measure

For a scaling-invariant function f that satisfies F1 or F2, the limit in Theorem 5 is expressed as $\mathbb{E}_\pi(\mathcal{R}_f)$ where the function $\mathcal{R}_f$ is defined as in (4.20) and π is a probability measure. Therefore the π-integrability of the function $z \mapsto \mathcal{R}_f(z)$ is necessary to obtain Theorem 5. In the following, we present a result stronger than its π-integrability, that is the boundedness of $\mathcal{R}_f$ under some assumptions.

Proposition 27. *Let f be a continuous scaling-invariant function with Lebesgue negligible level sets. Let $\{(X_k, \sigma_k)\,; k \in \mathbb{N}\}$ be the sequence defined in (4.6) and (4.7) such that Assumptions A4 and A5 are satisfied. Then $z \mapsto |\mathcal{R}_f|(z)$ is bounded by $\frac{\lambda!}{(\lambda-\mu)!}\mathbb{E}_{W\sim\mathcal{N}_{n\mu}}\left[|\log\circ\Gamma|(W)\right]$, where the function $z \mapsto \mathcal{R}_f(z)$ is defined as in (4.20).*

If in addition f satisfies F1 or F2, Assumptions A1–A3 are satisfied and $\mathbb{E}\left[\log\left(\Gamma^\star_{\text{linear}}\right)\right] > 0$ where $\Gamma^\star_{\text{linear}}$ is defined in (4.16), then

$$\mathbb{E}_\pi(|\mathcal{R}_f|) = \int |\mathcal{R}_f(z)|\,\pi(dz) < \infty$$

that is $z \mapsto \mathcal{R}_f(z)$ is π-integrable where π is the invariant probability measure of $\{Z_k\,; k \in \mathbb{N}\}$ defined as in Proposition 18.

Proof. Lemma 12 shows that for all $z \in \mathbb{R}^n$, $z \mapsto \log\left(\Gamma\left(\alpha_f(x^\star + z, u)\right)\right)$ is $\mathcal{N}_{n\lambda}$-integrable with $\mathbb{E}\left[|\log\left(\Gamma\left(\alpha_f(x^\star + z, U_1)\right)\right)|\right] \leq \dfrac{\lambda!}{(\lambda-\mu)!}\mathbb{E}_{W\sim\mathcal{N}_{n\mu}}\left[|\log\circ\Gamma|(W)\right]$. Then $|\mathcal{R}_f|$ is bounded since $|\mathcal{R}_f(z)| \leq \frac{\lambda!}{(\lambda-\mu)!}\mathbb{E}_{W\sim\mathcal{N}_{n\mu}}\left[|\log\circ\Gamma|(W)\right]$ for all $z \in \mathbb{R}^n$. If in addition Assumptions A1–A3 are satisfied and $\mathbb{E}\left[\log\left(\Gamma^\star_{\text{linear}}\right)\right] > 0$, Theorem 12 ensures that $\{Z_k\,; k \in \mathbb{N}\}$ is a positive Harris recurrent chain with invariant probability measure π. Hence the integrability with respect to π. □

We prove in the next proposition the π-integrability of the function $z \mapsto \log\|z\|$, where π is the invariant probability measure of the σ-normalized chain, under some assumptions.

Proposition 28. *Let f satisfy F1 or F2 and $\{Z_k\,; k \in \mathbb{N}\}$ be the Markov chain defined as in Proposition 18 such that Assumptions A1–A5 are satisfied. Assume that $\mathbb{E}\left[\log\left(\Gamma^\star_{\text{linear}}\right)\right] > 0$ where $\Gamma^\star_{\text{linear}}$ is defined in (4.16). Then $\{Z_k\,; k \in \mathbb{N}\}$ has an invariant probability measure π and $z \mapsto \log\|z\|$ is π-integrable.*

Proof. Theorem 12 ensures that $\{Z_k\,; k \in \mathbb{N}\}$ is V-geometrically ergodic with invariant probability measure π, where $V : \mathbb{R}^n \ni z \longmapsto 1 + \|z\|^\alpha \in \mathbb{R}_+$. We define for all $z \in \mathbb{R}^n$, $g(z) = \frac{(\lambda-\mu)!}{2\lambda!}\left|\log\|z\|\right|$. Based on [189, Theorem 1], the π-integrability of g is obtained if there exist a set A with $\pi(A) > 0$ such that $\int_A g(z)\pi(\mathrm{d}z) < \infty$, and a measurable function h with $h\mathbb{1}_{A^c} \geq g\mathbb{1}_{A^c}$ such that:

1. $\int_{A^c} P(z, \mathrm{d}y)h(y) < h(z) - g(z)\,, \forall z \in A^c$
2. $\sup_{z\in A} \int_{A^c} P(z, \mathrm{d}y)h(y) < \infty.$

For $z \in \overline{\mathbf{B}\,(0,1)}$ and $v \in \mathbb{R}^{n\mu}$, denote $\varphi(z, v)$ as

$$\varphi(z, v) = p_{\mathcal{N}_{n\mu}}\left(v - \frac{1}{\|w\|^2}\left(w_1 z, \ldots, w_\mu z\right)\right)\mathbb{1}_{\|w^\intercal v\|\leq 1}\ .$$

We prove in a first time that $\lim_{\|z\|\to 0} \int \left|\log\|w^\top v\|\right| \varphi(z,v)\mathrm{d}v < \infty$. We have

$$\begin{aligned}
(2\pi)^{n\mu/2}\varphi(z,v) &= \exp\left(\frac{1}{2}\left(-\|v\|^2 - \frac{\|w\|^2\|z\|^2}{\|w\|^4} + \frac{2(w^\top v)^\top z}{\|w\|^2}\right)\right)\mathbb{1}_{\|w^\top v\|\le 1} \\
&\le \exp\left(\frac{1}{2}\left(-\|v\|^2 - \frac{\|w\|^2\|z\|^2}{\|w\|^4} + \frac{2\|w^\top v\|\|z\|}{\|w\|^2}\right)\right)\mathbb{1}_{\|w^\top v\|\le 1} \\
&= \exp\left(\frac{1}{2}\left(-\|v\|^2 - \frac{\|w\|^2\|z\|^2}{\|w\|^4} + \frac{2\|z\|}{\|w\|^2}\right)\right)\mathbb{1}_{\|w^\top v\|\le 1} \qquad (4.40) \\
&\le (2\pi)^{n\mu/2}\exp\left(1/\|w\|^2\right)\varphi(0,v).
\end{aligned}$$

Since $v \mapsto \left|\log\|w^\top v\|\right|\varphi(0,v)$ is Lebesgue integrable, it follows by the dominated convergence theorem that $z \mapsto \int \left|\log\|w^\top v\|\right|\varphi(z,v)\mathrm{d}v$ is continuous on $\overline{\mathbf{B}(0,1)}$ and $\lim_{\|z\|\to 0}\int \left|\log\|w^\top v\|\right|\varphi(z,v)\mathrm{d}v < \infty$. In addition, $\lim_{\|z\|\to 0} g(z) = \infty$. Then there exists $\epsilon_1 \in (0,1)$ such that for $z \in \overline{\mathbf{B}(0,\epsilon_1)}$:

$$\int \left|\log\|w^\top v\|\right|\varphi(z,v)\mathrm{d}v + 2\,\mathbb{E}_{W\sim\mathcal{N}_{n\mu}}\left[\left|\log\circ\Gamma\right|(W)\right] \le g(z). \qquad (4.41)$$

We define ϵ_2 from Lemma 10 and denote $\epsilon = \min(\epsilon_1,\epsilon_2)$. Define $A = \mathbb{R}^n \setminus \overline{\mathbf{B}(0,\epsilon)}$. Then from Lemma 10 it follows that $\pi(A) > 0$. Note also that $A^c = \overline{\mathbf{B}(0,\epsilon)}$. In addition, g is dominated by the π-integrable function V around ∞, then $\int_A g(z)\pi(\mathrm{d}z) < \infty$. We define now the function h for all $z \in \mathbb{R}^n$ as $h(z) = 2g(z)\mathbb{1}_{A^c}(z)$. Then $h\mathbb{1}_{A^c} \ge g\mathbb{1}_{A^c}$.

It remains to verify the items 1 and 2 from above to obtain the π-integrability of g. To do so, we give in the following an upper bound of $K(z) = \int_{A^c} P(z,\mathrm{d}y)h(y)$ for all $z \in \mathbb{R}^n$. We have $K(z) = -\frac{(\lambda-\mu)!}{\lambda!}\mathbb{E}_z\left[\mathbb{1}_{\overline{\mathbf{B}(0,\epsilon)}}(Z_1)\log\|Z_1\|\right]$. Then $K(z) \le -\frac{(\lambda-\mu)!}{\lambda!}\int_{\|z+w^\top v\|\le\Gamma(v)} \log\frac{\|z+w^\top v\|}{\Gamma(v)} p_z^f(v)\mathrm{d}v$. With (4.15), we obtain that $\frac{(\lambda-\mu)!}{\lambda!}p_z^f \le p_{\mathcal{N}_{n\mu}}$. Then $K(z) \le \int |\log(\Gamma(v))|\,p_{\mathcal{N}_{n\mu}}(v)\mathrm{d}v +$
$\int_{\|z+w^\top v\|\le\Gamma(v)} \left|\log\|z+w^\top v\|\right| p_{\mathcal{N}_{n\mu}}(v)\mathrm{d}v$.

We split the latter integral between the events $\{\|z+w^\top v\| \le \min(1,\Gamma(v))\}$ and the events $\{1 < \|z+w^\top v\| \le \Gamma(v)\}$. Then

$K(z) \le \int_{\|z+w^\top v\|\le\min(1,\Gamma(v))} \left|\log\|z+w^\top v\|\right| p_{\mathcal{N}_{n\mu}}(v)\mathrm{d}v +$
$\int_{\Gamma(v)\ge 1} \log(\Gamma(v))p_{\mathcal{N}_{n\mu}}(v)\mathrm{d}v + \int |\log(\Gamma(v))|\,p_{\mathcal{N}_{n\mu}}(v)\mathrm{d}v$. Hence

$K(z) \leq 2\,\mathbb{E}_{W\sim\mathcal{N}_{n\mu}}\left[|\log\circ\Gamma|\,(W)\right] - \int_{\|z+w^\top v\|\leq 1} \log\|z+w^\top v\| p_{\mathcal{N}_{n\mu}}(v)\mathrm{d}v$. With a translation $v \to v - \frac{1}{\|w\|^2}\,(w_1 z, \ldots, w_\mu z)$ within the last integrand, we obtain:

$$K(z) \leq 2\,\mathbb{E}_{W\sim\mathcal{N}_{n\mu}}\left[|\log\circ\Gamma|\,(W)\right] + \int \left|\log\|w^\top v\|\right| \varphi(z,v)\mathrm{d}v. \tag{4.42}$$

Equations (4.41) and (4.42) show that for $z \in A^c = \overline{\mathbf{B}\,(0,\epsilon)}$, $\int_{A^c} P(z,\mathrm{d}y)h(y) \leq g(z) = h(z) - g(z)$. Therefore the item 1 follows.

With (4.40), it follows that there exist $c_1 > 0$ and $c_2 > 0$ such that for $\|z\| \geq c_1$ and $v \in \mathbb{R}^n$, $\varphi(z,v) \leq c_2\varphi(0,v)$. Thanks to the dominated convergence theorem, $\lim_{\|z\|\mapsto\infty} \int \left|\log\|w^\top v\|\right| \varphi(z,v)\mathrm{d}v = 0$. Therefore that integral is bounded outside of a compact. In addition, $z \mapsto \int \left|\log\|w^\top v\|\right| \varphi(z,v)\mathrm{d}v$ is continuous and is bounded on any compact included in $\overline{A}$. Then along with (4.42) it follows that $\sup_{z\in A} \int_{A^c} P(z,\mathrm{d}y)h(y) < \infty$. Hence the item 2 is also satisfied, which ends the integrability proof of $z \mapsto \log\|z\|$. □

4.6.2 Proof of Theorem 5

We start by proving (4.21). Theorem 12 ensures that $\{Z_k\,; k \in \mathbb{N}\}$ is a positive Harris recurrent chain with invariant probability measure π. With (4.25), we have that

$$\begin{aligned}\frac{1}{k}\log\frac{\|X_k - x^\star\|}{\|X_0 - x^\star\|} &= \frac{1}{k}\log\frac{\|Z_k\|}{\|Z_0\|} + \frac{1}{k}\log\left(\frac{\sigma_k}{\sigma_0}\right)\\ &= \frac{1}{k}\sum_{t=0}^{k-1}\log\frac{\|Z_{t+1}\|}{\|Z_t\|} + \frac{1}{k}\sum_{t=0}^{k-1}\log(\Gamma(\alpha_f(x^\star + Z_t, U_{t+1})))\ ,\end{aligned}$$

where Γ and α_f are defined in (4.7) and in (4.3).

Since $z \mapsto \log\|z\|$ is π-integrable, Theorem 7 ensures that the LLN holds with $\lim_{k\to\infty}\frac{1}{k}\sum_{t=0}^{k-1}\log\frac{\|Z_{t+1}\|}{\|Z_t\|} = \int \log\left(\|z\|\right)\pi(\mathrm{d}z) - \int \log\left(\|z\|\right)\pi(\mathrm{d}z) = 0$.

Let us consider the chain $\{W_k = (Z_k, U_{k+2})\,; k \in \mathbb{N}\}$. Then thanks to Proposition 20, $\{W_k = (Z_k, U_{k+2})\,; k \in \mathbb{N}\}$ is geometrically ergodic with invariant probability measure $\pi \times \mathcal{N}_{n\lambda}$. Define the function g for $((z_1,u_3),\,(z_2,u_4)) \in \left(\mathbb{R}^n \times \mathbb{R}^{n\lambda}\right)^2$ as $g\left((z_1,u_3),(z_2,u_4)\right) = \log(\Gamma(\alpha_f(x^\star + z_2, u_3)))$. We have by Proposition 27 that for all natural integer t,

$$\mathbb{E}_{\pi\times\mathcal{N}_{n\lambda}}(|g(W_t, W_{t+1})|) \leq \frac{\lambda!}{(\lambda-\mu)!}\mathbb{E}_{Y\sim\mathcal{N}_{n\mu}}\left[|\log\circ\Gamma|\,(Y)\right] < \infty\ .$$

By Theorem 9 or Corollary 4, for any initial distribution, $\frac{1}{k}\sum_{t=0}^{k-1}\log(\Gamma(\alpha_f(x^\star + Z_t, U_{t+1})))$ converges almost surely towards $\mathbb{E}_{\pi\times\mathcal{N}_{n\lambda}}(g(W_1,W_2)) = \mathbb{E}_\pi(\mathcal{R}_f)$.

Let us prove now (4.22). Equations (4.28) and (4.29) show that for all $z \in \mathbb{R}^n$

$$\begin{aligned}\mathbb{E}_z\left[\log\frac{\|X_{k+1}-x^\star\|}{\|X_k-x^\star\|}\right] &= \mathbb{E}_z\left[\log\frac{\|Z_{k+1}\|}{\|Z_k\|}\right] + \mathbb{E}_z\left[\log\frac{\sigma_{k+1}}{\sigma_k}\right] \\ &= \int P^{k+1}(z,\mathrm{d}y)\log(\|y\|) - \int P^k(z,\mathrm{d}y)\log(\|y\|) \\ &\quad + \int P^k(z,\mathrm{d}y)\mathcal{R}_f(y).\end{aligned}$$

Define h on $\mathbb{R}^n$ as $h(z) = 1 + |\log\|z\||$ for all $z \in \mathbb{R}^n$ which is π-integrable thanks to Proposition 28. Then $z \mapsto \log\|z\|$ is π-integrable, and by [140, Theorem 14.0.1], for $z \in \{y \in \mathbb{R}^n\,; V(y) < \infty\} = \mathbb{R}^n$, $\lim_{k\to\infty}\|P^k(z,\cdot) - \pi\|_h = 0$. Then $\lim_{k\to\infty}\int P^{k+1}(z,\mathrm{d}y)\log(\|y\|) = \lim_{k\to\infty}\int P^k(z,\mathrm{d}y)\log(\|y\|) = \int \log(\|y\|)\pi(\mathrm{d}y)$. In addition, $|\mathcal{R}_f|/h$ is bounded, then $\lim_{k\to\infty}\int P^k(z,\mathrm{d}y)\mathcal{R}_f(y) = \int \mathcal{R}_f(y)\pi(\mathrm{d}y) = \mathbb{E}_\pi(\mathcal{R}_f)$, and finally (4.22) follows:

$$\lim_{k\to\infty}\mathbb{E}_{\frac{x-x^\star}{\sigma}}\left[\log\frac{\|X_{k+1}-x^\star\|}{\|X_k-x^\star\|}\right] = \lim_{k\to\infty}\mathbb{E}_{\frac{x-x^\star}{\sigma}}\left[\log\frac{\sigma_{k+1}}{\sigma_k}\right] = \mathbb{E}_\pi(\mathcal{R}_f).$$

We also note that if f satisfies F2, then thanks to Proposition 17, for all $z \in \mathbb{R}^n$, $\mathcal{R}_f(z) = \mathbb{E}\left[\log\left(\Gamma^\star_{\text{linear}}\right)\right]$, hence $\mathcal{R}_f$ is constant. Then $\mathbb{E}_\pi\left(\mathcal{R}_f\right) = \int \mathcal{R}_f(z)\pi(\mathrm{d}z) = \mathbb{E}\left[\log\left(\Gamma^\star_{\text{linear}}\right)\right]$. If in addition $\mathbb{E}\left[\log\left(\Gamma^\star_{\text{linear}}\right)\right] > 0$, we obtain that $\|X_k\|$ and σ_k both diverge to ∞ when k goes to ∞.

4.6.3 Proof of Theorem 6

Thanks to Proposition 27, $|\mathcal{R}_f|$ is bounded. And then there exists a positive constant K large enough such that $\mathcal{R}_f^2 \le KV$ where V is the geometric drift function of $\{Z_k\,; k \in \mathbb{N}\}$ given by Theorem 12. Then KV remains a geometric drift function. Thanks to Theorem 8, the constant γ defined as $\mathbb{E}_\pi\left[\left(\mathcal{R}_f(Z_0) - \mathbb{E}_\pi(\mathcal{R}_f)\right)^2\right] + 2\sum_{k=1}^\infty \mathbb{E}_\pi\left[\left(\mathcal{R}_f(Z_0) - \mathbb{E}_\pi(\mathcal{R}_f)\right)\left(\mathcal{R}_f(Z_k) - \mathbb{E}_\pi(\mathcal{R}_f)\right)\right]$ is well defined, non-negative, finite and $\lim_{t\to\infty}\frac{1}{t}\mathbb{E}_\pi\left[\left(S_t(\mathcal{R}_f) - t\mathbb{E}_\pi(\mathcal{R}_f)\right)^2\right] = \gamma^2$.

Moreover if $\gamma^2 > 0$, then the CLT holds for any z_0 as follows

$$\lim_{t\to\infty} P_{z_0}\left((t\gamma^2)^{-\frac{1}{2}}\left(S_t(\mathcal{R}_f) - t\mathbb{E}_\pi(\mathcal{R}_f)\right) \le z\right) = \int_{-\infty}^z \frac{1}{\sqrt{2\pi}}e^{-u^2/2}\mathrm{d}u.$$

Which can be rephrased as $\frac{1}{\sqrt{t\gamma^2}}\left(S_t(\mathcal{R}_f) - t\mathbb{E}_\pi(\mathcal{R}_f)\right)$ converges in distribution to $\mathcal{N}(0,1)$ when $t \to \infty$. And if $\gamma = 0$, then $\lim_{t\to\infty}\left(S_t(\mathcal{R}_f) - t\,\mathbb{E}_\pi(\mathcal{R}_f)\right)/\sqrt{t} = 0$ *a.s.*

4.6.4 Proof of Proposition 19

We first prove the statement related to the $(\mu/\mu_w, \lambda)$-CSA1-ES. Then we show the condition regarding $(\mu/\mu_w, \lambda)$-xNES. And finally we prove the general practical condition that allows to obtain (4.24).

If m is a positive integer and $u = \left(u^1, \dots, u^m\right) \in \mathbb{R}^{nm}$, we denote $u_1 = \left(u_1^1, \dots, u_1^m\right)$ and $u_{-1} = \left(u_{-1}^1, \dots, u_{-1}^m\right)$ where $u_{-1}^i = \left(u_2^i, \dots, u_n^i\right)$ for $i = 1, \dots, m$. Define the nontrivial linear function $l^\star$ such that $l^\star(x) = x_1$ for $x \in \mathbb{R}^n$, and denote by e_1 the unit vector $(1, \dots, 0)$.

Part 1. It is enough to prove that $\mathbb{E}_{U_1 \sim \mathcal{N}_{n\lambda}}\left[\log\left(\Gamma_{\text{CSA1}}\left(\alpha_{l^\star}(e_1, U_1)\right)\right)\right]$ has the same sign than $\mathbb{E}\left[\left(\sum_{i=1}^{\mu} \frac{w_i}{\|w\|} \mathcal{N}^{i:\lambda}\right)^2\right] - 1$, and apply Theorem 5. We have:

$$\mathbb{E}_{U_1 \sim \mathcal{N}_{n\lambda}}\left[\log\left(\Gamma_{\text{CSA1}}\left(\alpha_{l^\star}(e_1, U_1)\right)\right)\right] = \frac{1}{2d_\sigma \|w\|^2 n}\left(\mathbb{E}_{U_1 \sim \mathcal{N}_{n\lambda}}\left[\left\| \sum_{i=1}^{\mu} w_i \left(\alpha_{l^\star}(e_1, U_1)\right)_i \right\|^2\right] - \|w\|^2 n\right).$$

Therefore it is enough to show that

$$\mathbb{E}\left[\left\|\sum_{i=1}^{\mu} \frac{w_i}{\|w\|} \left(\alpha_{l^\star}(e_1, U_1)\right)_i\right\|^2\right] - n = \mathbb{E}\left[\left(\sum_{i=1}^{\mu} \frac{w_i}{\|w\|} \mathcal{N}^{i:\lambda}\right)^2\right] - 1. \tag{4.43}$$

Recall that the probability density function of $\alpha_{l^\star}(e_1, U_1)$ is $p_{e_1}^{l^\star}$ defined for all $u \in \mathbb{R}^{n\mu}$ as $p_{e_1}^{l^\star}(u) = \frac{\lambda!}{(\lambda-\mu)!}(1 - Q_{e_1}^{l^\star}(u^\mu))^{\lambda-\mu} \prod_{i=1}^{\mu-1} \mathbb{1}_{\{l^\star(u^i) < l^\star(u^{i+1})\}} \prod_{i=1}^{\mu} p_{\mathcal{N}_n}(u^i)$.

Denote $A = \mathbb{E}_{U_1 \sim \mathcal{N}_{n\lambda}}\left[\left\|\sum_{i=1}^{\mu} \frac{w_i}{\|w\|} \left(\alpha_{l^\star}(e_1, U_1)\right)_i\right\|^2\right]$. It follows that

$$A = \frac{\lambda!}{(\lambda-\mu)!} \int \left\|\sum_{i=1}^{\mu} \frac{w_i}{\|w\|} u^i\right\|^2 (1 - Q_{e_1}^{l^\star}(u^\mu))^{\lambda-\mu} \prod_{j=1}^{\mu-1} \mathbb{1}_{\{l^\star(u^j) < l^\star(u^{j+1})\}}$$
$$\prod_{j=1}^{\mu} p_{\mathcal{N}_n}(u^j) \mathrm{d}u = \int \left(\left\|\sum_{i=1}^{\mu} \frac{w_i}{\|w\|} u_1^i\right\|^2 + \left\|\sum_{i=1}^{\mu} \frac{w_i}{\|w\|} u_{-1}^i\right\|^2\right) P\left(\mathcal{N} > u_1^\mu\right)^{\lambda-\mu}$$
$$\prod_{j=1}^{\mu-1} \mathbb{1}_{\left\{u_1^j < u_1^{j+1}\right\}} \prod_{j=1}^{\mu} p_{\mathcal{N}}(u_1^j) \prod_{j=1}^{\mu} p_{\mathcal{N}_{n-1}}(u_{-1}^j) \, \mathrm{d}u.$$

If we expand the integrand, the first term gives $\mathbb{E}\left[\left(\sum_{i=1}^{\mu} \frac{w_i}{\|w\|} \mathcal{N}^{i:\lambda}\right)^2\right]$, as it is the 1-D version of $\mathbb{E}_{U_1 \sim \mathcal{N}_{n\lambda}}\left[\left\|\sum_{i=1}^{\mu} \frac{w_i}{\|w\|} \left(\alpha_{l^\star}(e_1, U_1)\right)_i\right\|^2\right]$. Denote $B = \mathbb{E}\left[\left(\sum_{i=1}^{\mu} \frac{w_i}{\|w\|} \mathcal{N}^{i:\lambda}\right)^2\right]$ and $C = B - A$. It follows that

$$\frac{(\lambda-\mu)!}{\lambda!} C = \int \left\|\sum_{i=1}^{\mu} \frac{w_i}{\|w\|} u_{-1}^i\right\|^2 P\left(\mathcal{N} > u_1^\mu\right)^{\lambda-\mu} \prod_{j=1}^{\mu-1} \mathbb{1}_{\left\{u_1^j < u_1^{j+1}\right\}} \prod_{j=1}^{\mu} p_{\mathcal{N}}(u_1^j) p_{\mathcal{N}_{n-1}}(u_{-1}^j) \mathrm{d}u.$$

Then $C = \int_{\mathbb{R}^\mu} \frac{\lambda!}{(\lambda-\mu)!} P\left(\mathcal{N} > u_1^\mu\right)^{\lambda-\mu} \prod_{j=1}^{\mu-1} \mathbb{1}_{\left\{u_1^j < u_1^{j+1}\right\}} \prod_{j=1}^{\mu} p_{\mathcal{N}}(u_1^j) \mathrm{d}u_1$
$\int_{\mathbb{R}^{(n-1)\mu}} \left\| \sum_{i=1}^{\mu} \frac{w_i}{\|w\|} u_{-1}^i \right\|^2 \prod_{j=1}^{\mu} p_{\mathcal{N}_{n-1}}(u_{-1}^j) \mathrm{d}u_{-1}$.

The first integral equals 1 as it is the integral of a probability density function. The second integral is equal to $\mathbb{E}\left[\| \sum_{i=1}^{\mu} \frac{w_i}{\|w\|} W_i \|^2\right]$ where $W_1, \ldots, W_\mu$ are i.i.d. random variables of law $\mathcal{N}_{n-1}$. Then the law of $\sum_{i=1}^{\mu} \frac{w_i}{\|w\|} W_i$ is $\mathcal{N}_{n-1}$. Then $\mathbb{E}\left[\left\| \sum_{i=1}^{\mu} \frac{w_i}{\|w\|} W_i \right\|^2\right] = n-1$. Hence $\mathbb{E}_{U_1 \sim \mathcal{N}_{n\lambda}}\left[\left\| \sum_{i=1}^{\mu} \frac{w_i}{\|w\|} \left(\alpha_{l^\star}(e_1, U_1)\right)_i \right\|^2\right] - \mathbb{E}\left[\left(\sum_{i=1}^{\mu} \frac{w_i}{\|w\|} \mathcal{N}^{i:\lambda}\right)^2\right] = n-1$, which induces (4.43) and ends this part.

Part 2. For the second item, we show that $\mathbb{E}_{U_1 \sim \mathcal{N}_{n\lambda}}\left[\log\left(\Gamma_{\mathrm{xNES}}\left(\alpha_{l^\star}(e_1, U_1)\right)\right)\right]$ has the same sign than $\sum_{i=1}^{\mu} \frac{w_i}{\sum_{j=1}^{\mu} w_j} \mathbb{E}\left[\left(\mathcal{N}^{i:\lambda}\right)^2\right] - 1$, and apply Theorem 5. We have

$$\mathbb{E}_{U_1 \sim \mathcal{N}_{n\lambda}}\left[\log\left(\Gamma_{\mathrm{xNES}}\left(\alpha_{l^\star}(e_1, U_1)\right)\right)\right] = \frac{1}{2 d_\sigma n \sum_{i=1}^{\mu} w_i} \sum_{i=1}^{\mu} w_i \left(\mathbb{E}_{U_1 \sim \mathcal{N}_{n\lambda}}\left[\| \left(\alpha_{l^\star}(e_1, U_1)\right)_i \|^2\right] - n\right).$$

Then it is enough to show: $\sum_{i=1}^{\mu} w_i \left(\mathbb{E}_{U_1 \sim \mathcal{N}_{n\lambda}}\left[\| \left(\alpha_{l^\star}(e_1, U_1)\right)_i \|^2\right] - n\right) = \sum_{i=1}^{\mu} w_i \mathbb{E}\left[\left(\mathcal{N}^{i:\lambda}\right)^2\right] - \sum_{i=1}^{\mu} w_i$. Denote $A = \sum_{i=1}^{\mu} w_i \mathbb{E}_{U_1 \sim \mathcal{N}_{n\lambda}}\left[\| \left(\alpha_{l^\star}(e_1, U_1)\right)_i \|^2\right]$. It follows

$$A = \frac{\lambda!}{(\lambda-\mu)!} \int \sum_{i=1}^{\mu} w_i \|u^i\|^2 (1 - Q_{e_1}^{l^\star}(u^\mu))^{\lambda-\mu} \prod_{j=1}^{\mu-1} \mathbb{1}_{\{l^\star(u^j) < l^\star(u^{j+1})\}}$$
$$\prod_{j=1}^{\mu} p_{\mathcal{N}_n}(u^j) \mathrm{d}u = \frac{\lambda!}{(\lambda-\mu)!} \int \left(\sum_{i=1}^{\mu} w_i \|u_1^i\|^2 + \sum_{i=1}^{\mu} w_i \|u_{-1}^i\|^2 \right)$$
$$P\left(\mathcal{N} > u_1^\mu\right)^{\lambda-\mu} \prod_{j=1}^{\mu-1} \mathbb{1}_{\left\{u_1^j < u_1^{j+1}\right\}} \prod_{j=1}^{\mu} p_{\mathcal{N}}(u_1^j) \prod_{j=1}^{\mu} p_{\mathcal{N}_{n-1}}(u_{-1}^j) \, \mathrm{d}u.$$

Therefore if we expand the integrand, the integral of the first term of the integrand is equal to $\frac{(\lambda-\mu)!}{\lambda!} \sum_{i=1}^{\mu} w_i \mathbb{E}\left[\left(\mathcal{N}^{i:\lambda}\right)^2\right]$. Denote $B = \sum_{i=1}^{\mu} w_i \mathbb{E}\left[\left(\mathcal{N}^{i:\lambda}\right)^2\right]$ and $C = A - B$. Then
$\frac{(\lambda-\mu)!}{\lambda!} C = \int \sum_{i=1}^{\mu} w_i \|u_{-1}^i\|^2 \left(P\left(\mathcal{N} > u_1^\mu\right)\right)^{\lambda-\mu}$
$\prod_{j=1}^{\mu-1} \mathbb{1}_{\left\{u_1^j < u_1^{j+1}\right\}} \prod_{j=1}^{\mu} p_{\mathcal{N}}(u_1^j) p_{\mathcal{N}_{n-1}}(u_{-1}^j) \mathrm{d}u.$

Then $C = \int_{\mathbb{R}^\mu} \frac{\lambda!}{(\lambda-\mu)!} \prod_{j=1}^{\mu-1} \mathbb{1}_{\left\{u_1^j < u_1^{j+1}\right\}} P\left(\mathcal{N} > u_1^\mu\right)^{\lambda-\mu} \prod_{j=1}^{\mu} p_{\mathcal{N}}(u_1^j) \mathrm{d}u_1$
$\int_{\mathbb{R}^{(n-1)\mu}} \sum_{i=1}^{\mu} w_i \|u_{-1}^i\|^2 \prod_{j=1}^{\mu} p_{\mathcal{N}_{n-1}}(u_{-1}^j) \mathrm{d}u_{-1}$. The first integral is equal to 1 as it is the integral

of a probability density function. The second integral is equal to $\sum_{i=1}^{\mu} w_i \mathbb{E}\left[\|\mathcal{N}_{n-1}\|^2\right] = (n-1)\sum_{i=1}^{\mu} w_i$. We finally have that

$$\sum_{i=1}^{\mu} w_i \mathbb{E}_{U_1 \sim \mathcal{N}_{n\lambda}}\left[\| \left(\alpha_{l^\star}(e_1, U_1)\right)_i \|^2\right] - \sum_{i=1}^{\mu} w_i \mathbb{E}\left[\left(\mathcal{N}^{i:\lambda}\right)^2\right] = (n-1)\sum_{i=1}^{\mu} w_i.$$

Part 3. If $(X_1, \dots, X_\lambda)$ is distributed according to $\left(\mathcal{N}^{1:\lambda}, \dots, \mathcal{N}^{\lambda:\lambda}\right)$, then $X_1 \leq \dots \leq X_\lambda$ and then $-X_\lambda \leq \dots \leq -X_1$. Therefore $(-X_\lambda, \dots, -X_1)$ is also distributed according to $\left(\mathcal{N}^{1:\lambda}, \dots, \mathcal{N}^{\lambda:\lambda}\right)$. Assume that $\lambda \geq 3$ and $\mu > \frac{\lambda}{2}$. We show the results in two parts.

Part 3.1. First we assume that $w_1 = \dots = w_\mu = \frac{1}{\mu}$. In this case, we have to prove that: $1 < \sum_{i=1}^{\mu} \frac{w_i}{\sum_{j=1}^{\mu} w_j} \mathbb{E}\left[\left(\mathcal{N}^{i:\lambda}\right)^2\right] = \frac{1}{\mu}\sum_{i=1}^{\mu} \mathbb{E}\left[\left(\mathcal{N}^{i:\lambda}\right)^2\right]$. Since $\mathcal{N}^{1:\lambda} \leq \dots \leq \mathcal{N}^{\lambda:\lambda}$ is equivalent to $-\mathcal{N}^{\lambda:\lambda} \leq \dots \leq -\mathcal{N}^{1:\lambda}$, then $\left(\mathcal{N}^{1:\lambda}, \dots, \mathcal{N}^{\lambda:\lambda}\right)$ has the distribution of $\left(-\mathcal{N}^{\lambda:\lambda}, \dots, -\mathcal{N}^{1:\lambda}\right)$. And then for $i = 1, \dots, \lambda$, $\left(\mathcal{N}^{i:\lambda}\right)^2$ has the distribution of $\left(\mathcal{N}^{\lambda-i+1:\lambda}\right)^2$. It follows that:

$$\sum_{i=1}^{\lambda} \mathbb{E}\left[\left(\mathcal{N}^{i:\lambda}\right)^2\right] = 2\sum_{i=1}^{\mu} \mathbb{E}\left[\left(\mathcal{N}^{i:\lambda}\right)^2\right] + \sum_{i=\mu+1}^{\lambda-\mu} \mathbb{E}\left[\left(\mathcal{N}^{i:\lambda}\right)^2\right]. \tag{4.44}$$

Moreover,

$$\sum_{i=1}^{\lambda} \mathbb{E}\left[\left(\mathcal{N}^{i:\lambda}\right)^2\right] = \sum_{i=1}^{\lambda} \mathbb{E}\left[\left(\mathcal{N}^{i}\right)^2\right] = \lambda \quad , \tag{4.45}$$

meaning that we lose the selection effect of the order statistics when we do the above summation. Equations (4.44) and (4.45) ensure that

$$2\sum_{i=1}^{\mu} \mathbb{E}\left[\left(\mathcal{N}^{i:\lambda}\right)^2\right] + \sum_{i=\mu+1}^{\lambda-\mu} \mathbb{E}\left[\left(\mathcal{N}^{i:\lambda}\right)^2\right] = \lambda \quad . \tag{4.46}$$

For any $j \in \{\mu+1, \dots, \lambda-\mu\}$ and any $i \in \{1 \dots, \mu\}$, $\mathcal{N}^{i:\lambda} \leq \mathcal{N}^{j:\lambda} \leq \mathcal{N}^{\lambda+1-i:\lambda}$. Therefore if $\mathcal{N}^{j:\lambda} \geq 0$, $\left(\mathcal{N}^{j:\lambda}\right)^2 \leq \left(\mathcal{N}^{\lambda+1-i:\lambda}\right)^2$, and if $\mathcal{N}^{j:\lambda} \leq 0$, $\left(\mathcal{N}^{j:\lambda}\right)^2 \leq \left(\mathcal{N}^{i:\lambda}\right)^2$.

Since $\left(\mathcal{N}^{\lambda+1-i:\lambda}\right)^2$ has the distribution of $\left(\mathcal{N}^{i:\lambda}\right)^2$, then for all $j \in \{\mu+1, \dots, \lambda-\mu\}$ and $i \in \{1 \dots, \mu\}$: $\left(\mathcal{N}^{j:\lambda}\right)^2 \leq \left(\mathcal{N}^{i:\lambda}\right)^2$, and it is straightforward to see that the we do not have almost sure equality. It then follows that for all $j \in \{\mu+1, \dots, \lambda-\mu\}$ [3] and $i \in \{1 \dots, \mu\} : \mathbb{E}\left[\left(\mathcal{N}^{j:\lambda}\right)^2\right] < \mathbb{E}\left[\left(\mathcal{N}^{i:\lambda}\right)^2\right]$. Therefore for all $j \in \{\mu+1, \dots, \lambda-\mu\}$:

$$\mathbb{E}\left[\left(\mathcal{N}^{j:\lambda}\right)^2\right] < \frac{1}{\mu}\sum_{i=1}^{\mu} \mathbb{E}\left[\left(\mathcal{N}^{i:\lambda}\right)^2\right]. \tag{4.47}$$

[3] Note that the set $\{\mu+1, \dots, \lambda-\mu\}$ is not empty since $1 \leq \mu < \frac{\lambda}{2}$.

With (4.47) and (4.46), we have

$$\begin{aligned}\lambda &= 2\sum_{i=1}^{\mu}\mathbb{E}\left[\left(\mathcal{N}^{i:\lambda}\right)^2\right] + \sum_{i=\mu+1}^{\lambda-\mu}\mathbb{E}\left[\left(\mathcal{N}^{i:\lambda}\right)^2\right] \\ &< 2\sum_{i=1}^{\mu}\mathbb{E}\left[\left(\mathcal{N}^{i:\lambda}\right)^2\right] + \frac{\lambda-2\mu}{\mu}\sum_{i=1}^{\mu}\mathbb{E}\left[\left(\mathcal{N}^{i:\lambda}\right)^2\right] \\ &= \frac{\lambda}{\mu}\sum_{i=1}^{\mu}\mathbb{E}\left[\left(\mathcal{N}^{i:\lambda}\right)^2\right].\end{aligned}$$

Finally it follows that

$$\frac{1}{\mu}\sum_{i=1}^{\mu}\mathbb{E}\left[\left(\mathcal{N}^{i:\lambda}\right)^2\right] > 1. \tag{4.48}$$

Part 3.2. Now we fall back to the general assumption where $w_1 \geq \cdots \geq w_\mu$. Let us prove beforehand that:

$$\mathbb{E}\left[\left(\mathcal{N}^{1:\lambda}\right)^2\right] \geq \mathbb{E}\left[\left(\mathcal{N}^{2:\lambda}\right)^2\right] \geq \cdots \geq \mathbb{E}\left[\left(\mathcal{N}^{\mu:\lambda}\right)^2\right]. \tag{4.49}$$

Let $i \in \{1, \ldots, \mu-1\}$. We have that $\mathcal{N}^{i:\lambda} \leq \mathcal{N}^{i+1:\lambda} \leq \mathcal{N}^{\lambda+1-i}$. Then if $\mathcal{N}^{i+1:\lambda} \geq 0$, $\left(\mathcal{N}^{i+1:\lambda}\right)^2 \leq \left(\mathcal{N}^{\lambda+1-i:\lambda}\right)^2$ and if $\mathcal{N}^{i+1:\lambda} \leq 0$, $\left(\mathcal{N}^{i+1:\lambda}\right)^2 \leq \left(\mathcal{N}^{i:\lambda}\right)^2$. Since $\left(\mathcal{N}^{\lambda+1-i:\lambda}\right)^2$ and $\left(\mathcal{N}^{i:\lambda}\right)^2$ have the same distribution, it follows that $\left(\mathcal{N}^{i+1:\lambda}\right)^2 \leq \left(\mathcal{N}^{i:\lambda}\right)^2$. Therefore (4.49) holds.

To prove the general case, we use the Chebyshev's sum inequality which states that if $a_1 \geq a_2 \geq \cdots \geq a_\mu$ and $b_1 \geq b_2 \geq \cdots \geq b_\mu$, then

$$\frac{1}{\mu}\sum_{k=1}^{\mu} a_k b_k \geq \left(\frac{1}{\mu}\sum_{k=1}^{\mu} a_k\right)\left(\frac{1}{\mu}\sum_{k=1}^{\mu} b_k\right). \tag{4.50}$$

Therefore we apply the Chebyshev's sum inequality on $w_1 \geq \cdots \geq w_\mu$ and $\mathbb{E}\left[\left(\mathcal{N}^{1:\lambda}\right)^2\right] \geq \mathbb{E}\left[\left(\mathcal{N}^{2:\lambda}\right)^2\right] \geq \cdots \geq \mathbb{E}\left[\left(\mathcal{N}^{\mu:\lambda}\right)^2\right]$. It follows that

$$\frac{1}{\mu}\sum_{i=1}^{\mu} w_i \mathbb{E}\left[\left(\mathcal{N}^{i:\lambda}\right)^2\right] \geq \left(\frac{1}{\mu}\sum_{j=1}^{\mu} w_j\right)\left(\frac{1}{\mu}\sum_{i=1}^{\mu}\mathbb{E}\left[\left(\mathcal{N}^{i:\lambda}\right)^2\right]\right).$$

Therefore, $\sum_{i=1}^{\mu} \frac{w_i}{\sum_{j=1}^{\mu} w_j}\mathbb{E}\left[\left(\mathcal{N}^{i:\lambda}\right)^2\right] \geq \frac{1}{\mu}\sum_{i=1}^{\mu}\mathbb{E}\left[\left(\mathcal{N}^{i:\lambda}\right)^2\right]$. And the first case in (4.48) ensures that $\sum_{i=1}^{\mu} \frac{w_i}{\sum_{j=1}^{\mu} w_j}\mathbb{E}\left[\left(\mathcal{N}^{i:\lambda}\right)^2\right] \geq \frac{1}{\mu}\sum_{i=1}^{\mu}\mathbb{E}\left[\left(\mathcal{N}^{i:\lambda}\right)^2\right] > 1$.

4.7 Relation to previous works

We discuss in this section the positioning of our work with respect to previous studies establishing the convergence of Evolution Strategies.

Most theoretical analyses of linear convergence concern the so-called (1+1)-ES where a single candidate solution is sampled ($\lambda = 1$). The new mean is the best among the current mean and the sampled solution and in addition the one-fifth success rule is used to adapt the step-size [121, 159]. Jägersküpper established lower-bounds and upper-bounds related to linear convergence on spherical functions [106, 110] and on some convex-quadratic functions [107, 108]. The underlying methodology used for the proof is to a great extent hidden within the algorithm analysis but was later unveiled as connected to drift analysis where an overall Lyapunov function of the state of the algorithm (mean and step-size) is found [2]. This Lyapunov function is shown to satisfy drift conditions from which upper and lower bounds of the hitting time of an epsilon neighborhood of the optimum can be derived. This analysis technique was recently used to provide a simple analysis of the hitting time pertaining to linear convergence of the (1+1)-ES with one-fifth success rule on spherical functions [2]. It was generalized for classes of functions including strongly convex functions with Lipschitz gradient as well as positively homogeneous functions [3, 143].

In [24], the linear convergence of a $(1+1)$-ES is proven on increasing transformations of C^1 positively homogeneous functions p with a unique global argmin and upper bounds on the degree of p and on the norm of the gradient $\|\nabla p\|$. The methodology in [24] is similar to ours, as it consists in applying a LLN to ergodic Markov chains.

A few studies attempt to analyze ES with a covariance matrix adaptation: Diouane et al. [60] prove the convergence (but not linear convergence) of a variant of CMA-ES where the algorithm is modified to ensure a sufficient decrease and the convergence proof relies on this modification; an abstract covariance adaptation is included in the linear convergence analysis in [3] provided the eigenvalues stay upper bounded and bounded away from zero (hence the affine-invariant update of the original algorithm is not included). In addition, Akimoto et al. prove that when convergence occurs on a twice continuously differentiable function for CMA-ES without step-size adaptation, the limit point is a local (or global) optimum [8].

On the condition of step-size increase on linear functions Our main condition for the linear behavior proven in Theorem 5 is that "the logarithm of the step-size increases on linear functions", formally, stated as $\mathbb{E}\left[\log\left(\Gamma^\star_{\text{linear}}\right)\right] > 0$ where $\Gamma^\star_{\text{linear}}$ is the step-size change on nontrivial linear functions. This condition is equivalent to the geometric divergence of the step-size on nontrivial linear functions, as shown by the next lemma.

Lemma 14. *Let f be an increasing transformation of a nontrivial linear function, i.e.*

satisfy F2. Let $\{(X_k, \sigma_k)\,; k \in \mathbb{N}\}$ *be the sequence defined in* (4.6) *and* (4.7)*. Assume that* $\{U_{k+1}\,; k \in \mathbb{N}\}$ *satisfies Assumption A5 and that* Γ *satisfies Assumptions A2 and A4,* $i.e.$ Γ *is invariant under rotation and* $\log \circ \Gamma$ *is* $\mathcal{N}_{n\mu}$*-integrable. Then* $\lim_{k\to\infty} \frac{1}{k} \log \frac{\sigma_k}{\sigma_0} = \mathbb{E}\left[\log\left(\Gamma^\star_{\text{linear}}\right)\right]$.

Proof. We have $\frac{1}{k} \log \frac{\sigma_k}{\sigma_0} = \frac{1}{k} \sum_{t=0}^{k-1} \log \frac{\sigma_{t+1}}{\sigma_t}$. With (4.16) and Proposition 17, $\sigma_{t+1} = \sigma_t \Gamma\left(\alpha_{l^\star}(0, U_{t+1})\right)$ where $l^\star$ is the linear function defined as $l^\star(x) = x_1$ for $x \in \mathbb{R}^n$. Therefore $\frac{1}{k} \log \frac{\sigma_k}{\sigma_0} = \frac{1}{k} \sum_{t=0}^{k-1} \left(\log \circ \Gamma \circ \alpha_{l^\star}\right)(0, U_{t+1})$. Using Assumption A3 and Lemma 12, we have that the function $u \mapsto \left(\log \circ \Gamma \circ \alpha_{l^\star}\right)(0, u)$ is $\mathcal{N}_{n\lambda}$-integrable. Then by the LLN applied to the i.i.d. sequence $\{U_{k+1}\,; k \in \mathbb{N}\}$, $\frac{1}{k} \log \frac{\sigma_k}{\sigma_0}$ converges almost surely to $\mathbb{E}\left[\log\left(\Gamma^\star_{\text{linear}}\right)\right]$. □

We find in the literature a different condition for the $(1+1)$-ES [2,24] and the $(1, \lambda)$ self-adaptive ES [19], that is "the step-size increases on linear functions". That condition is formally stated as the existence of $\beta > 0$ such that $\mathbb{E}\left[\frac{1}{{\Gamma^\star_{\text{linear}}}^\beta}\right] < 1$. With the concavity of the logarithm, we have thanks to Jensen's inequality that $\log\left(\mathbb{E}\left[\frac{1}{{\Gamma^\star_{\text{linear}}}^\beta}\right]\right) \geq \mathbb{E}\left[\log\left(\frac{1}{{\Gamma^\star_{\text{linear}}}^\beta}\right)\right] = -\beta\, \mathbb{E}\left[\log\left(\Gamma^\star_{\text{linear}}\right)\right]$. Therefore if $\mathbb{E}\left[\frac{1}{{\Gamma^\star_{\text{linear}}}^\beta}\right] < 1$, then $\mathbb{E}\left[\log\left(\Gamma^\star_{\text{linear}}\right)\right] > 0$. Thereby our condition "the logarithm of the step-size increases on linear functions" is tighter than the traditional condition "the step-size increases on linear functions".

Previous results on CSA-ES For the $(\mu/\mu_w, \lambda)$-CSA-ES algorithm without cumulation, our main condition in Proposition 19 for the linear behavior is formulated based on λ, μ, the weights w and the order statistics of the standard normal distribution. It reads $\mathbb{E}\left[\left(\sum_{i=1}^{\mu} \frac{w_i}{\|w\|} \mathcal{N}^{i:\lambda}\right)^2\right] > 1$. For $\mu = 1$, this condition is satisfied when $\lambda \geq 3$, thanks to Proposition 19.

In [48], the linear divergence of both the incumbent and the step-size is obtained in a $(1, \lambda)$ scenario without cumulation on linear functions whenever $\lambda \geq 3$, with a divergence rate equal to $\frac{\mathbb{E}\left[(\mathcal{N}^{1:\lambda})^2\right]-1}{2d_\zeta n}$. This result is therefore incorporated in Proposition 19. Note that we have simultaneously linear divergence on strictly increasing transformations of nontrivial linear functions and linear behavior on strictly increasing transformations of C^1 scaling-invariant functions with a unique global argmin.

While our framework does not include cumulation by a path for the step-size update via CSA [95], cumulation is encompassed in [48] and linear divergence of the step-size holds on linear functions for the $(1, \lambda)$-CSA-ES. The key aspect consists in applying a LLN to the cumulation path. Linear divergence is only proven for the step-size as it requires the application of the LLN to a more complex Markov chain to prove it for the mean [48].

4.8 Conclusion and discussion

We have proven the asymptotic linear behavior of step-size adaptive $(\mu/\mu_w, \lambda)$-ESs on composites of strictly increasing functions with continuously differentiable scaling-invariant functions. The step-size update has been modeled as an abstract function of the random input multiplied by the current step-size. Two well-known step-size adaptation mechanisms are included in this model, namely derived from the Exponential Natural Evolution Strategy (xNES) [70] and the Cumulative Step-size Adaptation (CSA) [88] without cumulation.

Our methodology leans on investigating the stability of the σ-normalized homogeneous Markov chain to be able to apply a LLN and obtain the limit of the log-distance to the optimum divided by the iteration index. Then we obtain an exact expression of the rate of convergence or divergence as an expectation with respect to the stationary distribution of the σ-normalized chain. This is an elegant feature of our analysis. Other approaches (see previous section) provide bounds on the convergence rate but not its exact expression. Bounds are often expressed depending on dimension or population size which are relevant parameters in practice.

The class of scaling-invariant functions is, as far as we can see, the largest class to which our methodology can conceivably be applied—because on any wider class of functions, a selection function for the σ-normalized Markov chain can not anymore reflect the selection operation in the underlying chain. We require additionally that the objective function is a strictly increasing transformation of either a continuously differentiable function with a unique global argmin or a nontrivial linear function. Many non-convex functions with non-convex sublevel sets are included.

The implied requirement of smooth level sets is instrumental for our analysis. We believe that there exist unimodal functions with non-smooth level sets on which scale invariant ESs can not converge to the global optimum with probability one, for example $x \mapsto \sum_{i=1}^{n} \sqrt{|x_i|}$. However, we also believe that smooth level sets are not a necessary condition for convergence—we consistently observe convergence on $x \mapsto \sum_{i=1}^{n} |x_i|$ for smaller values of n and understand the reason why ESs succeed on the one-norm but fail on the $\frac{1}{2}$-norm function. Capturing this distinction in a rigorous analysis of the Markov chain remains an open challenge.

In contrast, the approach used in [3] allows to handle functions that are not scaling-invariant. This approach requires a drift condition to hold on the whole state-space while our methodology requires that the drift condition only holds outside of a small set which means here when the step-size is much smaller than the distance to the optimum. Hence in our approach, it suffices to control the behavior in the limit when the step-size normalized by the distance to the optimum approaches zero.

A major limitation of our current analysis is the omission of cumulation that is used in the $(\mu/\mu_w, \lambda)$-CSA-ES to adapt the step-size (we have set the cumulation param-

eter to 1, see Section 4.2.2). In case of a parent population of size $\mu = 1$, Chotard et al. obtain linear divergence of the step-size on linear functions also with cumulation [48]. However, no proof of linear behavior exists, to our knowledge, on functions whose level sets are not affine subspaces. While we consider cumulation a crucial component in practice, proving the drift condition for the stability of the Markov chain is much harder when the state space is extended with the cumulative evolution path and this remains an open challenge.

Technically, our results rely on proving φ-irreducibility, positivity and Harris-recurrence of the σ-normalized Markov chain. The φ-irreducibility is difficult to prove directly for the class of algorithms studied in this paper while it is relatively easy to prove for the $(1, \lambda)$-ES with self-adaptation [19] or for the (1+1)-ES with one-fifth success rule [24]. With the tools developed in [47], proving φ-irreducibility, aperiodicity and a T-chain property is much easier, illustrating how useful the connection between stability of Markov chains with stability of deterministic control models can be. Positivity and Harris-recurrence are proven using Foster-Lyapunov drift conditions [140]. We prove a drift condition for geometric ergodicity that implies positivity and Harris-recurrence. It relies on the convergence in distribution of the step-size change towards the step-size change on a linear function when $Z_k = z$ goes to infinity. We also prove in Lemma 10 the existence of non-negligible sets with respect to the invariant probability measure π, outside of a neighborhood of a steadily attracting state. This is used in Proposition 28 to obtain the π-integrability of the function $z \mapsto \log \|z\|$.

We have developed generic results to facilitate further studies of similar Markov chains. More specifically, applying a LLN to the σ-normalized chain is not enough to conclude linear convergence. We introduce the technique to apply the generalized LLN to an abstract chain $\{(Z_k, U_{k+2}) ; k \in \mathbb{N}\}$ and prove that stability properties from $\{Z_k ; k \geq 0\}$ are transferred to $\{(Z_k, U_{k+2}) ; k \in \mathbb{N}\}$.

Chapter 5

Background on multiobjective evolutionary algorithms

Contents

Multiobjective optimization consists of optimizing a vector-valued function $\mathbf{f} : x \in \mathbb{R}^n \longmapsto \mathbf{f}(x) = (\mathbf{f}_1(x), \ldots, \mathbf{f}_m(x)) \in \mathbb{R}^m$ where $m \geq 2$. The Pareto notions are used to define a partial order in $\mathbb{R}^m$. The optimal solutions of a multiobjective optimization problem $\mathbf{f}$ form a set called Pareto set, and their images with respect to $\mathbf{f}$ are called Pareto front.

The goal of a multiobjective optimization problem is to find the Pareto set and the Pareto front. Typically the Pareto set is an infinite set and in practice,

a multiobjective optimization algorithm aims to find a subset of the Pareto set as interesting as possible, $i.e.$ with search points that have images as distinct as possible. Several indicators exist to quantify this quality of a subset. Typically they introduce another order (in addition to the partial order of the Pareto notions) so that the overall order on $\mathbb{R}^m$ is total, and thus all the points are comparable.

We present in this chapter basic notions of multiobjective optimization, that are Pareto notions and quality indicator methods. We also highlight how these notions are used to design well-known multiobjective evolutionary algorithms.

5.1 Basic notions of multiobjective optimization

In what follows, we assume without loss of generality the minimization of a vector-valued function

$$\mathbf{f} : x \in \mathbb{R}^n \longmapsto \mathbf{f}(x) = (\mathbf{f}_1(x), \ldots, \mathbf{f}_m(x)) \in \mathbb{R}^m \tag{5.1}$$

that maps a search point from the search space $\mathbb{R}^n$ to the objective space $\mathbf{f}(\mathbb{R}^n) \subseteq \mathbb{R}^m$.

5.1.1 Pareto notions

The minimization of $\mathbf{f}$ is generally formalized in terms of the weak Pareto dominance relation. A search point $x \in \mathbb{R}^n$ weakly Pareto-dominates (or weakly dominates) another search point $y \in \mathbb{R}^n$ (written in short as $x \preceq y$ or as $\mathbf{f}(x) \preceq \mathbf{f}(y)$) if and only if $\mathbf{f}_i(x) \leq \mathbf{f}_i(y)$ for all $i \in \{1, \ldots, m\}$. Note also that we can naturally extend the (weak) Pareto dominance relation to subsets $A, B \subset \mathbb{R}^n$ as $A \preceq B$ if and only if for all $b \in B$, there exists $a \in A$ such that $a \preceq b$. If the relation $\leq$ is strict for at least one objective function, we say that x Pareto-dominates y (and write $x \prec y$). The set of non-dominated search points constitutes the so-called Pareto set, its image under $\mathbf{f}$ is called the Pareto front.

Assume that P is a finite set of $\mathbb{R}^n$. For an element s of P, we say that s is non-dominated if s is not Pareto dominated by any element of $P \setminus \{s\}$. The set of non-dominated points of P is denoted by $\text{Non-dom}(P)$ and has by definition a Pareto rank equal to 1. The set $\text{Non-dom}(P)$ is refer to as a Pareto approximation set [206] and $\{\mathbf{f}(s), s \in \text{Non-dom}(P)\}$ as the Pareto approximation front, with respect to P.

We define the sequence $\{\mathcal{F}_i\}_{i=1,2,\ldots}$ as follows

$$\mathcal{F}_i = \text{Non-dom}\left(P \setminus \left\{\bigcup_{j=1}^{i-1} \mathcal{F}_j\right\}\right). \tag{5.2}$$

Note beforehand that $\mathcal{F}_1$ is equal to $\text{Non-dom}(P)$ and that the sequence is stationary with a limit equal to the empty set. Denote by $l = \max_i \{\mathcal{F}_i \neq \varnothing\}$. We say that $(\mathcal{F}_1, \ldots, \mathcal{F}_l)$ are the different layers of P and for each $i = 1, \ldots, l$, the Pareto rank of $\mathcal{F}_i$ is by definition equal to i.

Equation (5.2) gives the different layers of the population P in $O(m|\mathbf{f}(P)|^3)$ computations. Indeed, finding the non-dominated points of P based on (5.2) requires each element of P to be compared with each other element of P to see whether it is dominated. This operation costs $O(m|\mathbf{f}(P)|^2)$. And to find the next layer, all non-dominated points are removed and the same procedure is repeated. In the worse case, the second layer (with Pareto rank 2) also requires a complexity of $O(m|\mathbf{f}(P)|^2)$, which happens when $O(|\mathbf{f}(P)|)$ number of solutions belong to the second layer. The same argument holds for the other layers, namely when there is one element per layer. This requires $O(m|\mathbf{f}(P)|^3)$ computations.

In [56], a fast non-dominated sorting method has been proposed, allowing to reduce the complexity of finders the different layers of P until $O(m|\mathbf{f}(P)|^2)$. The corresponding algorithm achieving that complexity is presented in Algorithm 4. First, two entities are always computed for each solution p : the domination count n_p which is the number of solutions dominating p, S_p which is the set of solutions that p dominates. This operation cost $O(m|\mathbf{f}(P)|^2)$. Therefore the solutions for which their domination count is equal to 0 represents the first non-dominated front (the first layer). Next, for each solution p with $n_p = 0$, each element q of the set S_p is visited and has its domination count decremented by 1. The q for which n_q becomes equal to 0 are put in a separate list Q, which constitutes the second non-dominated front (second layer). The procedure continues until all layers are known.

For each solution p, $n_p \leq |\mathbf{f}(P)|-1$ (with equality if each layer has exactly one point), therefore p is visited at most $|\mathbf{f}(P)|-1$ times before n_p decremented to 0. And whenever $n_p = 0$, the layer of p is detected and p is then removed from the population. The cost of that operation is $O(|\mathbf{f}(P)|^2)$ since there are at most $|\mathbf{f}(P)|-1$ such solutions. Therefore the overall time complexity is $O(m|\mathbf{f}(P)|^2)$.

5.1.2 Sorting among a set of non-dominated points

As reminded in the introduction of this chapter, the weak Pareto-dominance is a partial order in $\mathbb{R}^m$ that we denote by $\preceq$. It is typically not total, $i.e.$ the connex property (saying that any pair can be compared with respect to $\preceq$) is not satisfied. To repair this defect, many workarounds have been proposed throughout the years, that consist of finding a well-suited metric able to compare the non-dominated points of a set. A standard technique consists of deriving a metric from a generic indicator called a quality indicator.

Algorithm 4 fast-non-dominated-sort(P)

Given: a population P, an empty set $\mathcal{F}_1 = \varnothing$
for each $p \in P$ **do**
    $S_p = \varnothing$
    $n_p = 0$
    for each $q \in P$ **do**
        if $p \prec q$ **then** ▷ if p dominates q
            $S_p = S_p \bigcup \{q\}$ ▷ add q to the set of solutions dominated by p
        else if $q \prec p$ **then**
            $n_p = n_p + 1$ ▷ increment the domination counter of p
        end if
    end for
    if $n_p = 0$ **then** ▷ p belongs to the first front (is of Pareto rank 1)
        $p_{\text{rank}} = 1$
        $\mathcal{F}_1 = \mathcal{F}_1 \bigcup \{p\}$
    end if
end for
$i = 1$ ▷ initialize the front counter
while $\mathcal{F}_i \neq \varnothing$ **do**
    $Q = \varnothing$ ▷ for the storage of the next front's members
    for each $p \in \mathcal{F}_i$ **do**
        for each $q \in S_p$ **do**
            $n_q = n_q - 1$
            if $n_q = 0$ **then** ▷ q belongs to the next front
                $q_{rank} = i + 1$
                $Q = Q \bigcup \{q\}$
            end if
        end for
    end for
    $i = i + 1$
    $\mathcal{F}_i = Q$
end while

Let us denote by Ω^m the set of all Pareto set approximations in $\mathbb{R}^n$, $i.e.$ the set of all sets containing only non-dominated points. A quality indicator $\mathcal{I}$ is defined as [206][1]

[1]Note that in [206], the set of departure is defined as the set of all Pareto front approximations, while we opt here for the set of all Pareto set approximations. We can take $\mathcal{I} \circ \mathbf{f}$ from our definition to get the same formalism than [206].

$$\mathcal{I} : \Omega^m \longrightarrow \mathbb{R} \tag{5.3}$$

A quality indicator is called monotone [203] if $A \preceq B \implies \mathcal{I}(A) \geq \mathcal{I}(B)$, *i.e.* if it does not contradict the weak Pareto dominance relation. If $A \prec B \implies \mathcal{I}(A) > \mathcal{I}(B)$, then $\mathcal{I}$ is said to be strictly monotone.

For a monotone quality indicator $\mathcal{I}$, if S is a Pareto set approximation and $\mathbf{s} \in S$, then $S \setminus \{\mathbf{s}\}$ is still a Pareto set approximation and $\mathcal{I}(S) \geq \mathcal{I}(S \setminus \{\mathbf{s}\})$. We then denote by $\Delta\mathcal{I}$ the indicator contribution, defined for a Pareto set approximation S and an element $\mathbf{s} \in S$ as:

$$\Delta\mathcal{I}(\mathbf{s}, S) = \mathcal{I}(S) - \mathcal{I}(S \setminus \{\mathbf{s}\}) \ . \tag{5.4}$$

It represents the contribution in $\mathcal{I}(S)$ that we lose if we remove $\mathbf{s}$ from S.

The indicator improvement is a variant of the indicator contribution. It is denoted by $\widetilde{\Delta}\mathcal{I}$ is defined for a Pareto set approximation S and an element $\mathbf{s} \in S$ as:

$$\widetilde{\Delta}\mathcal{I}(\mathbf{s}, S) = \mathcal{I}(S \cup \{\mathbf{s}\}) - \mathcal{I}(S) \ . \tag{5.5}$$

It represents the indicator value that we gain when we add s to the set S.

Well-known metrics among the performance indicators are the hypervolume contribution [30, 104], the hypervolume improvement [63, 198] and the crowding distance [56, 104]. We give a succinct reminder of these methods in the following.

5.1.3 Hypervolume-based performance indicators

Natural candidates for practically relevant quality indicators are monotone indicators such as the epsilon-indicator [204], the R2 indicator [78] and strictly monotone indicators such as the hypervolume indicator, that is (with its variants) the unique known strictly monotone indicator to date.

The hypervolume $HV_{\mathbf{r}}$ [205] of a finite set of solutions $S \subset \mathbb{R}^n$ with respect to the reference point $\mathbf{r} \in \mathbb{R}^m$ is defined as $HV_{\mathbf{r}}(S) = \lambda_m\left(\{z \in \mathbb{R}^m ; \exists y \in S, \mathbf{f}(y) \prec z \prec \mathbf{r}\}\right)$, where λ_m is the Lebesgue measure on the objective space $\mathbb{R}^m$ and $\mathbf{f}$ is the objective function.

The hypervolume indicator allows to formulate a multiobjective optimization problem $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ as a single-objective optimization problem where $(x_1, \ldots, x_p) \mapsto HV_{\mathbf{r}}\left(\{x_1, \ldots, x_p\}\right)$ where p is fixed. Thanks to the strictly monotone quality, an argmax of this single-objective problem in $\mathbb{R}^{np}$ is a tuple of p elements of the Pareto set of $\mathbf{f}$. It is the so-called optimal p-distribution of the Pareto set [16, 22].

Two perfomance indicators are derived from the hypervolume indicator: the hypervolume contribution and the hypervolume improvement. The hypervolume contribution $\mathsf{HVC}_{\mathbf{r}}(\mathbf{s}, S)$ of a search point $\mathbf{s} \in \mathbb{R}^n$ to a set $S \subseteq \mathbb{R}^n$ with respect to the reference point $\mathbf{r} \in \mathbb{R}^m$ is defined as in (5.4). The hypervolume improvement [63,198] of $\mathbf{s} \in \mathbb{R}^n$ to a set $S \subseteq \mathbb{R}^n$ with respect to the reference point $\mathbf{r} \in \mathbb{R}^m$ is defined in (5.5) and denoted by $\mathsf{HVI}_{\mathbf{r}}(\mathbf{s}, S)$.

Thanks to its unique strictly-monotonic property among the quality indicators, well-known modern multiobjective evolutionary algorithms consider the hypervolume indicator (along with the hypervolume contribution) as a way to satisfy the second-order ranking (after the Pareto ranking), during the selection phase of the algorithm [30, 104]. In Figure 5.1, the hypervolume contributions with respect to a reference point $\mathbf{r}$ are shown in the hatched boxes.

Along with the partial order $\preceq$, we can construct total orders $\preceq_{\mathsf{HVC}_{\mathbf{r}}}$ and $\preceq_{\mathsf{HVI}_{\mathbf{r}}}$ in $\mathbb{R}^n$ as follows:

- If $\mathbf{s}_1 \preceq \mathbf{s}_2$ then $\mathbf{s}_1 \preceq_{\mathsf{HVC}_{\mathbf{r}}} \mathbf{s}_2$ and $\mathbf{s}_1 \preceq_{\mathsf{HVI}_{\mathbf{r}}} \mathbf{s}_2$.
- For a Pareto set approximation (non-dominated points) S, if $(\mathbf{s}_1, \mathbf{s}_2) \in S^2$, then

$$\mathbf{s}_1 \preceq_{\mathsf{HVC}_{\mathbf{r}}} \mathbf{s}_2 \iff \mathsf{HVC}_{\mathbf{r}}(\mathbf{s}_1, S) \geq \mathsf{HVC}_{\mathbf{r}}(\mathbf{s}_2, S). \tag{5.6}$$

$$\mathbf{s}_1 \preceq_{\mathsf{HVI}_{\mathbf{r}}} \mathbf{s}_2 \iff \mathsf{HVI}_{\mathbf{r}}(\mathbf{s}_1, S) \geq \mathsf{HVI}_{\mathbf{r}}(\mathbf{s}_2, S). \tag{5.7}$$

That settles the connex property that is missing with the weak Pareto dominance relation $\preceq$.

5.1.4 The crowding distance

The crowding distance defined in Algorithm 5 is a performance indicator that gives an estimate of the density of solutions in the neighbor of a specific point in a given population. For each $j = 1, \ldots, k$, $\mathbf{f}_j(S)$ is sorted so that the distance of the two neighbors of $\mathbf{f}_j(\mathbf{s})$ is known (by convention equal to ∞ if $\mathbf{s}$ is an extreme point and then may not have two neighbors with respect to the objective function $\mathbf{f}_j$). In Figure 5.2, the distance of the neighbors of i is determined in the objective functions $\mathbf{f}_1$ and $\mathbf{f}_2$. Afterwards, the crowding distance is computed as the sum of the above distance values (normalized in each objective j by $\mathbf{f}_j^{max} - \mathbf{f}_j^{min}$ where $\mathbf{f}_j^{max}$ represents the maximum of $\mathbf{f}_j$ and $\mathbf{f}_j^{min}$ represents the minimum of $\mathbf{f}_j$) corresponding to each objective function. We denote the crowding distance of the point $\mathbf{s} \in S$ with respect to the Pareto set approximation S as $c(\mathbf{s}, S)$. It is uses in [56] as a second-order ranking of the selection method.

The computation of the crowding distances of all elements of S requires then to sort $\mathbf{f}(S)$ with respect to each objective function, which is of time complexity $O(m|S|\log(|S|))$.

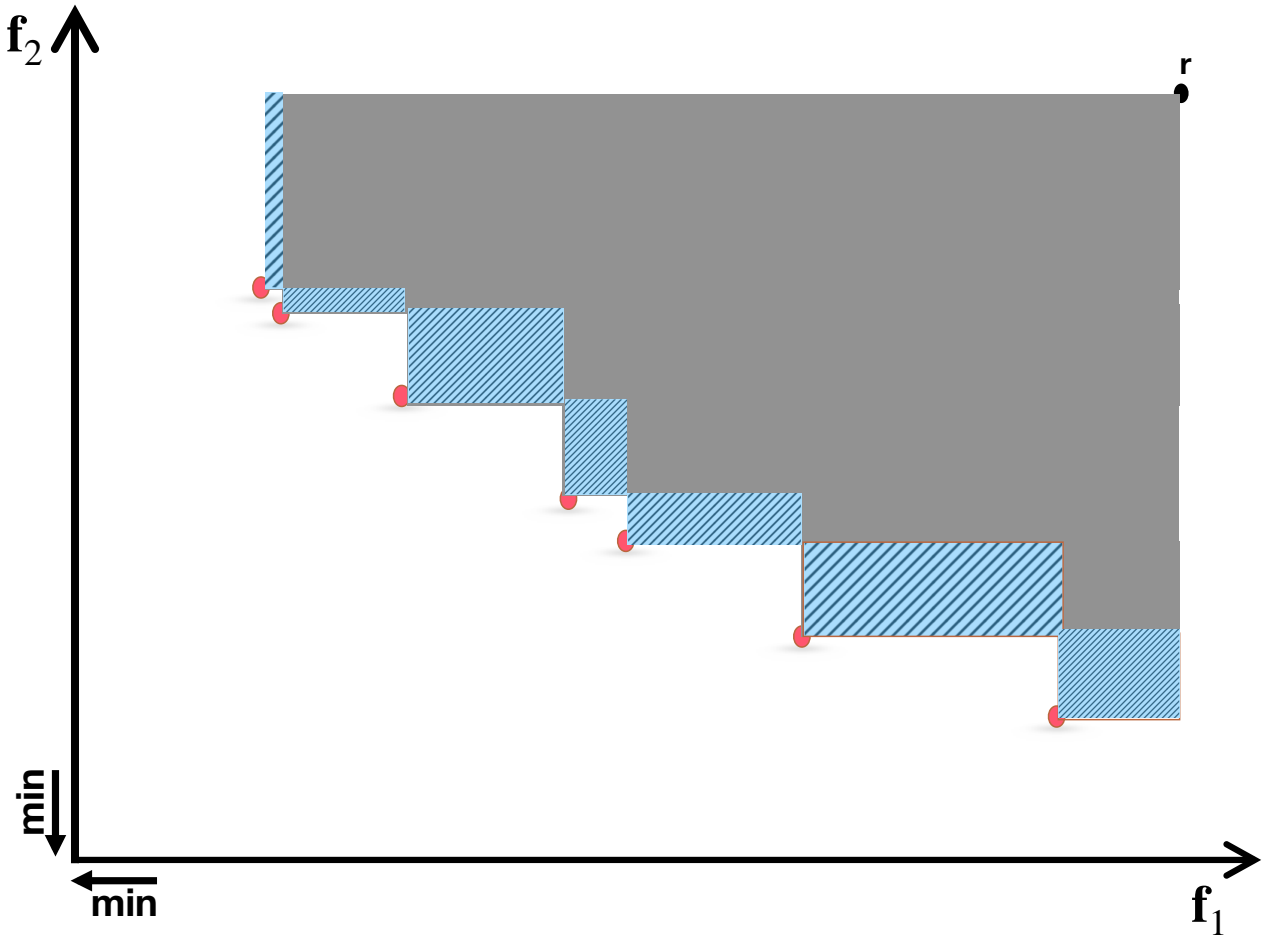


Figure 5.1: Hypervolume contributions of non-dominated points with respect to a reference point $\mathbf{r}$. A filled box (cuboid) with a hatch represents the hypervolume contributed by a non-dominated point.

The new total order called $\preceq_n$ in $\mathbb{R}^m$ [56] that derives from the crowding distance and the partial order $\preceq$ is defined as follows:

- If $\mathbf{s}_1 \preceq \mathbf{s}_2$ then $\mathbf{s}_1 \preceq_n \mathbf{s}_2$.
- For a Pareto set approximation (non-dominated points) S, if $(\mathbf{s}_1, \mathbf{s}_2) \in S^2$, then

$$\mathbf{s}_1 \preceq_n \mathbf{s}_2 \iff c(\mathbf{s}_1, S) \geq c(\mathbf{s}_2, S).$$

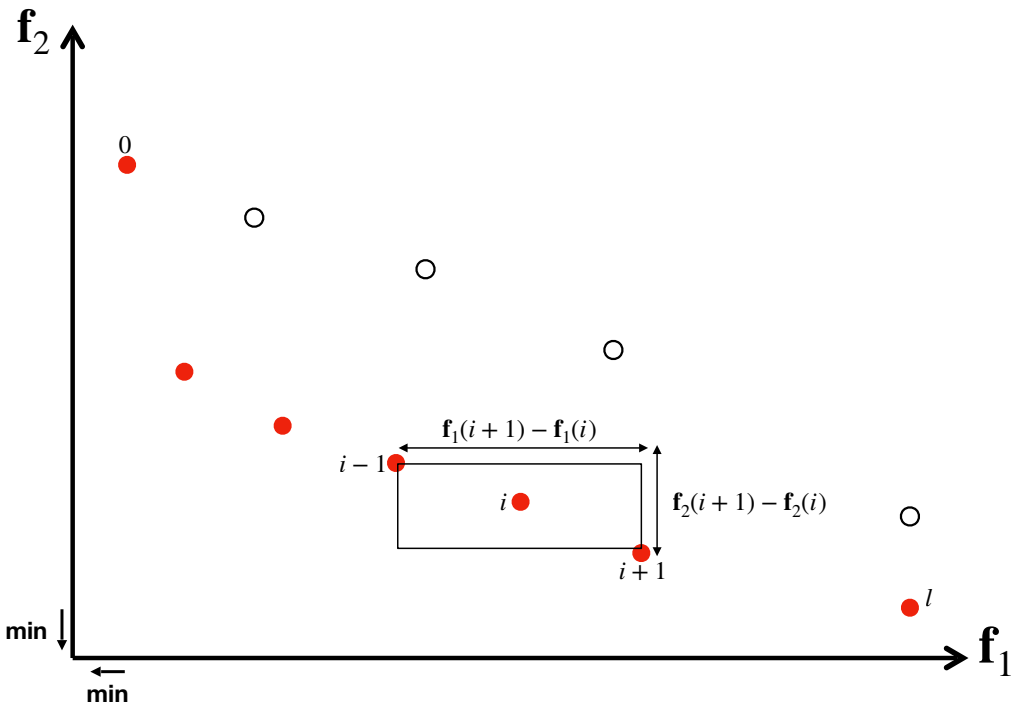


Figure 5.2: Computation of the crowding-distance of point i. Filled circles are points of Pareto rank 1, no filled circles are points of Pareto rank 2.

5.2 Multiobjective evolutionary algorithms with non-dominated sorting methods

A multiobjective evolutionary algorithm is similar to its single-objective counterpart in the way that it proceeds. A population is initialized, then evaluations are done with a fitness function, and finally recombination, mutation and selection are operated.

We present in the following non-dominated sorting multiobjective genetic algo-

Algorithm 5 crowding-distance-assignment(S)

Given: a population of non-dominated points $S = \{S[1], \ldots, S[l]\} \subset \mathbb{R}^n$, with $l = |S|$
for $i = 1, \ldots, l$ **do**
    $c(S[i], S) = 0$
end for
for each objective m **do**
    $S = \text{sort}\,(S, m)$ ▷ sort with respect to objective function $\mathbf{f}_m$
    $c(S[1], S) = c(S[l], S) = \infty$ ▷ so that boundary points are always selected
    for i = 2, . . . , l - 1 **do** ▷ for all other points
        $c(S[i], S) = c(S[i], S) + \dfrac{\mathbf{f}_m(S[i+1]) - \mathbf{f}_m(S[i-1])}{\max \mathbf{f}_m - \min \mathbf{f}_m}$
    end for
end for

rithms that embed the popular family of NSGA algorithms. SMS-EMOA and MO-CMA-ES are also briefly presented, as examples of multiobjective algorithms using the non-dominated sorting method. Finally we present some general drawbacks related to the non-dominated sorting methods.

5.2.1 Non-dominated sorting multiobjective genetic algorithms

The first practical multiobjective genetic algorithm is the Vector Evaluated Genetic Algorithm (VEGA) [169]. However, its bias towards some specific Pareto-optimal solutions is too large and then VEGA encounters a major difficulty to produce sufficient diversity among the solutions [178]. That leads to the development of a non-dominated sorting method in [73] to tackle this problem. That technique has been firstly incorporated in [67] and in [102]. However it is in [178], with the creation of the Non-dominated Sorting Genetic Algorithm (NSGA), that it encounters more successes by the practitioners. The crossover and mutation operators remain the same as in single-objective Genetic Algorithms. Only the selection method among the solutions changes. The same observation holds among the variant of NSGA algorithms in [55, 56, 113, 178, 192].

The main principle of the NSGA in [178] is based on a two-way ranking. Firstly, the solutions are ranked with respect to their Pareto rank (this phase is known as a ranking selection method). Lastly, the solutions with the same Pareto rank are sorted with respect to a specific method referred to the niche method.

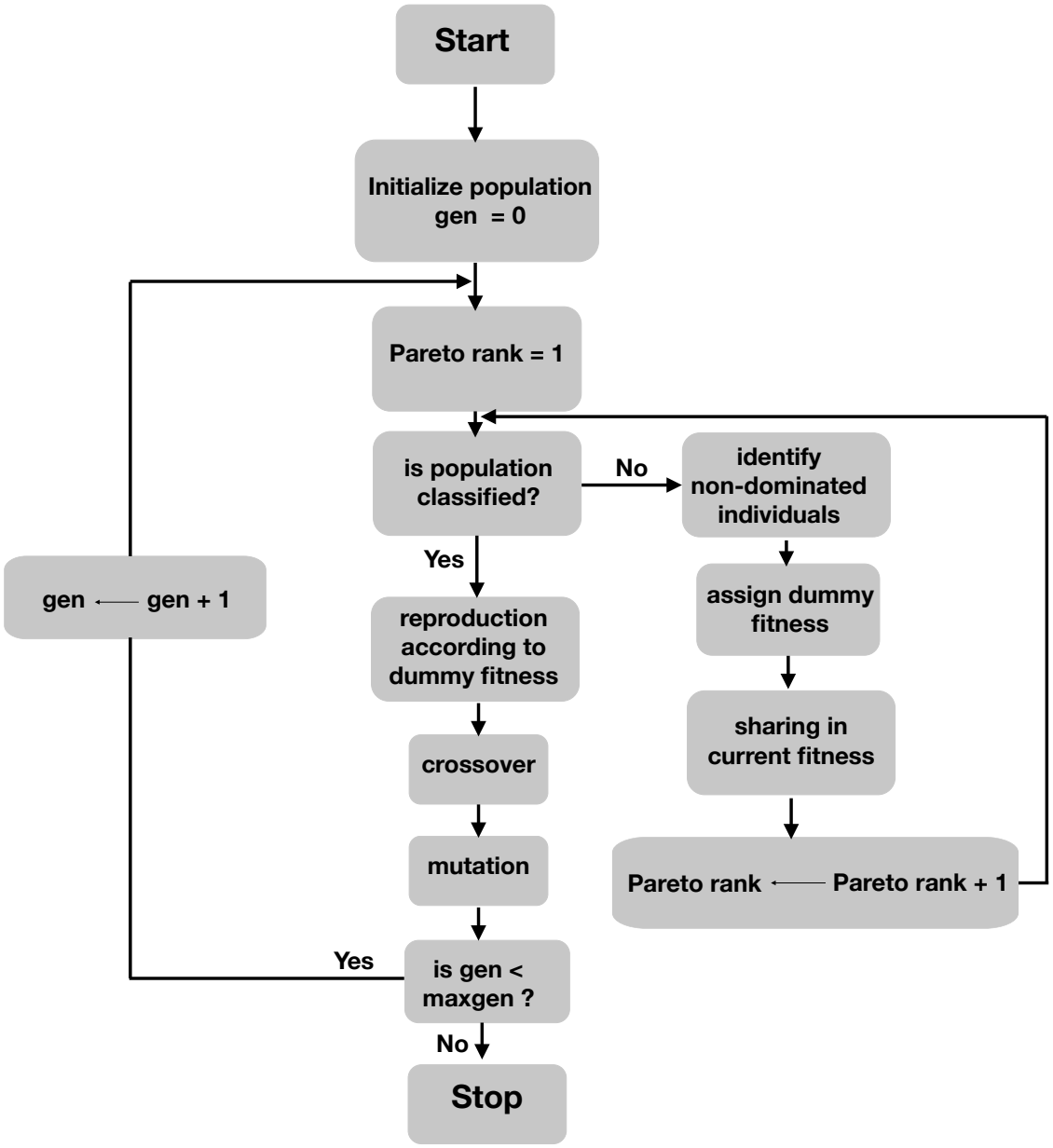


Figure 5.3: Illustration of the NSGA procedure.

Within the solutions of the same Pareto rank, a “dummy” fitness is used to always ensure that the maximum fitness value of a layer is smaller than the minimum fitness value of another layer which has a smaller Pareto rank, so that the solutions with smaller Pareto rank are always selected before solutions with greater Pareto rank. Figure 5.3 illustrates the complete structure of the NSGA algorithm.

However, the heavy-dependence of the NSGA algorithm to the sharing parameter σ_{share} (with little a priori information that guarantee a good parameter setting of σ_{share}) and the expensive time complexity deriving from the sharing function ($O(|P|^2)$ where P is the population) make the use of NSGA less practical. Therefore in [56], NSGA-II is created, where the second-order ranking is replaced with the crowding distance that we reminded in 5.1.4. NSGA-II does not depend heavily on a parameter setting (like NSGA) as it is shown in Algorithm 6, and the crowding distance technique aims to conduct niching by allocating a crowding distance score to each element of a population. The role of the crowding distance is then to keep a diverse front during the convergence of the NSGA-II.

The NSGA-II algorithm is well established and still used in a variety of very recent industries like automatic refactoring [151], automated Machine learning [175]. And in the multiobjective research community, NSGA-II has been used for the design of Multiobjective Bayesian Optimization Algorithm (Multiobjective BOA) developed in [124].

Algorithm 6 NSGA-II algorithm

Given: $t = 0$, the parent population P_0 with $|P_0| = N$, the offspring population Q_0
while not stopping criterion **do**
    $R_t = P_t \bigcup Q_t$ ▷ combine parent and offspring population
    $\mathcal{F} = \text{fast-non-dominated-sort}(R_t)$ ▷ $\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2, \dots)$, all non-dominated fronts of R_t
    $P_{t+1} = \varnothing$ and $i = 1$
    while $|P_{t+1}| + |\mathcal{F}_i| \leq N$ **do** ▷ until the parent population is filled
        $\text{crowding-distance-assignment}(\mathcal{F})$ ▷ compute crowding-distance in $\mathcal{F}_i$
        $P_{t+1} = P_{t+1} \bigcup \mathcal{F}_i$ $i = i + 1$
    end while
    $\text{sort}(\mathcal{F}_i, \prec_n)$ ▷ sort in descending order using $\prec_n$
    $P_{t+1} = P_{t+1} \bigcup \mathcal{F}_i[1 : (N - |P_{t+1}|)]$ ▷ choose the first $(N - |P_{t+1}|)$ elements of $\mathcal{F}_i$
    $Q_{t+1} = \text{make\_new\_pop}(P_{t+1})$ ▷ use selection, crossover and mutation to create a new population Q_{t+1}
    $t = t + 1$ ▷ increment the generation counter
end while

5.2.2 SMS-EMOA

SMS-EMOA [30] is an Evolutionary Multiobjective Optimization Algorithm using the non-dominated sorting method and the hypervolume contribution as the first and sec-

ond order ranking during the selection phase. It stands for $\mathscr{S}$ *metric selection* Evolutionary Multiobjective Optimization Algorithm, where $\mathscr{S}$ *metric* refers to the hypervolume indicator.

SMS-EMOA is inspired by archiving strategies presented in [125, 127], and uses simulated binary crossover (SBX) and a polynomial mutation operator [54]. The optimization algorithm is recalled in Algorithm 7. Given a non-empty population $P \subset \mathbb{R}^n$ of λ individuals, we generate a new element $p \in \mathbb{R}^n$ from P using the above variation operators. Then the subset S of the elements of $P \bigcup \{p\}$ with the largest Pareto rank is found using the non-dominated sorting algorithm presented in Algorithm 4. Later, an element of S minimizing $\mathbf{s} \mapsto \mathrm{HVC}_{\mathbf{r}}(\mathbf{s}, S)$ where $\mathbf{r}$ is a reference point is chosen and removed from $P \bigcup \{p\}$. This process is repeated until a stopping criteria is met.

Algorithm 7 SMS-EMOA

1: **Given:** initialize randomly a population $P \subset \mathbb{R}^n$ of λ individuals
2: **while not** stopping criterion **do**
3: $p \longleftarrow \mathrm{generate}(P)$ ▷ generate an offspring by variation.
4: $(\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_v) = \text{fast-non-dominated-sort}(P \bigcup \{p\})$, $S = \mathcal{F}_v$ ▷ all non-dominated fronts of $P \bigcup \{p\}$.
5: choose a reference point $\mathbf{r}$
6: $q \longleftarrow$ an element of $\arg\min_{\mathbf{s} \in S} \mathrm{HVC}_{\mathbf{r}}(\mathbf{s}, S)$
7: $P \longleftarrow (P \bigcup \{p\}) \setminus \{q\}$ ▷ select λ best individuals.
8: **end while**
9: **return** P

The experiments in [30] show that SMS-EMOA is well-suited for an optimization with two or three objectives, if the number of individuals is fixed and small. It outperforms NSGA-II, with a more diversified Pareto approximation thanks to the hypervolume indicator. SMS-EMOA also focuses the evolutionary search towards less explored regions near the Pareto set.

In [66], the SMS-EMOA algorithm uses the hypervolume contribution as a performance indicator to approximate the optimal distribution of p points on the Pareto set. The corresponding Pareto front is empirically shown to be well-distributed [63, 127].

5.2.3 MO-CMA-ES

The Multiobjective Covariance Matrix Adaptation Evolution Strategy (MO-CMA-ES) is introduced in [104, 193]. It is based on an elitist variant of the true CMA-ES (which is non-elitist). For the selection method, MO-CMA-ES has also a two-way ranking, with the first ranking being the Pareto rank. The performance indicator for the second ranking can be either the crowding distance (c-MO-CMA-ES) or the hypervolume

contribution (s-MO-CMA-ES). Therefore c-MO-CMA-ES differs from NSGA-II and s-MO-CMA-ES from SMS-EMOA mainly due to the variation operators that guide the genetic algorithm or the evolution strategy. With the elitist CMA-ES defined in Section 1.3.2, $(1 + \lambda)$-CMA-ES, λ can be chosen as small as 1.

Based on the $(1 + \lambda)$-CMA-ES introduced in Section 1.3.2, the multiobjective variant is constructed by evolving a population of λ_{MO} individuals, where each individual is a candidate solution for an elitist $(1 + \lambda)$-CMA-ES. The corresponding algorithm is called $\lambda_{\mathrm{MO}} \times (1 + \lambda)$-MO-CMA-ES.

Algorithm 8 depicts the evolution of a $\lambda_{\mathrm{MO}} \times (1 + 1)$-MO-CMA-ES, which can be naturally generalized to a $\lambda_{\mathrm{MO}} \times (1 + \lambda)$-MO-CMA-ES version. First, we consider λ_{MO} $(1 + 1)$-CMA-ES denoted by $es_1, \ldots, es_{\lambda_{\mathrm{MO}}}$, we then make a copy of each es_k for $k = 1, \ldots, \lambda_{\mathrm{MO}}$, and replace the candidate solution of each copy by the offspring of its original's parent. Then all the $2\lambda_{\mathrm{MO}}$ parents are merged in one population Q, that we sort with a two-way ranking order (the Pareto ranking and then either the hypervolume contribution or the crowding distance) denoted by $\prec_Q$. Then, the covariance matrices of the λ_{MO} copied evolution strategies are updated, the same for all the $2\lambda_{\mathrm{MO}}$ step-sizes. For $k = 1, \ldots, \lambda_{\mathrm{MO}}$, the success rate is the same for es_k and its copy, denoted by $\lambda_{succ,Q,k}$ and is equal to 1 if the offspring is smaller than the parent with respect to $\prec_Q$ and 0 otherwise. And finally, the best λ_{MO} candidate solutions among the $2\lambda_{\mathrm{MO}}$ become the new population.

The experiments in [104] show that the MO-CMA-ES variant with the hypervolume contribution, namely s-MO-CMA-ES, generally outperforms the variant with the crowding distance, $i.e.$ c-MO-CMA-ES and the NSGA-II. And for non-separable problems, s-MO-CMA-ES significantly outperforms NSGA-II.

5.2.4 Drawbacks of the non-dominated sorting methods

We have presented multiobjective optimization algorithms based on a two-way ranking, with an elitist selection. This type of algorithms prevails among the multiobjective evolutionary algorithms. Recently in [181], non-elitist evolutionary multi-objective optimizers are shown to experimentally perform better than their elitist counter-parts.

Another observation contests the quality of the points that the non-dominated sorting algorithms presented above aim to obtain. In Figure 5.4, six points are plotted with their corresponding hypervolume improvement's level sets. The left plot is in the search space and the right plot is in the objective space. We observe the total order effect of the hypervolume improvement as the level sets are not flat on same Pareto rank regions, which could not be done by the Pareto ranking. However the dominated point goes towards the direction orthogonal to the level sets. It steers towards a region already occupied by a non-dominated point. Therefore by observing the level

Algorithm 8 $\lambda_{\mathrm{MO}} \times (1+1)$-MO-CMA

Given: $g = 0$, initialize $a_k^{(g)}$ for $k = 1, \ldots, \lambda_{\mathrm{MO}}$
while not stopping criterion **do**
  for $k = 1, \ldots, \lambda_{\mathrm{MO}}$ **do**
    $a_k'^{(g+1)} \longleftarrow a_k^{(g)}$
    $\mathbf{x}_k^{g+1} \sim \mathcal{N}\left(x_{parent}^{(g)}, \sigma^{(g)^2}\ \mathbf{C}^{(g)}\right)$
  end for
  $Q^{(g)} = \left\{a_k'^{(g+1)}, a_k^{(g)}\ ;\ 1 \leq k \leq \lambda_{\mathrm{MO}}\right\}$
  for $k = 1, \ldots, \lambda_{\mathrm{MO}}$ **do**
    $\mathrm{updateStepSize}\left(a_k^{(g)}, \lambda_{succ,Q^{(g)},k}^{(g+1)}\right)$ ▷ defined in Algorithm 1
    $\mathrm{updateStepSize}\left(a_k'^{(g)}, \lambda_{succ,Q^{(g)},k}^{(g+1)}\right)$
    $\mathrm{updateCovariance}\left(a_k'^{(g)}, \frac{x_k'^{(g+1)} - x_k^{(g)}}{\sigma_k^{(g)}}\right)$ ▷ defined in Algorithm 2
  end for
  for $i = 1, \ldots, \lambda_{\mathrm{MO}}$ **do**
    $Q_{<:i}^{(g)} \longleftarrow i$ first elements in $Q^{(g)}$ sorted with respect to the $\prec_{Q^{(g)}}$ order.
    $a_i^{g+1} \longleftarrow Q_{<:i}^{(g)}$
  end for
  $g \longleftarrow g + 1$
end while

sets, it follows that the two-way ranking tends to make the final solutions crowded in some regions.

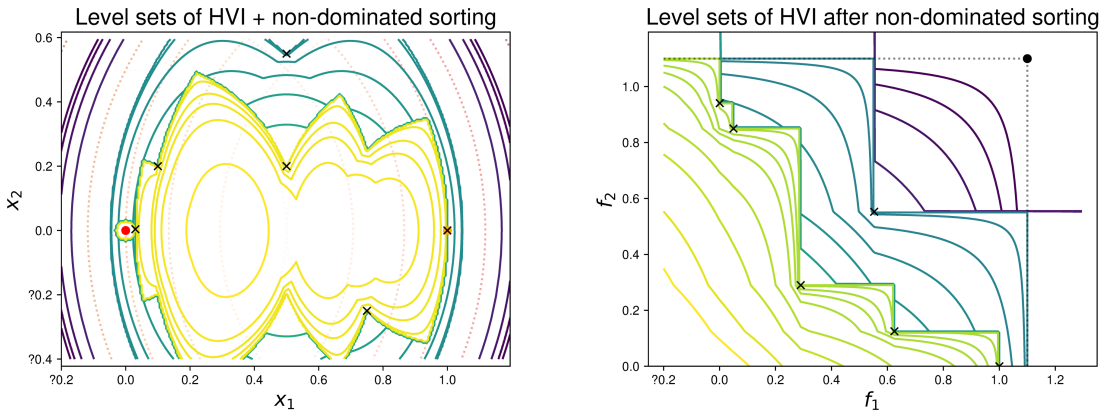


Figure 5.4: Level sets of the two-way ranking fitness selection with the hypervolume contribution for a bi-objective problem with two spheres of optimum respectively at $(0, 0)$ and at $(1, 0)$. Left: search space. Right: objective space. The black dot indicates the reference point of $[1.1, 1.1]$.

Chapter 6

Theoretical analysis of Pareto fronts and Pareto sets of bi-objective convex-quadratic functions

Contents

Note: the content of this chapter is presented in [183]:
Cheikh Toure, Anne Auger, Dimo Brockhoff and Nikolaus Hansen, *On Bi-Objective convex-quadratic problems*, International Conference on Evolutionary Multi-Criterion Optimization, 2019, pp 3–14.

In this chapter we analyze theoretical properties of bi-objective convex-quadratic problems. We give a complete description of their Pareto set and

prove the convexity of their Pareto front. We show that the Pareto set is a line segment when both Hessian matrices are proportional.

We then propose a novel set of convex-quadratic test problems, describe their theoretical properties and the algorithm abilities required by those test problems. This includes in particular testing the sensitivity with respect to separability, ill-conditioned problems, rotational invariance, and whether the Pareto set is aligned with the coordinate axis.

6.1 Introduction

Convex-quadratic functions are among the simplest yet very useful test functions in optimization. Given a positive definite matrix Q of $\mathbb{R}^{n\times n}$, a convex-quadratic function is defined as

$$f(x) = \frac{1}{2}(x - x^*)^\top Q(x - x^*)$$

where x^* is the unique optimum of the function. The Hessian of f coincides with the matrix Q. The level-sets of f defined as $\{x \in \mathbb{R}^n : (x - x^*)^\top Q(x - x^*) = c, c \geq 0\}$ are hyper-ellipsoids whose main axes are the eigenvectors of the matrix Q with length proportional to the inverse of the eigenvalues of Q.

By changing the eigenvalues and eigenvectors of Q, one can model different essential difficulties in numerical optimization: if the eigenvectors are not aligned with the coordinate axes (if the matrix Q is not diagonal), then the associated function is non-separable: it cannot be efficiently optimized by coordinate-wise search. In practice, difficult optimization problems are non-separable. Having a large condition number for Q, that is a large ratio between the largest and smallest eigenvalue of Q models ill-conditioned problems where the characteristic scale along different directions is very different. Ill-conditioning is very frequent in real-world problems. They arise naturally as one often optimizes quantities that have different natures and different intrinsic scales (some variables can be akin to time, others to weights, ...) such that a unit change along each variable can have a completely different impact on the function optimized. More generally, the eigenspectrum of Q entirely characterizes the scale among the different axes of the hyper-ellipsoidal level sets and parametrizes the difficulty of the function: from the arguably easiest function, the sphere function $f(x) = \sum_{i=1}^n x_i^2$, to very difficult ill-conditioned functions where condition numbers of Q of up to 10^{10} have been observed in real-world problems, for example in [50].

Convex-quadratic functions have been central to the design of several important classes of optimization algorithms for single-objective optimization. Newton or quasi-Newton methods use or learn a second order approximation of the objective function optimized [148]. This second order approximation is done by convex-quadratic functions (assuming that the function is twice continuously differentiable and convex). Introduced more recently, the class of derivative-free-optimization (DFO) trust-region

based algorithms builds a second-order approximation of the objective function by interpolation [156]. In the evolutionary computation (EC) context, convex-quadratic functions have also played a central role for the design of algorithms like CMA-ES: they have been intensively used for designing the algorithm and the performance of the method has been carefully quantified on different eigenspectra of the matrix Q for different condition numbers [95].

Given that a multiobjective problem is "simply" the simultaneous optimization of single-objective problems, the typical difficulties of each objective function are the same as the typical difficulties of single-objective problems. In particular non-separability and ill-conditioning are important difficulties that the single functions have. Therefore, combining convex-quadratic problems seems natural for testing and designing multiobjective algorithms. This has already been done in the past for instance for the design of multiobjective versions of CMA-ES [104] or as a subset of the biobjective BBOB test function suite [39, 188].

Yet, while the difficulties encoded and parametrized within a convex-quadratic problem are well-understood for single-objective optimization, the situation is different for multiobjective optimization, starting from bi-objective optimization. Simple properties like convexity of the Pareto front associated to bi-objective convex-quadratic problems as well as properties of the Pareto set have not been systematically investigated. Additionally, convex-quadratic bi-objective test problems used in the literature do not capture all important properties one could be testing with convex-quadratic problems. There is more degree of freedom than for single objective optimization that is not exploited: we can combine two functions having the same Hessian matrix, place the optima on the functions both on one axis of the search space, ... and this will affect how the Pareto set and Pareto front look like.

This paper aims at filling the gaps from the literature on multiobjective optimization with respect to convex-quadratic problems. More precisely the objectives are twofold: clarify theoretical Pareto properties of bi-objective problems where each function is convex-quadratic and define sets of bi-objective convex-quadratic problems that allow to test different (well-understood) difficulties of bi-objective problems. The paper is organized as follows: in subsection 6.2 we present theoretical properties of convex-quadratic problems and discuss new test functions in subsection 6.3.

6.2 Theoretical Properties of Bi-Objective Convex-Quadratic Problems

6.2.1 Preliminaries

We consider bi-objective problems (f_1, f_2) defined on the search space $\mathbb{R}^n$.

The Pareto set of (f_1, f_2) is defined as the set of all non-dominated (or efficient) solutions $\{x \in \mathbb{R}^n \mid / \exists y \in \mathbb{R}^n \; ; f_1(y) \leq f_1(x)$ and $f_2(y) \leq f_2(x)$ and at least one inequality is strict$\}$. The image of the Pareto set (in the objective space $\mathbb{R}^2$) is called the Pareto front of (f_1, f_2). We first remark that the Pareto set remains unchanged if we compose the objective functions with a strictly increasing function. More precisely the following lemma holds.

Lemma 15 (Invariance of the Pareto set to strictly increasing transformations of the objectives)**.** *Given a bi-objective problem* $x \mapsto (f_1(x), f_2(x))$ *and* $g_1 : \mathrm{Im}(f_1) \longmapsto \mathbb{R}$, $g_2 : \mathrm{Im}(f_2) \longmapsto \mathbb{R}$ *two strictly increasing functions, then* (f_1, f_2) *and* $(g_1 \circ f_1, g_2 \circ f_2)$ *have the same Pareto set.*

Proof. If x is not in the Pareto set of $(g_1 \circ f_1, g_2 \circ f_2)$, then their exists y such that $g_1 \circ f_1(y) \leq g_1 \circ f_1(x)$ and $g_2 \circ f_2(y) \leq g_2 \circ f_2(x)$ with one inequality being strict, which is equivalent to the fact that $f_1(y) \leq f_1(x)$ and $f_2(y) \leq f_2(x)$, with one inequality being strict. And vice versa. Hence x is not in the Pareto set of $(g_1 \circ f_1, g_2 \circ f_2)$ if and only if it is not in the Pareto set of (f_1, f_2), which shows that both problems have the same Pareto set. □ □

From now on (f_1, f_2) denote a bi-objective convex-quadratic problem. More precisely, let x_1, x_2 be two *different* vectors in $\mathbb{R}^n$, and $\alpha, \beta > 0$. Let Q_1 and Q_2 (in $\mathbb{R}^{n^2}$) be two positive definite matrices and consider the bi-objective *minimization* problem (f_1, f_2) defined for $x \in \mathbb{R}^n$ as

$$f_1(x) = \frac{1}{\alpha} (x - x_1)^\top Q_1 (x - x_1) \, , f_2(x) = \frac{1}{\beta} (x - x_2)^\top Q_2 (x - x_2) \, . \tag{6.1}$$

We denote this general bi-objective convex-quadratic problem by $\mathcal{P}$, and assume that the optimization goal is to find (an approximation of) the Pareto set of $\mathcal{P}$.

6.2.2 Pareto set

We characterize in this subsection the Pareto set of $\mathcal{P}$. We use the linear scalarization method to obtain the whole Pareto set. This is doable, whenever f_1 and f_2 are strict convex functions (see [112]). Then the Pareto set of $\mathcal{P}$ is described by the solutions of

$$\min_{x \in \mathbb{R}^n} (1 - t) f_1(x) + t f_2(x) \text{ , for } t \in [0, 1] \, .$$

We prove in the next proposition that the Pareto set of $\mathcal{P}$ is a continuous and differentiable parametric curve of $\mathbb{R}^n$ whose extremes are x_1 and x_2.

Proposition 29. *The Pareto set of* $\mathcal{P}$ *is the image of the function* φ *defined as*

$$\varphi : t \in [0, 1] \mapsto \left[(1 - t) Q_1 + t Q_2 \right]^{-1} \left[(1 - t) Q_1 x_1 + t Q_2 x_2 \right] \quad . \tag{6.2}$$

The function φ is differentiable and verifies for any t in $[0,1]$

$$(1-t)Q_1\left(\varphi(t)-x_1\right) = tQ_2\left(x_2-\varphi(t)\right), \tag{6.3}$$
$$t\left[(1-t)Q_1+tQ_2\right]\varphi'(t) = Q_1\left(\varphi(t)-x_1\right). \tag{6.4}$$

Hence, the Pareto set is a continuous (differentiable) curve of $\mathbb{R}^n$ whose extremes are $x_1 = \varphi(0)$ and $x_2 = \varphi(1)$.

Proof. For any s in $[0,1]$, define $g_s \stackrel{\text{def}}{=} (1-s)f_1 + sf_2$. We observe that g_s, like f_1 and f_2, is strictly convex, differentiable, and diverges to ∞ when $\|x\|$ goes to ∞ (where $\|x\|$ denotes the Euclidean norm). Then its critical point minimizes g_s. Let us now compute the gradient of g_s times $\alpha\beta$ for x in $\mathbb{R}^n$:

$$\alpha\beta\nabla g_s(x) = (1-s)\alpha\beta\,\nabla f_1(x) + s\alpha\beta\,\nabla f_2(x) = 2(1-s)\beta\,Q_1(x-x_1) + 2s\alpha\,Q_2(x-x_2)$$

$$\text{Thus, } \alpha\beta\nabla g_s(x) = 2\left[(1-s)\beta\,Q_1 + s\alpha\,Q_2\right]x - 2(1-s)\beta\,Q_1x_1 - 2s\alpha\,Q_2x_2.$$

Then it follows that for any s in $[0,1]$, the point that minimizes g_s (its critical point), denoted by $\tilde{x}_s$ verifies $\dfrac{(1-s)\beta\,Q_1 + s\alpha\,Q_2}{(1-s)\beta+s\alpha}\,\tilde{x}_s = \dfrac{(1-s)\beta\,Q_1x_1 + s\alpha\,Q_2x_2}{(1-s)\beta+s\alpha}$. Since $[0,1] \ni s \longmapsto \dfrac{s\alpha}{(1-s)\beta+s\alpha} \in [0,1]$ is bijective (its derivative is $s \longmapsto \dfrac{\alpha\beta}{\left((1-s)\beta+s\alpha\right)^2}$), then it is equivalent to parametrize the Pareto set with $t \stackrel{\text{def}}{=} \dfrac{s\alpha}{(1-s)\beta+s\alpha}$. Hence, the Pareto set is fully described by $(\varphi(t))_{t\in[0,1]}$ such that:

$$\left[(1-t)Q_1+tQ_2\right]\varphi(t) = (1-t)Q_1x_1 + tQ_2x_2, \tag{6.5}$$
$$(1-t)Q_1\left(\varphi(t)-x_1\right) = tQ_2\left(x_2-\varphi(t)\right). \tag{6.6}$$

The function $t \to \left[(1-t)Q_1+tQ_2\right]^{-1}$ is differentiable as inverse of a differentiable and invertible matrix function. Then φ is differentiable.
We differentiate (6.5) and multiply by t to obtain $t\left[(1-t)Q_1+tQ_2\right]\varphi'(t) = tQ_2x_2 - tQ_1x_1 + tQ_1\varphi(t) - tQ_2\varphi(t)$. Injecting in (6.6) gives: $t\left[(1-t)Q_1+tQ_2\right]\varphi'(t) = Q_1\left(\varphi(t)-x_1\right)$, for any $t \in [0,1]$. □ □

We obtain as corollary that when f_1 and f_2 have proportional Hessian matrices, then the Pareto set is the line segment between the optima of the functions f_1 and f_2.
Corollary 5. *In the case where f_1 and f_2 have proportional Hessian matrices, the Pareto set of $\mathcal{P}$ is the line segment between x_1 and x_2.*

Proof. In that case, their exists a real γ such that $\dfrac{Q_1}{\alpha} = \gamma\dfrac{Q_2}{\beta}$. Then, Proposition 29 implies that for any $t \in [0,1]$,

$$\varphi(t) = \left[(1-t)\gamma\frac{\alpha}{\beta}Q_2 + tQ_2\right]^{-1}\left[(1-t)\gamma\frac{\alpha}{\beta}Q_2x_1 + tQ_2x_2\right] = \frac{\gamma\alpha(1-t)x_1 + t\beta x_2}{(1-t)\alpha\gamma + t\beta},$$

which is $[x_1,x_2]$, since $[0,1] \ni t \longmapsto \frac{t\beta}{(1-t)\alpha\gamma+t\beta} \in [0,1]$ is a bijection. □ □

Using Lemma 15, we directly deduce the following corollary.

Corollary 6. *If f_1 and f_2 have proportional Hessian matrices, $g_1 : \mathrm{Im}(f_1) \longmapsto \mathbb{R}$, $g_2 : \mathrm{Im}(f_2) \longmapsto \mathbb{R}$ are two strictly increasing functions, then the Pareto set of the problem $(g_1 \circ f_1, g_2 \circ f_2)$ is the line segment between x_1 and x_2.*

As an example, the double-norm problem defined as:
$(x \to \|x - x_1\|_2, x \to \|x - x_2\|_2)$ can be seen as: $(g \circ f_1, g \circ f_2)$ where $g(x) = \sqrt{x}$, $f_1(x) = \|x - x_1\|_2^2$ and $f_2(x) = \|x - x_2\|_2^2$.
Then $(g \circ f_1, g \circ f_2)$ has the same Pareto set than the double-sphere problem (f_1, f_2), which is the line segment between x_1 and x_2. Therefore the Pareto front of the double-norm problem is described by $(t\|x_2 - x_1\|_2, (1-t)\|x_2 - x_1\|_2)_{t\in[0,1]}$. Thereby, the front is described by the function $u \longmapsto \|x_2 - x_1\|_2 - u$. We recover the well-known result that the double-norm problem has a linear front.

Corollary 6 allows also to recover the Pareto set description for the one-peak scenario in the Mixed-Peak Bi-Objective Problem (see [122] and [123]).[1]

In general, the Pareto set of a bi-objective convex-quadratic problem is not necessarily a line segment. Consider for instance for $n = 2$ the case where $x_1 = (0,0)^\top$, $x_2 = (1,1)^\top$ and where we generate two different matrices Q_1 and Q_2 by randomly rotating a diagonal matrix with eigenvalues 1 and 10. Two resulting Pareto fronts associated to different random rotations are depicted in Figure 6.1.

For $n = 10$, we also define $\mathcal{P}_{10}$ setting $x_1 = (0,\dots,0)^\top$, $x_2 = (1,\dots,1)^\top$ and Q_1 and Q_2 as diagonal matrices such that for $i = 1,\dots,10$

$$Q_1(i,i) = 100^{\frac{i-1}{9}}, \text{and } Q_2(i,i) = 10^{\frac{i-1}{9}}. \tag{6.7}$$

The different coordinates of the Pareto set given in (6.3) are depicted in Figure 6.1.

6.2.3 Convexity of the Pareto front

Corollary 5 proves that in the case where we have proportional Hessian matrices in problem $\mathcal{P}$, the Pareto set is a line segment. Then it is reasonable to expect a simple analytic expression for the corresponding Pareto front. In what follows, we will express the Pareto front of a bi-objective problem as a one-dimensional function $u \in \mathbb{R} \mapsto g(u)$. Formally, if $t \in \mathbb{R} \mapsto \varphi(t) \in \mathbb{R}^n$ is a parametrization of the Pareto set, then the function g satisfies $f_2(\varphi(t)) = g(f_1(\varphi(t)))$. It is well-known that when (f_1, f_2) is the double-sphere, that is $f_1(x) = \frac{1}{n}\sum_{i=1}^n x_i^2$ and $f_2(x) = \frac{1}{n}\sum_{i=1}^n (x_i - 1)^2$, then the Pareto front expression is given by $g(u) = (1 - \sqrt{u})^2$ [65]. In the next proposition, we show

[1] In that scenario, we set $f_1(x) = (x-c)^\top \Sigma (x-c)$, $f_2(x) = (x-c')^\top \Sigma' (x-c')$ (f_1 and f_2 are seen as squares of the Mahalanobis distance to the optima, with respect to the Hessian matrices), $g_1(u) = 1 - \dfrac{h_1}{1 + \frac{\sqrt{u}}{r_1}}$, $g_2(u) = 1 - \dfrac{h_2}{1 + \frac{\sqrt{u}}{r_2}}$.

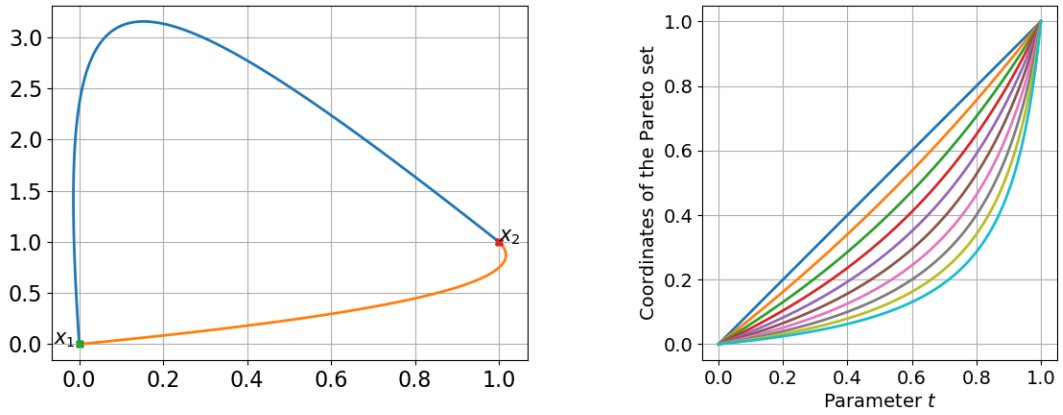


Figure 6.1: Left: Two Pareto sets for $n = 2$ represented in $\mathbb{R}^2$ with Q_1 and Q_2 randomly sampled and different. Right: Pareto set for $n = 10$ with matrices given in (6.7) represented as the function of the parameter t given in (6.3). The coordinates are ordered, the first one is on top and last one below.

that this expression of the Pareto front holds (up to a normalization) for all bi-objective convex-quadratic problems, provided the Hessians of f_1 and f_2 are proportional.

Proposition 30. *When we have proportional Hessian matrices in the problem $\mathcal{P}$, the Pareto front is described by the following continuous and convex function:*

$$u \in [0, \kappa_\alpha] \mapsto \kappa_\beta \left(1 - \sqrt{\frac{u}{\kappa_\alpha}}\right)^2, \text{ where } \begin{cases} \kappa_\alpha = \dfrac{(x_2 - x_1)^\top Q_1 (x_2 - x_1)}{\alpha} \\ \kappa_\beta = \dfrac{(x_2 - x_1)^\top Q_2 (x_2 - x_1)}{\beta} \end{cases} \tag{6.8}$$

Proof. Denote $u \overset{\text{def}}{=} f_1 \circ \varphi$ and $v \overset{\text{def}}{=} f_2 \circ \varphi$, where $\varphi : [0,1] \ni t \longmapsto (1-t)x_1 + tx_2 \in [x_1, x_2]$ is the line segment between x_1 and x_2.
For any $t \in [0,1]$, $u(t) = f_1(\varphi(t)) = \frac{1}{\alpha}(x_2 - x_1)^\top Q_1 (x_2 - x_1) t^2, v(t) = f_2(\varphi(t)) = \frac{1}{\beta}(x_2 - x_1)^\top Q_2 (x_2 - x_1)(1-t)^2$. It follows that for any $t \in [0,1]$:

$$v(t) = \frac{(x_2 - x_1)^\top Q_2 (x_2 - x_1)}{\beta} \left(1 - \sqrt{\frac{\alpha u(t)}{(x_2 - x_1)^\top Q_1 (x_2 - x_1)}}\right)^2. \qquad \square$$

$\square$

From Proposition 30, we deduce that if we set $\kappa_\alpha = \kappa_\beta = 1$, then the Pareto front will be independent from the Hessian matrix and will be described by the front of the double-sphere problem: $u \mapsto (1 - \sqrt{u})^2$.

We investigate now the general case where the Hessians of the functions f_1 and f_2 are not necessarily proportional. Yet, before digging into the general convex-quadratic problems, we show a result on the shape of the Pareto front of a larger class of bi-objective problems.

Theorem 13. *Let $f_1 : \mathbb{R}^n \longmapsto \mathbb{R}$ and $f_2 : \mathbb{R}^n \longmapsto \mathbb{R}$ be strict convex differentiable functions such that the problem (f_1, f_2) has, as Pareto set, the image of a differentiable function $\varphi : [0,1] \longmapsto \mathbb{R}^n$.*

*Assume that: (i) $f_1 \circ \varphi$ is strictly monotone, (ii) $\lim_{t\to 0} \dfrac{(f_1 \circ \varphi)'(t)}{t} \neq 0$ and (iii) $\lim_{t\to 1} \dfrac{(f_2 \circ \varphi)'(t)}{1-t} \neq 0$. Then, the Pareto front is a **convex** curve, with **vertical** tangent at $t = 0$ and horizontal tangent at $t = 1$.*

Proof. Denote by $u \overset{\text{def}}{=} f_1 \circ \varphi$ and $v \overset{\text{def}}{=} f_2 \circ \varphi$. Then the Pareto front is described by the parametric equation $(u(t), v(t))$, for $t \in [0,1]$. We will show that $u'v'' - u''v' > 0$ which implies the convexity of the curve.

By linear scalarization (see [112], or weighted sum method in [76]), as in the proof of Proposition 29, we have $(1-t)\nabla f_1(\varphi(t)) + t\nabla f_2(\varphi(t)) = 0$. If we take the scalar product of the former equation with $\varphi'(t)$, we obtain that

$$(1-t)\left\langle \nabla f_1(\varphi(t)),\, \varphi'(t)\right\rangle + t\left\langle \nabla f_2(\varphi(t)),\, \varphi'(t)\right\rangle = 0. \tag{6.9}$$

Moreover, for any differentiable function f with suitable domains,

$$(f\circ\varphi)'(t) = \mathrm{d}\,(f\circ\varphi)_t(1) = \mathrm{d}f_{\varphi(t)}\left(\mathrm{d}\varphi_t(1)\right) = \left\langle \nabla f(\varphi(t)),\, \varphi'(t)\right\rangle. \tag{6.10}$$

Inserting this in (6.9) shows $(1-t)(f_1\circ\varphi)'(t) + t(f_2\circ\varphi)'(t) = 0$, which is the same as:

$$(1-t)u'(t) + tv'(t) = 0, \text{ for any } t \in [0,1]. \tag{6.11}$$

Since $\lim_{t\to 0} \dfrac{(f_1\circ\varphi)'(t)}{t}$ exists, (6.11) implies that:

$$v'(t) = \left(1 - \frac{1}{t}\right)u'(t), \text{ for any } t \in [0,1]. \tag{6.12}$$

By deriving (6.12) and multiplying by $u'(t)$ in a suitable way, we obtain

$$u'(t)v''(t) = \frac{1}{t^2}u'(t)^2 + \left(1-\frac{1}{t}\right)u'(t)u''(t), \text{ for any } t \in [0,1]. \tag{6.13}$$

Using (6.12) in (6.13) gives $u'(t)v''(t) = \frac{1}{t^2}u'(t)^2 + v'(t)u''(t)$. Thanks to the assertions on $f_1 \circ \varphi$, we have that $u'(t)v''(t) - u''(t)v'(t) > \frac{1}{t^2}u'(t)^2 > 0$, for any $t \in [0,1]$. Thus, the Pareto front is a **convex** curve.

Evaluating (6.11) at $t = 0$ and at $t = 1$ implies that $u'(0) = 0, v'(1) = 0$. And if we divide (6.11) by t (resp. $1-t$) and take the limit to 0 (resp. 1), it follows that $v'(0) \neq 0$ (resp. $u'(1) \neq 0$). Thereby we also obtain the derivative assumptions on the extremal points. □ □

Remark 1. *Note that the above result about the tangents in the extremal points have additional consequences: according to [21], the assumptions of Theorem 13 imply that the extremal points are never included in any optimal μ-distributions of the Hypervolume indicator.*

We now deduce the convexity of the Pareto front for convex-quadratic bi-objective problems and characterize the derivatives at the extremes of the front.

Corollary 7. *For the problem $\mathcal{P}$, the Pareto front is a **convex** curve, with **vertical** tangent at $(0, f_2(x_1))$ and **horizontal** tangent at $(f_1(x_2), 0)$.*

Proof. We will show that $f_1 \circ \varphi$ verifies the assumptions of Theorem 13. From (6.10) we know that

$$(f_1 \circ \varphi)'(t) = \langle \nabla f_1(\varphi(t)), \varphi'(t) \rangle . \tag{6.14}$$

In addition, $\nabla f_1(\varphi(t)) = \frac{2}{\alpha} Q_1 (\varphi(t) - x_1)$ and Eq. (6.4) of Proposition 29 gives

$$t\left[(1-t)Q_1 + tQ_2\right]\varphi'(t) = Q_1(\varphi(t) - x_1).$$

Multiplying (6.14) by $t \in [0,1]$ shows

$$t(f_1 \circ \varphi)'(t) = \frac{2}{\alpha} \left\langle \left[(1-t)Q_1 + tQ_2\right]^{-1} Q_1(\varphi(t) - x_1), Q_1(\varphi(t) - x_1) \right\rangle . \tag{6.15}$$

Since $\left[(1-t)Q_1 + tQ_2\right]^{-1}$ is a positive definite matrix, then $t(f_1 \circ \varphi)'(t) \geq 0$. Let us prove that $\varphi(t) \neq x_1$, for $t \in (0,1]$. By contradiction, assume that there exists $t \in (0,1]$ such that $\varphi(t) = x_1$. Then Equation (6.3) in Proposition 29 shows that: $tQ_2(x_2 - \varphi(t)) = (1-t)Q_1(\varphi(t) - x_1) = 0$, which implies that $x_2 = \varphi(t) = x_1$: that is impossible since $x_1 \neq x_2$. Hence, by *reductio ad absurdum*, $\varphi(t) \neq x_1$, for $t \in (0,1]$. From (6.15), it follows that

$$(f_1 \circ \varphi)'(t) > 0, \text{ for any } t \in (0,1]. \tag{6.16}$$

If we use again the relation from Proposition 29, we obtain

$$\lim_{t \to 0} \frac{Q_1(\varphi(t) - x_1)}{t} = Q_2(x_2 - \varphi(0)) = Q_2(x_2 - x_1).$$

Injecting this result in (6.15), it follows that:

$$\begin{aligned} \lim_{t \to 0} \frac{(f_1 \circ \varphi)'(t)}{t} &= \frac{2}{\alpha} \left\langle Q_1^{-1} Q_2(x_2 - x_1), Q_2(x_2 - x_1) \right\rangle \\ &> 0, \text{ since } (Q_1^{-1} \text{ is a positive definite matrix}) \end{aligned} \tag{6.17}$$

In the same way as above, we obtain that

$$\lim_{t \to 1} \frac{(f_2 \circ \varphi)'(t)}{1-t} = -\frac{2}{\beta} \left\langle Q_2^{-1} Q_1(x_1 - x_2), Q_1(x_1 - x_2) \right\rangle < 0 \ . \tag{6.18}$$

Equations (6.16), (6.17), and (6.18) allow us to apply Theorem 13. □ □

We illustrate the previous corollary by taking three random instances of our general problem $\mathcal{P}$, with the scalings always chosen as $\alpha = \beta = \max(f_1(x_2), f_2(x_1))$. The Pareto fronts are presented in Figure 6.2. We observe that the Pareto fronts are convex and their derivatives are infinite on the left and zero on the right.

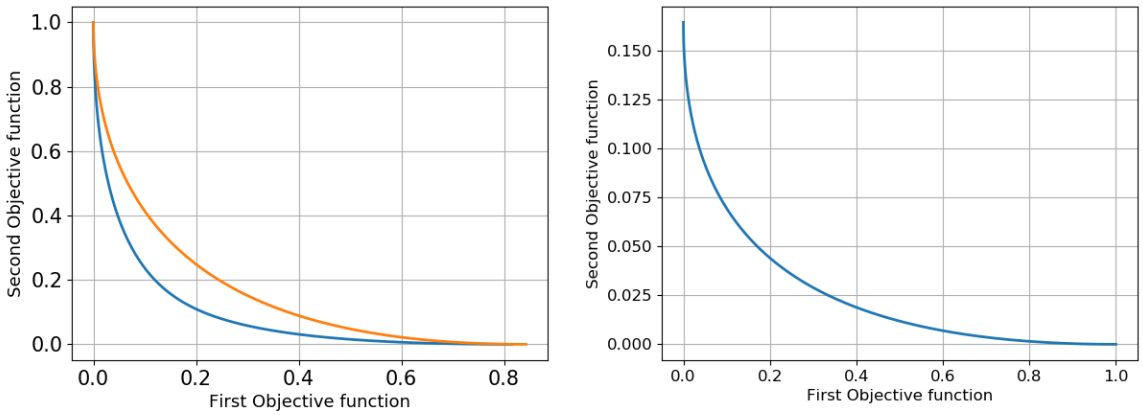


Figure 6.2: Left: Two Pareto fronts for $n = 2$ represented in $\mathbb{R}^2$ with Q_1, Q_2 randomly sampled and different. Right: Pareto front for $n = 10$ with matrices given in (6.7).

6.3 New Classes of bi-objective test functions

Bi-objective problems using convex-quadratic functions have been used to test MO algorithms (see for example [104]). Problems where both Hessian matrices have the same eigenvalues have been used in particular. Yet, test problems considered so far do not explore the full possibilities of properties that can be tested. We therefore extend the test problems from the literature to be able to capture more properties. To do so we present seven classes of bi-objective convex-quadratic problems where the eigenspectra of both Hessian matrices are equal. A natural extension of these classes is to use in each objective different eigenspectra, Δ, which leads in general to a nonlinear Pareto set.

The proposed construction parametrizes, apart from search space translations, *all* bi-objective convex-quadratic functions with identical Hessian eigenspectrum in seven classes with increasing difficulty. The particular focus is on problems with a linear Pareto set in five of the seven classes. Some classes represent essentially different problems, hence we do not expect uniform performance over all problems within each class. Independently of the given construction, invariance to search space rotation can be tested by applying an orthogonal transformation to the input argument.

We start from a diagonal matrix Δ with positive entries that define a separable convex-quadratic function $f(x) = \frac{1}{\alpha} x^\top \Delta x$. For instance, Δ can be equal to the identity and we recover the sphere function. If $\Delta(1,1) = 1$, $\Delta(n,n) = 10^8$ and $\Delta(i,i) = 10^4$, we recover the separable cig-tab function and if $\Delta(i,i) = 10^{6\frac{i-1}{n-1}}$, we recover the separable ellipsoid function.

In the sequel, O and O_2 denote orthogonal matrices. O_1 is either a permutation matrix, or an orthogonal matrix, depending on the context. The classes of problems proposed are summarized in Table 6.1 and Table 6.2.

The Sep problem classes

We define the **Sep-k** class by considering two separable functions and place the optimum of f_1 in 0 and of f_2 in the k^{th} unit vector: $f_{1,\Delta}^{\textbf{sep-}k}(x) = \frac{1}{\alpha}(x - x_1)^\top \Delta (x - x_1)$ and $f_{2,\Delta}^{\textbf{sep-}k}(x) = \frac{1}{\beta}(x - x_2)^\top \Delta (x - x_2)$, where $x_1 = (0, \ldots, 0)^\top$ and $x_2 = (0, \ldots, 0, \sqrt{n}, 0, \ldots, 0)^\top$ where $\sqrt{n}$ is at coordinate k. According to Corollary 5, the Pareto set of this class of problems is the line segment between the optima of the single-objective problems. These problems allow to test the performance on separable problems with a Pareto set aligned with the coordinate axis and check the sensibility with respect to different axes (by varying k).

For the **Sep-O** class, we only change the location of the optimum of the second objective by taking $x_2 = O(1, \ldots, 1)^\top$. If O has elements $\notin \{-1, 0, 1\}$, the Pareto set is not anymore aligned with the coordinate system, but the objectives f_1 and f_2 themselves remain separable. Comparing with class **Sep-k**, we can test whether having the Pareto set not aligned with the coordinate axis has an influence on the performance of the algorithm.

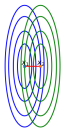
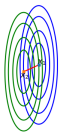
For the **Sep-Two-O** class, we define $f_{1,\Delta}^{\textbf{sep-Two-O}}(x) = \frac{1}{\alpha}(x - x_1)^\top \Delta (x - x_1)$ and

$f_{2,\Delta}^{\textbf{sep-Two-O}}(x) = \frac{1}{\beta}(x - x_2)^\top O_1^\top \Delta O_1 (x - x_2)$ where O_1 is a permutation matrix, $x_1 = (0, \ldots, 0)^\top$ and $x_2 = O(1, \ldots, 1)^\top$. The matrix $O_1^\top \Delta O_1$ is also diagonal, and thereby each function is separable. Yet the Pareto set is generally not a line segment anymore since we have different Hessian matrices. We can test here the difficulty of having a nonlinear Pareto set on separable functions.

The One and the One-O problem classes.

We now consider non-separable problems with a line segment as Pareto set. We define $f_{1,\Delta}^{\textbf{one}}(x) = \frac{1}{\alpha}(x - x_1)^\top O_1^\top \Delta O_1 (x - x_1)$ and $f_{2,\Delta}^{\textbf{one}}(x) = \frac{1}{\alpha}(x - x_2)^\top O_1^\top \Delta O_1 (x - x_2)$, where O_1 is an orthogonal matrix, $x_1 = (0, \ldots, 0)^\top$ and $x_2 = (1, \ldots, 1)^\top$. We replace x_2 by Ox_2 to obtain the **One-O** problems.

Table 6.1: Unconstrained quadratic bi-objective test problems: Δ is a positive diagonal matrix, O is an orthogonal matrix, O_1 is a permutation matrix.

	Sep-k	**Sep-O**	**Sep-Two-O**
x_1	$(0,\ldots,0)^\top$	$(0,\ldots,0)^\top$	$(0,\ldots,0)^\top$
x_2	$\underbrace{(0,..,\sqrt{n},..,0)^\top}_{\sqrt{n} \text{ is at row } k}$	$O(1,\ldots,1)^\top$	$O(1,\ldots,1)^\top$
Q_1, Q_2	Δ, Δ	Δ, Δ	Δ, $O_1^\top \Delta O_1$
Level sets			

These two problem classes allow to test the performance on non-separable problems that have a line segment as Pareto set comparing in particular to class **Sep-O**. Up to a reformulation, the problems ELLI1 and CIGTAB1 from [104] are from the **One-O** problem class. Generally, we do not expect different performance over all problems of the **One** vs the **One-O** class.

The Two and the Two-O problem classes.

For these classes, we rotate each function independently; then the Pareto set is generally not a line segment anymore. We define $f^{\textbf{two}}_{1,\Delta}(x) = \frac{1}{\alpha}(x - x_1)^\top O_1^\top \Delta O_1 (x - x_1)$ and $f^{\textbf{two}}_{2,\Delta}(x) = \frac{1}{\alpha}(x - x_2)^\top O_2^\top \Delta\ O_2 (x - x_2)$, with O_1 orthogoanal, $x_1 = (0,\ldots,0)^\top$ and $x_2 = (1,\ldots,1)^\top$. The corresponding **O** problems are obtained with Ox_2 replacing x_2. All presented classes are subsets of the **Two-O** class. ELLI2 and CIGTAB2 from [104] fall within the **Two-O** class. Compared to the respective **One** classes, we can test the impact of having a nonlinear Pareto set.

6.4 Summary

We have presented an analytic description of the Pareto set for quadratic bi-objective problems. We have shown that the Pareto set is a line segment when both objectives have proportional Hessian matrices and deduced a complete description of the Pareto front in that case. We have also proven that some properties of the double-sphere are conserved in a wider framework that includes the general quadratic bi-objective problem: the Pareto front remains convex and its vertical and horizontal tangents remain

Table 6.2: Unconstrained quadratic bi-objective test problems: Δ is a positive diagonal matrix, O, O_1 and O_2 are three independent orthogonal matrices.

	One	**One-O**	**Two**	**Two-O**
x_1	$(0,\ldots,0)^\top$	$(0,\ldots,0)^\top$	$(0,\ldots,0)^\top$	$(0,\ldots,0)^\top$
x_2	$(1,\ldots,1)^\top$	$O(1,\ldots,1)^\top$	$(1,\ldots,1)^\top$	$O(1,\ldots,1)^\top$
Q_1, Q_2	$O_1^\top \Delta O_1$, $O_1^\top \Delta O_1$	$O_1^\top \Delta O_1$, $O_1^\top \Delta O_1$	$O_1^\top \Delta O_1$, $O_2^\top \Delta O_2$	$O_1^\top \Delta O_1$, $O_2^\top \Delta O_2$
Level sets				

at the extremal points of the front. Such assumptions on the derivatives imply that when looking at the optimal μ-distributions of the Hypervolume indicator, the extremal points are always excluded [21]. We have also presented several classes of problems, where each one tests a specific capability of the multiobjective algorithm.

Chapter 7

Uncrowded Hypervolume Improvement: COMO-CMA-ES and the Sofomore framework

Contents

Note: the content of this chapter is presented in [185]:
Cheikh Toure, Nikolaus Hansen, Anne Auger and Dimo Brockhoff, *Uncrowded hyper-*

volume improvement: COMO-CMA-ES and the sofomore framework, Proceedings of the Genetic and Evolutionary Computation Conference, 2019, pp 638–646.

We present a framework to build a multiobjective algorithm from single-objective ones. This framework addresses the $p \times n$-dimensional problem of finding p solutions in an n-dimensional search space, maximizing an indicator by dynamic subspace optimization. Each single-objective algorithm optimizes the indicator function given $p-1$ fixed solutions. Crucially, dominated solutions minimize their distance to the empirical Pareto front defined by these $p-1$ solutions. We instantiate the framework with CMA-ES as single-objective optimizer. The new algorithm, COMO-CMA-ES, is empirically shown to converge linearly on bi-objective convex-quadratic problems and is compared to MO-CMA-ES, NSGA-II and SMS-EMOA.

7.1 Introduction

Multiobjective optimization problems must be solved frequently in practice. In contrast to the optimization of a single objective, solving a multiobjective problem involves to handle trade-offs or incomparabilities between the objective functions such that the aim is to approximate the Pareto set–the set of all Pareto-optimal, or non-dominated solutions. One might be interested to obtain an approximation of unbounded size (the more points the better) or just to have p points approximating the Pareto set. Evolutionary Multiobjective Optimization (EMO) algorithms aim at such an approximation in a single algorithm run whereas more classical approaches, e.g. optimizing a weighted sum of the objectives with changing weights, operate in multiple runs.

The first introduced EMO algorithms simply changed the selection of an existing single-objective evolutionary algorithm keeping the exact same search operators. The population at a given iteration was then providing an approximation of the Pareto set. This idea led to the practically highly successful NSGA-II algorithm [56] that employs a two-step fitness assignment: after a first non-dominated ranking [71], solutions with equal non-domination rank are further distinguished by their crowding distance—based on the distance of each solution to its neighbors in objective space. However, it has been pointed out that NSGA-II does not converge to the Pareto set in a mathematical sense due to so-called deteriorative cycles: if all population members of the algorithm are non-dominated at some point in time, it is only the crowding distance that is optimized, without indicating any search direction towards the Pareto set to the algorithm. As a result, solutions which had been non-dominated solutions at some point in time can be replaced by previously dominated ones during the optimization, ending up in a cyclic but not in convergent behavior [29].

To improve the convergence properties of EMO algorithms, different approaches have been introduced later, most notably the indicator-based algorithms and especially algorithms based on the hypervolume indicator. They replace the crowding distance of NSGA-II with the (hypervolume) indicator contribution, see e.g. [30, 104]. Using the hypervolume indicator has the advantage that it is the only known strictly monotone quality indicator [126] (see also next section) and thus, its optimization will result in solution sets that are subsets of the Pareto set.

The optimization goal of indicator-based algorithms such as SMS-EMOA [30] or MO-CMA-ES [104] is to find the best set of p solutions with respect to a given quality indicator (the set with the largest quality indicator value among all sets of size p). This optimal set of p solutions is known as the optimal p-distribution [21]. In principle, the search for the optimal p-distribution can be formalized as a $p \cdot n$-dimensional optimization problem where p is the number of solutions and n is the dimension of the search space.

As we will discuss later, it turns out that this optimization problem is not only of too high dimension in practice but also flat in large regions of the search space if the hypervolume indicator is the underlying quality indicator. The combination of non-dominated ranking and hypervolume contribution as in SMS-EMOA or MO-CMA-ES corrects for this flatness, but also introduces search directions that are pointing towards already existing non-dominated solutions and not towards not-yet-covered regions of the Pareto set. In this chapter, we show that we can correct the flat region of the hypervolume indicator by introducing a search bias towards yet-uncovered regions of the Pareto set by adding the distance to the empirical non-domination front, which leads to the new notion of Uncrowded Hypervolume Improvement. Then, we define a (dynamic) fitness function that can be optimized by single-objective algorithms. From there, going back to this original idea of EMO algorithms to use single-objective optimizers to build an EMO, we define the *Single-objective Optimization FOr Optimizing Multiobjective Optimization pRoblEms* framework (Sofomore) to build in an elegant manner, a multiobjective algorithm from a set of p single-objective optimizers. Each single-objective algorithm optimizes (iteratively or in parallel) a dynamic fitness that depends on the output of the other $p-1$ optimizers.

We instantiate the Sofomore framework with the state-of-the-art single-objective algorithm CMA-ES. We show experimentally that the ensuing COMO-CMA-ES (Comma-Selection Multiobjective CMA-ES) exhibits linear convergence towards the optimal p-distribution on a wide variety of bi-objective convex quadratic functions. In contrast, default implementations of the SMS-EMOA where the reference point is fixed and NSGA-II do not exhibit this linear convergence. The comparison between COMO-CMA-ES and a previous MATLAB implementation of the elitist MO-CMA-ES also shows the same or an improved convergence *speed* in COMO-CMA-ES except for the double sphere function.

The chapter is structured as follows. In the next section, we start with preliminaries

related to multiobjective optimization and quality indicators. Section 7.3 discusses the fitness landscape of indicator- and especially hypervolume-based quality measures and eventually introduces our Sofomore framework. Section 7.4 gives details about the new COMO-CMA-ES algorithm as an instantiation of Sofomore with CMA-ES. Section 7.5 experimentally validates the new algorithm and compares it with three existing algorithms from the literature and Section 7.6 discusses the results and concludes the chapter.

7.2 Preliminaries

In the remainder, we will use the term *empirical non-dominated front* or *empirical Pareto front* ($\mathbf{EPF}_{S,\mathbf{r}}$) for objective vectors that are on the boundary of the (objective space) region dominating a reference point $\mathbf{r} \in \mathbb{R}^m$, and not dominated by any element of $\mathbf{f}(S)$ with $S \subset \mathbb{R}^n$:

$$\mathbf{EPF}_{S,\mathbf{r}} = \partial U_{S,\mathbf{r}}, \text{ with } U_{S,\mathbf{r}} = \{z \prec \mathbf{r}; \forall s \in S, \mathbf{f}(s) \not\prec z\} \tag{7.1}$$

where $\partial U_{S,\mathbf{r}}$ is the boundary of the non-dominated region $U_{S,\mathbf{r}}$. Note that $\mathbf{EPF}_{S,\mathbf{r}} \cap \mathbf{f}(\mathbb{R}^n)$ is the Pareto front when S contains the Pareto set.

Indicator-Based Set Optimization Problems Pareto sets and Pareto fronts are, under mild assumptions, $m-1$ dimensional manifolds. In practice, we are often interested in a finite size approximation of these sets with, let us say, $p\,(\geq 1)$ many search points. To assess the quality of a Pareto set approximation $S \subseteq \mathbb{R}^n$, we slightly modify the set of departure of a quality indicator with respect to Equation (5.3):

$$\mathcal{I} : 2^{\mathbb{R}^n} \longrightarrow \mathbb{R} \tag{7.2}$$

assigns a real valued quality $\mathcal{I}(S)$ to S. Formally speaking, this transforms the original multiobjective optimization of $\mathbf{f}(x)$ into the single-objective set problem of finding the so-called optimal p-distribution [22]

$$X_p^* = \underset{X \subseteq \mathbb{R}^n,\, |X| \leq p}{\arg\max} \quad \mathcal{I}(X) \tag{7.3}$$

as the set of search points of cardinality p (or lower) with the highest indicator value among all sets of this size [21].

Natural candidates for practically relevant quality indicators are monotone or even strictly monotone indicators such as the epsilon-indicator [204], the R2 indicator [78], or the hypervolume indicator ([21,205], still the only known strictly monotone indicator family to date). We remind that an indicator is called monotone if $A \preceq B \implies \mathcal{I}(A) \geq \mathcal{I}(B)$, for A, B subsets of $\mathbb{R}^n$—or in other words, if it does not contradict the weak Pareto dominance relation. If $A \prec B \implies \mathcal{I}(A) > \mathcal{I}(B)$, we say that $\mathcal{I}$ is strictly monotone.

Hypervolume, Hypervolume Contribution, and Hypervolume Improvement Because the hypervolume indicator [21, 205] and its weighted variant is the only known strictly monotone indicator, we use it as well in our framework. We remind here that the hypervolume $HV_{\mathbf{r}}$ [205] of a finite set of solutions $S \subset \mathbb{R}^n$ with respect to the reference point $\mathbf{r} \in \mathbb{R}^m$ is defined as $HV_{\mathbf{r}}(S) = \lambda_m\left(\{z \in \mathbb{R}^m; \exists y \in S, \mathbf{f}(y) \prec z \prec \mathbf{r}\}\right)$, where λ_m is the Lebesgue measure on the objective space $\mathbb{R}^m$ and $\mathbf{f}$ is the objective function. In the case of two objective functions, the hypervolume indicator value of p non-dominated solutions $S = \{s^{(1)}, \ldots, s^{(p)}\}$ with $\mathbf{f}_1(s^{(1)}) \leq \mathbf{f}_1(s^{(2)}) \leq \ldots \leq \mathbf{f}_1(s^{(p)})$ can also be written as the sum of the area of p axis parallel rectangles: $HV_{\mathbf{r}}(S) = \sum_{i=1}^{p}(\mathbf{f}_1(s^{(i+1)}) - \mathbf{f}_1(s^{(i)})) \cdot (r_2 - \mathbf{f}_2(s^{(i)}))$; $\mathbf{f}_1(s^{(p+1)}) \stackrel{\text{def}}{=} r_1$.

Furthermore, the hypervolume *c*ontribution $\mathrm{HVC}(\mathbf{s}, S)$ of a search point $\mathbf{s} \in \mathbb{R}^n$ to a solution set $S \subseteq \mathbb{R}^n$ with respect to the reference point $\mathbf{r} \in \mathbb{R}^m$ is the hypervolume indicator value that we lose when we remove $\mathbf{s}$ from the set [38]: $\mathrm{HVC}_{\mathbf{r}}(\mathbf{s}, S) = HV_{\mathbf{r}}(S) - HV_{\mathbf{r}}(S \setminus \{\mathbf{s}\})$.

Also, the Hypervolume Improvement $\mathrm{HVI}_{\mathbf{r}}(\mathbf{s}, S)$ of a search point $\mathbf{s} \in \mathbb{R}^n$ to a finite set $S \subset \mathbb{R}^n$ with respect to the reference point $\mathbf{r} \in \mathbb{R}^m$ is defined as [63, 198] : $\mathrm{HVI}_{\mathbf{r}}(\mathbf{s}, S) = HV_{\mathbf{r}}(S \cup \{\mathbf{s}\}) - HV_{\mathbf{r}}(S)$. Or in other words, $\mathrm{HVI}_{\mathbf{r}}(\mathbf{s}, S)$ equals the increase in hypervolume when $\mathbf{s}$ is added to the set S. Up to a null set $\mathrm{HVI}_{\mathbf{r}}(\mathbf{s}, S) = \mathrm{HVC}_{\mathbf{r}}(\mathbf{s}, S \cup \{\mathbf{s}\})$.

7.3 Sofomore: Building Multiobjective from Single-Objective Algorithms

Quality indicators have been introduced as a way to measure the quality of a set of objective vectors but also to define a multiobjective optimization problem as a single-objective set problem of maximizing the quality indicator as in (7.3). This naturally defines a single-objective $p \times n$ dimensional problem to be maximized

$$\mathcal{F} : (x_1, \ldots, x_p) \in (\mathbb{R}^n)^p \mapsto \mathcal{I}(\{x_1, \ldots, x_p\}) \quad . \tag{7.4}$$

Because n and in particular p are typically large in practice, we usually do not attempt to solve a multiobjective optimization problem by directly optimizing (7.4). Nevertheless, when $\mathcal{I}$ is the hypervolume indicator, Hernández et al. suggest to use a Newton method to directly solve (7.4). It assumes that $\mathbf{f}$ is twice continuously differentiable, in which case the gradient and Hessian of $\mathcal{F}$ can be computed analytically [99]. Yet, directly attacking (7.4) is not possible because dominated points have a zero sub-gradient and the Newton direction is therefore zero. Thus, Hernández et al. need to start from a set of non-dominated points, close enough to the Pareto set, which requires in practice to couple the approach with another algorithm [99].

Instead of directly optimizing (7.4), our proposed Sofomore framework performs iterative *s*ubspace optimization of the function $\mathcal{F}$ and penalizes the flat landscape of $\mathcal{F}$ in dominated regions. More precisely, the basic idea behind Sofomore is to optimize $\mathcal{F}$ subspace- or component-wise, by iteratively fixing all but one search point $x^{(i)}$ and only optimizing the indicator with respect to $x^{(i)}$ while the other search points $X^{\neg i} := \{x^{(1)}, \ldots, x^{(i-1)}, x^{(i+1)}, \ldots, x^{(p)}\}$ are temporarily fixed. Hence we maximize the functions

$$\Phi_{\mathcal{I},X^{\neg i}} : x \in \mathbb{R}^n \mapsto \mathcal{I}(\{x\} \cup X^{\neg i}) \ . \tag{7.5}$$

If the placement of each of the p search points $x^{(i)}$ $(1 \leq i \leq p)$ is optimized iteratively by fixing a different point set each time, as we suggest in our Sofomore framework below, we are in the setup of optimizing a dynamic fitness. More details on this aspect of our Sofomore framework will be given below in Section 7.3.2.

7.3.1 A Fitness Function for Subspace Optimization

If we use as quality indicator $\mathcal{I}$ in (7.5) a (strictly) monotone indicator like the hypervolume indicator, the overall fitness Φ is flat in the interior domain of regions where points are dominated. Hence, we suggest to not optimize (7.5) directly but to unflatten it in dominated areas of the search space without changing the optimization goal.

Any solution x that is dominated by the other points in $X^{\neg i}$ will receive zero fitness Φ when we use as indicator in (7.5) the hypervolume indicator of the entire set $\{x\} \cup X^{\neg i}$ with respect to the reference point $\mathbf{r}$ or replace it with the hypervolume improvement $\mathrm{HVI}_{\mathbf{r}}(x, X^{\neg i})$ of the solution x to $X^{\neg i}$. This situation is depicted in the first column of Figure 7.1 where for a fixed set of six arbitrarily chosen search points, the hypervolume improvement's level sets (of equal fitness) in both search and objective space are shown. This flat fitness with zero gradient will not allow to steer the search towards better search points which has also been highlighted by Hernández et al. [99].

A common approach to guide an optimization algorithm in the dominated space is to use the hypervolume (contribution) as secondary fitness after non-dominated sorting [71], as it is done for example in the SMS-EMOA [30] or the MO-CMA-ES [104]. The idea is that all search points with a worse non-domination rank get assigned a fitness that is worse than for search points with a better non-domination rank. Within a set of the same rank, the hypervolume contribution with respect to all points with the same rank is used to refine the fitness. The middle column of Figure 7.1 shows the resulting level sets of equal fitness. As we can see, this fitness assignment distinguishes between dominated solutions, i.e. the fitness is not flat anymore. Yet it still has another major disadvantage: the search direction in the dominated area (perpendicular to its level sets) points towards already existing non-dominated solutions. Attracting dominated solutions towards non-dominated solutions seems however undesirable,

as they will compete for the same hypervolume area. Instead, we want dominated points to enter the uncrowded space *between* non-dominated points thereby complementing their hypervolume contribution (improvement).

Uncrowded Hypervolume Improvement For this purpose, we define the Uncrowded Hypervolume Improvement UHVI based on the Hypervolume Improvement for non-dominated search points and on the Euclidean distance to the non-dominated region for dominated search points. More concretely, $\mathsf{UHVI}_{\mathbf{r}}(\mathbf{s}, S)$ of a search point $\mathbf{s} \in \mathbb{R}^n$ with respect to a finite set $S \subset \mathbb{R}^n$ and the reference point $\mathbf{r} \in \mathbb{R}^m$ is defined as

$$\mathsf{UHVI}_{\mathbf{r}}(\mathbf{s}, S) = \begin{cases} \mathsf{HVI}_{\mathbf{r}}(\mathbf{s}, S) & \text{if } \mathbf{EPF}_{S,\mathbf{r}} \not\prec \mathbf{f}(\mathbf{s}) \\ -d_{\mathbf{r}}(\mathbf{s}, S) & \text{if } \mathbf{EPF}_{S,\mathbf{r}} \prec \mathbf{f}(\mathbf{s}) \end{cases} , \tag{7.6}$$

where $d_{\mathbf{r}}(\mathbf{s}, S) = \inf_{y \in \mathbf{EPF}_{S,\mathbf{r}}} d(\mathbf{f}(\mathbf{s}), y)$ is the distance between an objective vector $\mathbf{f}(\mathbf{s}) \in \mathbb{R}^m$ and the empirical non-domination front of the set S defined as in (7.1).

We define the fitness $\Phi_{\mathsf{UHVI},X^{\neg i}}(x)$ for a search point $x \in \mathbb{R}^n$ with respect to other solutions in $X^{\neg i}$ as

$$\Phi_{\mathsf{UHVI},X^{\neg i}}(x) = \mathsf{UHVI}_{\mathbf{r}}(x, X^{\neg i}) \ . \tag{7.7}$$

Note that $\Phi_{\mathsf{UHVI},X^{\neg i}}$ is continuous on the empirical non-domination front where both the hypervolume improvement and the considered distance are zero.

Figure 7.2 illustrates this fitness for one non-dominated and two dominated search points (blue plusses) with respect to a set of six other search points (black crosses). The right-hand column of Figure 7.1 shows the level sets of this fitness. The newly introduced hypervolume improvement and distance based fitness Φ_{UHVI} shows smooth level sets, both in search and in objective space. Maybe most importantly, in the dominated area, the fitness function's descent direction (perpendicular to its level sets) now points towards the gaps in the current Pareto front approximation.

7.3.2 Iteratively Optimizing the Φ_{UHVI} Fitness: The Sofomore Framework

After we have discussed a fitness assignment that looks worth to optimize, we come back to our initial idea of subspace optimization and define the underlying algorithmic framework behind Sofomore.

At first, we consider a *single-objective* optimizer in an abstract manner as an iterative algorithm with state $\theta \in \Theta_n$ updated as $\theta_{t+1} = G^f(\theta_t, U_{t+1})$ where $f : \mathbb{R}^n \to \mathbb{R}$ is the single-objective function optimized by the optimizer and U_{t+1} encodes possible random variables sampled within one iteration if we consider a randomized algorithm (and can be taken as constant in the case of a deterministic optimizer). The transition function G^f contains all updates done within the algorithm in one iteration.

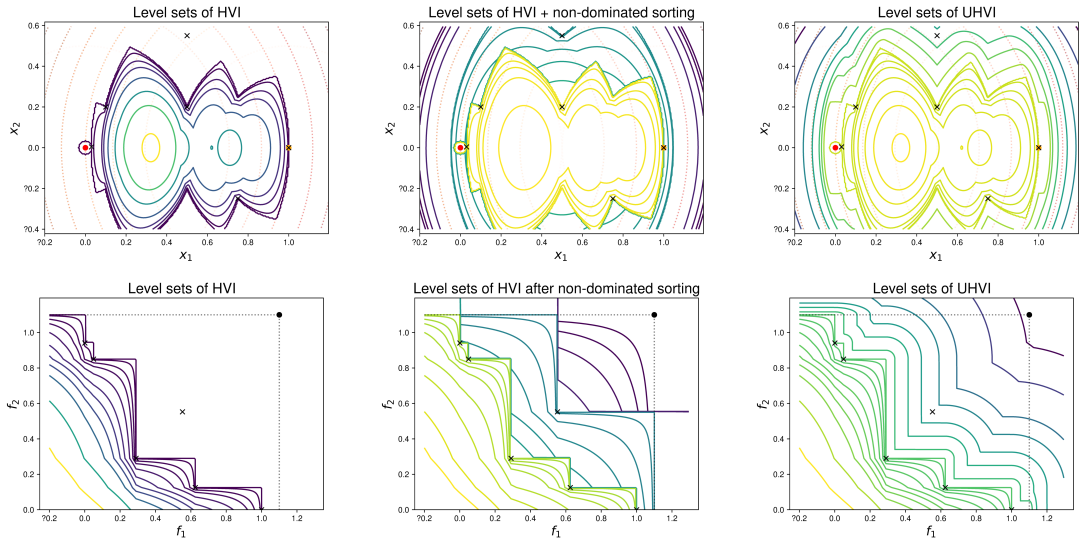


Figure 7.1: Comparison of block-coordinate-wise fitness functions on the double sphere problem. Given a fixed set of six solutions, $\{[0.5, 0.2], [0.75, -0.25], [0.1, 0.2], [1, 0], [0.03, 0.004], [0.5, 0.55]\}$, the above three plots show level sets of equal fitness for a new search point in the search space, the second row shows the same level sets in objective space. Left: standard hypervolume improvement HVI. Middle: HVI within the local non-dominated fronts. Right: newly proposed hypervolume improvement (if non-dominated) together with distance to the non-dominated front (if dominated), denoted as uncrowded hypervolume improvement UHVI. Note that the colors for the fitness levels are not comparable over indicators, but are the same for a given indicator in both search and objective space. The search space plots further show the single-objective's level sets as dotted lines. The black dot indicates the reference point of $[1.1, 1.1]$.

We assume that in each iteration t, the optimizer returns a best estimate of the optimum, often called incumbent solution or recommendation. This is the solution that the optimizer would return if we stop it at iteration t. We denote this incumbent as $\mathcal{E} : \theta \in \Theta_n \longmapsto \mathcal{E}(\theta) \in \mathbb{R}^n$—mapping the state of the algorithm to the estimate of the optimum given this state.

The overall idea behind the subspace optimization and the Sofomore framework can then be formalized as in Algorithm 9: after initializing p single-objective algorithms with their states $\theta_1, \ldots, \theta_p$ and denoting their transition functions as G_i^f $(1 \leq i \leq p)$, we consider their incumbents or recommendations $\mathcal{E}(\theta_i)$ as the p search points that are expected to approximate the optimal p-distribution.

In each step of the Sofomore framework, we choose one of the algorithms (denoted by its number i, with $1 \leq i \leq p$) and run it τ_i iterations on the fitness $\Phi_{\text{UHVI},X^{(\neg i)}}$ to

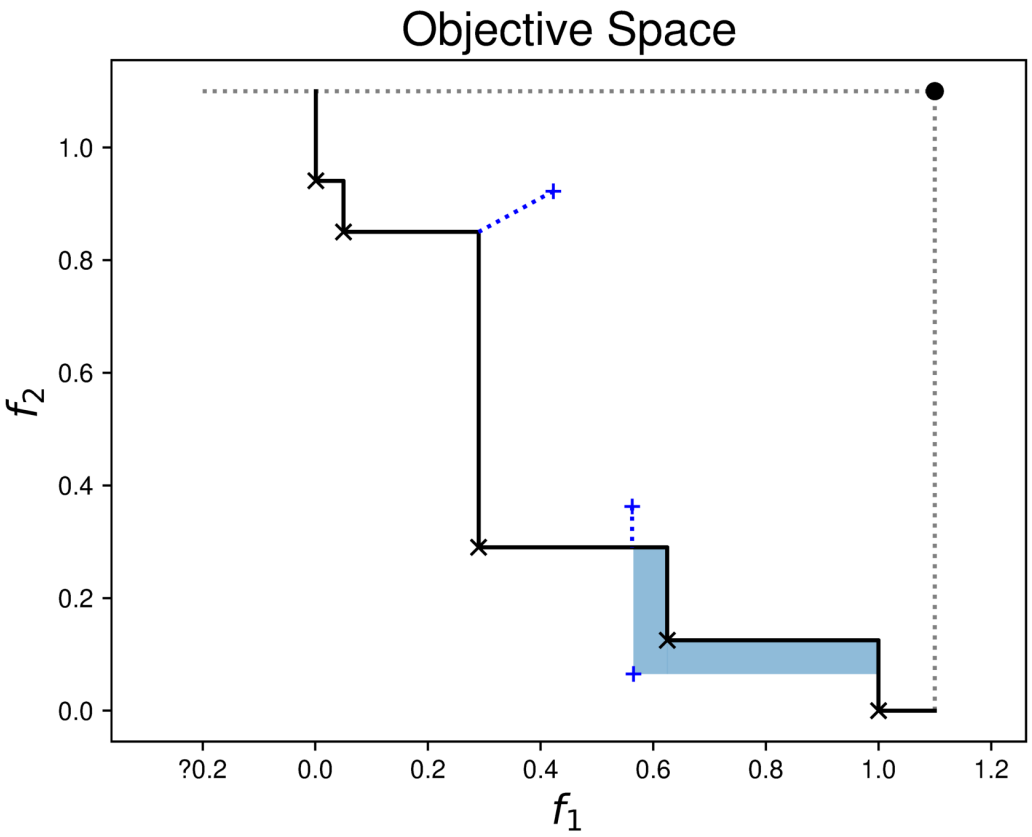


Figure 7.2: Illustration of the proposed UHVI fitness for three points in objective space (blue $+$). The top two points are dominated by the given set $\mathcal{S}$ of five points (black crosses) and are thus assigned a fitness of their distance to the empirical non-dominated set while the bottom point is non-dominated and thus gets assigned the hypervolume improvement with respect to $\mathcal{S}$.

update the recommendation $x^{(i)}$ while keeping all other recommendations fixed. It is important to note that the fitness used for algorithm i is actually changing *dynamically* with the optimization because it depends on all the other incumbents but $x^{(i)}$ which, over time, are expected to move towards the Pareto set as well.

Algorithm 9 proposes a generic framework where the order in which the single-objective algorithms are run and the number of iterations for them are not explicitly defined. A simple strategy would be to choose the algorithms at random or in a given, fixed order and run each single-objective algorithm a fixed number of time steps. But also more elaborate strategies can be envisioned, for example based on the idea of multi-armed bandits [42]: we can log the changes in the fitness value of each incumbent over time and favor as the next chosen algorithms the ones that give the highest expected fitness improvements. Note also that the single-objective algorithms'

types may be different such that we can combine local with global algorithms or even change the algorithms over time, allow restarts etc. In the following experimental validation of our concept, however, we choose a single optimization algorithm and a simple, random strategy to choose which of them to run next.

Algorithm 9 General Sofomore Framework, with fitness of (7.7)

Given: the initial states of p single-objective optimizers, $\theta_1, \ldots, \theta_p$
Initialize incumbents: $X = \{x^{(1)} = \mathcal{E}(\theta_1), \ldots, x^{(p)} = \mathcal{E}(\theta_p)\}$
while not stopping criterion met **do**
    Choose i in $\{1, \ldots, p\}$ and $\tau_i \in \mathcal{N}$
    REPEAT τ_i times:
        $\theta_i \leftarrow G_i^{\Phi_{\text{UHVI},X^{(\neg i)}}}(\theta_i, U_i)$ ▷ run ith algo on fitness $\Phi_{\text{UHVI},X^{(\neg i)}}$
    $x^{(i)} \leftarrow \mathcal{E}(\theta_i)$ ▷ update $x^{(i)}$ in X
end while
return $x^{(1)}, \ldots, x^{(p)}$

With a simple change, Algorithm 9 can be made parallelizable (resulting in slightly different search dynamics though): postponing the updates of the $x^{(i)}$ after every algorithm has been touched at least once makes the optimization of the fitness functions independent such that they can be performed in parallel.

Relation of Sofomore with other existing algorithms We briefly discuss how some existing algorithms and algorithm frameworks relate to the new Sofomore proposal.

The coupling of single-objective algorithms to form a multiobjective one has been done before, especially in the MOEA/D framework [201]. In MOEA/D, p static search directions (in objective space) are defined via p (single-objective) scalarizing functions. Each of them is optimized in parallel with solutions potentially shared between neighboring search directions. On the contrary, the fitness in Sofomore is dynamic, depending on the other incumbents. Optimizing a set of scalarizing functions in classical approaches to multiobjective optimization have static optimization problems to solve without any interaction between them [141].

Many other EMO algorithms, such as NSGA-II, SMS-EMOA, or MO-CMA-ES are not covered by the Sofomore framework. One simple reason is that the UHVI is newly defined.

The already mentioned Newton algorithm on the hypervolume indicator fitness of [99] is probably the closest existing approach from Sofomore, but [99] needs to initialize the Newton algorithm with a set of non-dominated solutions in order for the algorithm to optimize due to the flat regions of its objective space. Also algorithms for

expensive multiobjective optimization based on the optimization of the expected hypervolume improvement [194] can be seen as related to Sofomore, although the proposal of new solutions in algorithms like SMS-EGO [152] or S-metric based ExI [64] use Gaussian Processes to model the objective function. These algorithms, in contrary to Sofomore, propose iteratively a single solution based on the *expected* hypervolume improvement over all known solutions and do not aim at replacing successively a single recommendation by another (better) one. Interesting to note is that algorithms like SMS-EGO and S-metric based ExI employ the expected hypervolume indicator improvement as fitness while the approach of Keane [120] "uses the Euclidean distance to the nearest vector in the Pareto front" [194].

7.4 COMO-CMA-ES

In this section, we instantiate Sofomore with the CMA-ES as single objective optimizer.

Regarding the choice of which optimization algorithm to run (and how long), we opt for a simple strategy: we sample a permutation from S_p, the set of all permutations on $\{1, \ldots, p\}$ uniformly at random and use this fixed permutation to touch each algorithm i once in the order of the permutation. Once all algorithms have been touched, we then resample a new permutation. We run each algorithm for a single iteration. Letting the algorithms run for a too long period right from the start seems suboptimal. As the fitness is dynamic, we do not need to optimize it too precisely. We mainly have two requirements for the choice of single objective optimizers: (i) an optimization algorithm has to be stoppable at any iteration and resumable thereafter and (ii) an optimization algorithm needs to be able to give a good recommendation about the best estimate of the optimum, given its current state. The Covariance Matrix Adaptation Evolution Strategy (CMA-ES, [95]) is a natural choice. Not only is it a state-of-the-art algorithm for difficult blackbox optimization problems but also does it fulfill our requirements. In CMA-ES, the state of the algorithm θ is composed of a step size σ and the parameters of a multivariate normal distribution, namely a mean vector $\mathbf{m}$ representing the favorite solution and a covariance matrix $\mathbf{C}$. In addition, two n-dimensional evolution paths speed up step-size and covariance matrix adaptation. For each θ_i, the incumbent solution $x^{(i)} = \mathcal{E}(\theta_i)$ is the mean $\mathbf{m}$ of the CMA algorithm. A convenient implementation of CMA-ES is via the ask and tell interface [51], where the ask function returns λ candidate solutions and the tell function updates the state from their fitness values. The interface allows to easily stop and resume the optimization and to integrate the dynamic fitness of Sofomore, see Algorithm 10. We call this instantiation of the Sofomore framework COMO-CMA-ES. The p CMA-ES instances are called kernels.

We see in particular how CMA-ES is integrated into Sofomore via its ask-and-

Algorithm 10 The COMO-CMA-ES: an instance of the Sofomore framework with the CMA-ES as single-objective optimizer

1: **Required:**
2: objective function $\mathbf{f} = (\mathbf{f}_1, \ldots, \mathbf{f}_m)$ in dimension n
3: lower and upper bounds for each variable $\texttt{lower} \in \mathbb{R}^n$ and $\texttt{upper} \in \mathbb{R}^n$ of a region of interest
4: number of desired solutions p
5: global initial step-size σ_0 for all CMA-ES
6: fixed reference point $\mathbf{r}$ for the hypervolume indicator
7: **Initialization:**
8: $x^{(i)} = \text{uniformSample}(\texttt{lower}, \texttt{upper})$ for all $1 \leq i \leq p$
9: evaluate all $x^{(i)}$ on $\mathbf{f}$ and store the $\mathbf{f}(x^{(i)})$ for later use
10: $\text{es}^{(i)} \leftarrow (\mu/\mu, \lambda)$-CMA-ES $\left(x^{(i)}, \sigma_0\right)$ for all $1 \leq i \leq p$
11: **while not** stopping criterion **do**
12: sample uniformly at random a permutation π from all permutations on $\{1, \ldots, p\}$
13: **for** i = 1 to p **do**
14: $\{s^1, \ldots, s^\lambda\} \leftarrow \text{es}^{(\pi(i))}.\text{ask}()$ ▷ get λ offspring from ▷ $\pi(i)$th CMA-ES
15: compute the fitness $\Phi(s^j) = \Phi_{\text{UHVI},X^{(\neg\pi(i))}}(s^j)$ for all $1 \leq j \leq \lambda$
16: $\text{es}^{(\pi(i))}.\text{tell}(\Phi(s^1), \ldots, \Phi(s^\lambda))$
17: $x^{(\pi(i))} \leftarrow \text{es}^{(\pi(i))}.\text{mean}$
18: update the stored objective vector $\mathbf{f}(x^{(\pi(i))})$
19: **end for**
20: **end while**
21: **Return** $x^{(1)}, \ldots, x^{(p)}$

tell interface. After choosing the next kernel i, the corresponding CMA-ES instance samples λ solutions ("ask"). It then evaluates them on the uncrowded hypervolume improvement based fitness defined in Eq (7.7)—given all other kernels being fixed. After sorting the λ solutions with respect to their fitness, COMO-CMA-ES feeds the sampled points with their fitness values back to the CMA-ES instance ("tell") which updates all its internal algorithm parameters. Finally, the new mean of the corresponding CMA-ES instance updates the list of the COMO-CMA-ES's proposed p solutions. Note here that CMA-ES is usually not evaluating the mean of the sample distribution which therefore is done in line 18.

7.5 Experimental Validation

We present in this section numerical experiments of the COMO-CMA-ES. Though, in principle, the algorithm can be defined for any number of m objectives, we present results only for $m = 2$. We use the `pycma` Python package [89] version $2.6.0$ for CMA-ES as single-objective optimizer without further parameter tuning.

7.5.1 Test Functions and Performance Measures

For a matrix P and two vectors x and y, we denote

$$\textbf{Quad}(P, x, y) = (x - y)^\top P\,(x - y)\,. \tag{7.8}$$

We also denote by $\mathbf{0}$ the all-zeros vector, $\mathbb{1}$ the all-ones vector, and e_k the unit vector with its only nonzero value at position k. Starting from a positive diagonal matrix Δ, and two independent orthogonal matrices O_1 and O_2, we consider the classes of bi-objective convex quadratic problems **Sep-k**, **One** and **Two** defined as follows [183]

- $\mathbf{f}^{\textbf{sep-}k}_{1,\Delta}(x) = \frac{\textbf{Quad}(\Delta, x, \mathbf{0})}{\textbf{Quad}(\Delta, \mathbf{0}, e_k)}$, $\mathbf{f}^{\textbf{sep-}k}_{2,\Delta}(x) = \frac{\textbf{Quad}(\Delta, x, e_k)}{\textbf{Quad}(\Delta, \mathbf{0}, e_k)}$.
- $\mathbf{f}^{\textbf{one}}_{1,\Delta}(x) = \frac{\textbf{Quad}(O_1^T \Delta O_1, x, \mathbf{0})}{\textbf{Quad}(O_1^T \Delta O_1, \mathbf{0}, \mathbb{1})}$, $\mathbf{f}^{\textbf{two}}_{2,\Delta}(x) = \frac{\textbf{Quad}(O_1^T \Delta O_1, x, \mathbb{1})}{\textbf{Quad}(O_1^T \Delta O_1, \mathbf{0}, \mathbb{1})}$
- $\mathbf{f}^{\textbf{two}}_{1,\Delta}(x) = \frac{\textbf{Quad}(O_1^T \Delta O_1, x, \mathbf{0})}{\alpha}$, $\mathbf{f}^{\textbf{two}}_{2,\Delta}(x) = \frac{\textbf{Quad}(O_2^T \Delta O_2, x, \mathbb{1})}{\alpha}$
  with $\alpha = \max\big(\textbf{Quad}(O_1^T \Delta O_1, \mathbf{0}, \mathbb{1}), \textbf{Quad}(O_2^T \Delta O_2, \mathbf{0}, \mathbb{1})\big)$.

If Δ is the identity matrix, we call the problems as **sphere-sep-k** in the first case and **bi-sphere** in the second and third cases (the rotations are ineffective). If $\Delta(i,i) = 10^{6\frac{i-1}{n-1}}$ for $i = 1, \ldots, n$, then we denote the problems as **elli-sep-k**, **elli-one** or **elli-two**. If $\Delta(1,1) = 10^{-4}$, $\Delta(2,2) = 10^4$ and $\Delta(i,i) = 1$ for $i = 3, \ldots, n$, then we have **cigtab-sep-k**, **cigtab-one** or **cigtab-two**.

We fix the reference point to $\mathbf{r} = (1.1, 1.1)$. The scalings above ensure that the reference point is dominated by all Pareto fronts considered, and that the **Sep-k** and the **One** problems have the same Pareto front (see [183]) than the bi-sphere $g : [0,1] \ni t \longmapsto (1 - \sqrt{t})^2 \in \mathbb{R}$. Note that the expression does not depend on the dimension n.

We use two performance measurements in each run of an algorithm. First the **convergence gap** defined as the difference between an offset called `hv_max` and the hypervolume of the p points $\{x^{(1)}, \ldots, x^{(p)}\}$ found by the algorithm (in case of COMO-CMA-ES or of the population for the other algorithms tested) called `hv`; and second the **archive gap** defined as the difference between an offset called `hvarchive_max` and the hypervolume of all non-dominated points found by the algorithm called `hvarchive`. The setting of `hv_max` is done for each problem as the maximum hypervolume value

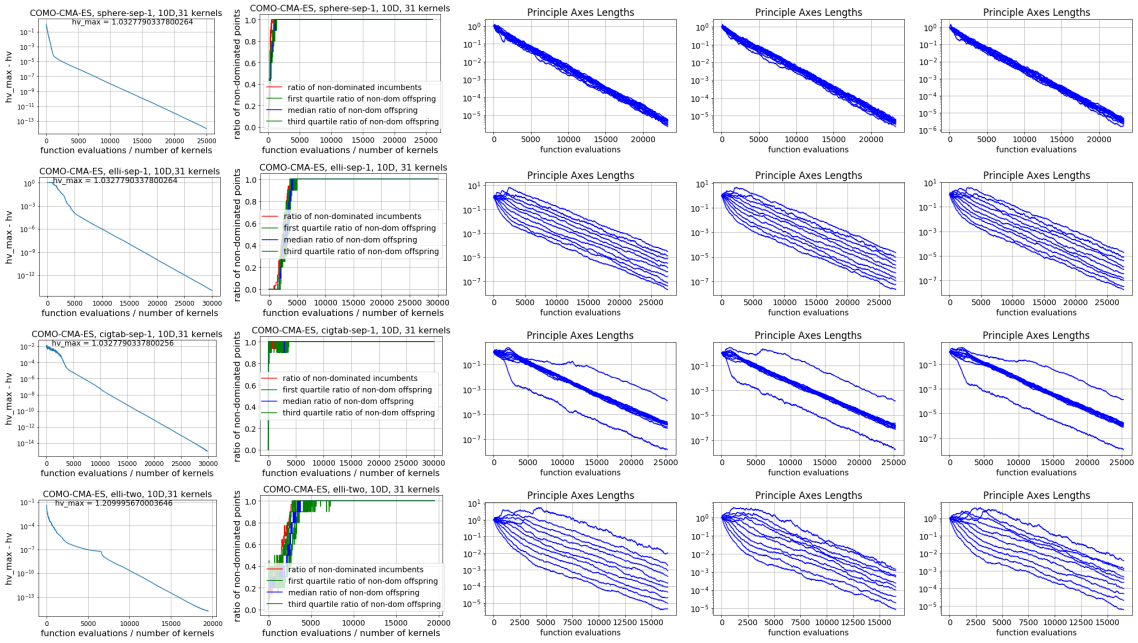


Figure 7.3: Convergence of COMO-CMA-ES on **sphere-sep-1** (first row), **elli-sep-1** (second row), **cigtab-sep-1** (third row) and **elli-two** in 10D with 31 kernels. The first column represents the convergence gap. The second column is the ratio of non-dominated points among the kernels incumbents (red) and the quartiles of the 31 ratios of non-dominated points among each kernel's incumbent and its offspring (median in blue and the remaining quartiles in green). And the last three columns are square root eigenspectra of uniform randomly chosen 3 among the 31 kernels' covariance matrices.

of p kernels found so far anytime the problem was optimized in our machines, plus a small number ($< 10^{-14}$). For the **Sep-k** and the **One** problems, we take `hvarchive_max` as $1.1^2 - \int_0^1 g(t)\mathrm{d}t = 1.21 - \frac{1}{6}$ which corresponds to the hypervolume of the theoretical Pareto front. For the two-class of problems, we use the analytic expression of their Pareto set [183] to sample a large number of points on the Pareto set, and compute their hypervolume as `hvarchive_max`. Thus for the **elli-two** problem in dimension 10, we sample 10^7 points.

7.5.2 Linear convergence of COMO-CMA-ES

We investigate the convergence of COMO-CMA-ES for different dimensions and number of kernels, and display the results on the **sphere-sep-1**, **elli-sep-1**, **cigtab-sep-1** and **elli-two** functions for $n = 10$ and $p = 31$. The global initial step-size is set to $\sqrt{10}$ and the initial lower, upper bounds (line 8 of Algorithm 10) respectively to $-5\mathbb{1}$, $5\mathbb{1}$. In

Figure 7.3, we observe linear convergence in the convergence gap (first column) on all test functions, starting roughly when all displayed ratios of non-dominated points reach 1 (second column). The last three columns of Figure 7.3 illustrate the eigenspectra of the kernels covariance matrices. The first two columns reveal two phases.

First, the kernels incumbents approach the non-dominated region: for **sphere-sep-1** this takes about 1500 evaluations per kernel, for **elli-sep-1**, **cigtab-sep-1** and **elli-two** it takes about 5000, 4000 and 6600 evaluations per kernel. Afterwards, the convergence gap converges linearly. In our settings, there are 11 evaluations per kernel during the update of a kernel, thus for the ***-1** functions (which have the same Pareto set and front), the linear convergence rate is about $10^{-\frac{6\times11}{15000}}$ and for **elli-two**, it is about $10^{-\frac{3\times11}{5000}}$.

For the first 1000 function evaluations per kernel on **elli-sep-1**, there is no point dominating the reference point, which means that the algorithm started far from the Pareto front. Looking at **elli-two**, we confirm that it has a different Pareto front than the three other problems: `hv_max` $= 1.2099...$ (instead of $1.0327...$).

The Uncrowded Hypervolume Improvement depends on other kernels' incumbents and therefore changes in each iteration. Yet, the last three columns are similar to what one would observe when optimizing a single objective convex-quadratic function with corresponding Hessian matrix. After a large enough number of iterations, the probability that the incumbents and their offspring are in the Pareto set becomes close to 1. Then if the incumbents $X = \left\{x^{(1)}, \dots x^{(p)}\right\}$ is a subset of the Pareto set and x is non-dominated, Eq (7.7) becomes: $\Phi_{\mathsf{UHVI},X^{\neg i}}(x) = \mathsf{UHVI}_{\mathbf{r}}(x, X^{\neg i}) = \mathsf{HVI}_{\mathbf{r}}(x, X^{\neg i})$. Its Hessian on smooth bi-objective problems is $\nabla^2\Phi_{\mathsf{UHVI},X^{\neg i}}(x) = \left(\mathbf{f}_2(x) - \mathbf{f}_2(x^{(i-1)})\right)\nabla^2\mathbf{f}_1(x) + \left(\mathbf{f}_1(x) - \mathbf{f}_1(x^{(i+1)})\right)\nabla^2\mathbf{f}_2(x) + \nabla\mathbf{f}_2(x)\nabla\mathbf{f}_1(x)^\top + \nabla\mathbf{f}_1(x)\nabla\mathbf{f}_2(x)^\top$. For our test functions, it is a linear combination of the single objectives Hessian matrices, up to a rank-one matrix and its transpose (the gradients are colinear on the Pareto set of bi-objective convex quadratic problems [183]). That might give a glimpse on the behavior seen in the last three columns of Figure 7.3.

7.5.3 Comparing COMO-CMA-ES with MO-CMA-ES, NSGA-II and SMS-EMOA

We compare four multiobjective algorithms: COMO-CMA-ES, MO-CMA-ES [104], NSGA-II [56] and SMS-EMOA [30], by testing them on classes of bi-objective convex-quadratic problems. We draw once and for all one rotation for **elli-one** in $10D$ and two different rotations for **elli-two** in $10D$. The Simulated Binary Crossover operator (SBX) and the polynomial mutation are used for NSGA-II (run with the `evoalgos` package [196]) and SMS-EMOA (run with the Matlab version by Fabian Kretzschmar and Tobias Wagner [195]): we use a crossover probability of 0.7 and a mutation probability of 0.1, and the distribution indexes for crossover and mutation operators are both

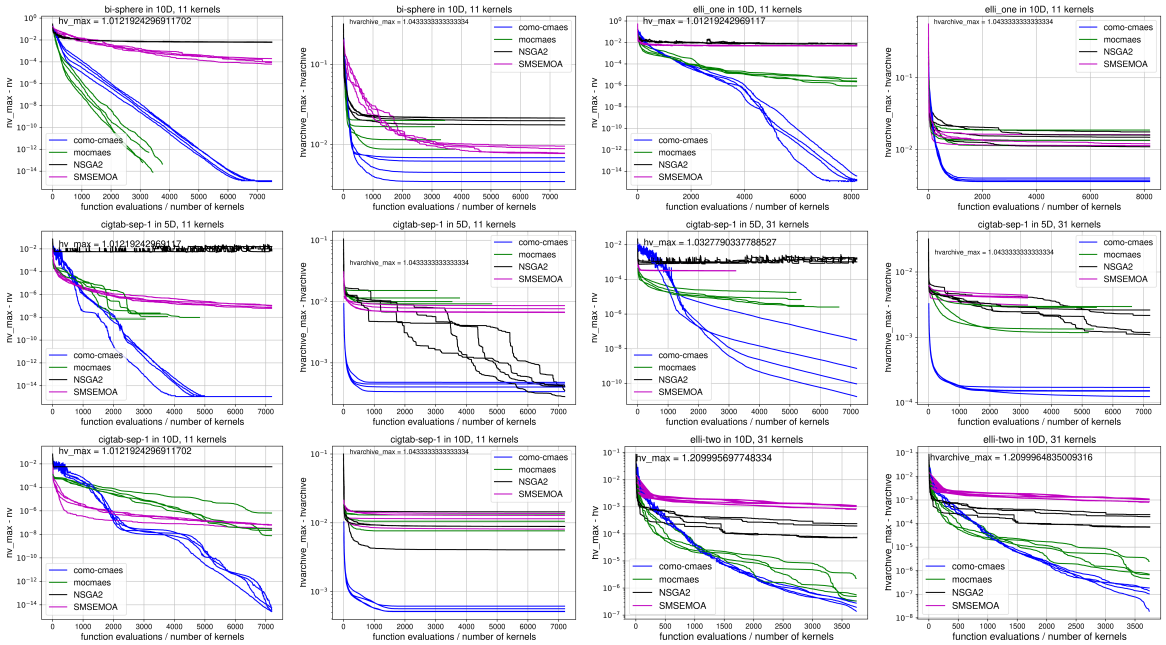


Figure 7.4: Convergence gaps (odd columns) and archive gaps (even columns) for **bi-sphere**, **elli-one**, **cigtab-sep-1** and **elli-two**. Each algorithm is run 4 times, in 5D or 10D, with 11 or 31 kernels. The random matrices are drawn from the same seed in all the algorithms.

equal to 10. We use the version of MO-CMA-ES from [193]. The number of kernels for COMO-CMA-ES corresponds to the population size of the other algorithms, that we set to either 11 or 31, and the dimensions considered are 5 and 10. The global initial step-size of COMO-CMA-ES is set to 0.2 with initial lower, upper bounds (line 8 of Algorithm 10) set to the all-zeros and all-ones vectors. The initial population for the three other algorithms is sampled uniformly at random in $[0, 1]^n$.

We run each multiobjective optimization 4 times and display the convergence gap (of the population or the incumbent solutions of the kernels) and the archive gap.

In Figure 7.4, the values of the convergence gap reached by COMO-CMA-ES and MO-CMA-ES are several orders of magnitude lower than for the two other algorithms.

On the 5-dimensional **bi-sphere**, COMO-CMA-ES and MO-CMA-ES appear to show linear convergence, where the latter appears to be about 30% faster than the former. On the **cigtab-sep-1** function, COMO-CMA-ES is initially slow, but catches up after about 1000 evaluations per kernel. In all other cases, COMO-CMA-ES shows superior performance for the convergence gap. On the 10-dimensional **cigtab-sep-1**, COMO-CMA-ES shows a plateau between 2000 and 4000 evaluations per kernel. This kind of plateau cannot be observed in the MO-CMA-ES and the observed final convergence speed is better for COMO-CMA-ES than for MO-CMA-ES. The observed

plateau is typical for the behavior of non-elitist multi-recombinative CMA-ES on the tablet function, because CSA barely reduces an initially large step-size before the tablet-shape has been adapted, which is related to the neutral subspace defect found in [129]. Elitism as in the MO-CMA-ES, on the other hand, also helps to decrease an initially too large step-size.

Although COMO-CMA-ES was not designed to perform well on the archive gap, it shows consistently the best results over all experiments. Only on the **cigtab-sep-1** in $5D$ with 11 kernels, NSGA-II reaches and slightly surpasses the archive gap of COMO-CMA-ES after 7000 function evaluations per kernel. This suggests, as expected from the known dependency between optimal step-size and population size [90], that COMO-CMA-ES adds valuable diversity while approaching the optimal p-distribution of the Pareto front at the same time.

7.6 Conclusions

We have proposed (i) the Sofomore framework to define multiobjective optimizers from single-objective ones, (ii) a fitness for dominated solutions to be the distance to the empirical Pareto front (Uncrowded Hypervolume Improvement UHVI) and (iii) the non-elitist "comma" CMA-ES to instantiate the framework (COMO-CMA-ES). We observe that COMO-CMA-ES converges linearly towards the p-optimal distribution of the hypervolume indicator on several bi-objective convex quadratic problems. The COMO-CMA-ES appears to be robust to independently rotating the Hessian matrices of convex-quadratic problems, even if such rotations transform the Pareto set from a line segment to a bent curve. In our limited experiments, COMO-CMA-ES performed generally better than MO-CMA-ES, SMS-EMOA and the NSGA-II, w.r.t. convergence gap and archive gap while COMO-CMA-ES was solely designed to optimize the convergence gap. We conjecture that the advantage on the archive gap is due to (i) the large stationary variance obtained with non-elitist evolution strategies and (ii) the fitness assignment of dominated solutions which favors the vacant (uncrowded) space between non-dominated solutions and hence serves as implicit crowding distance penalty measure.

Chapter 8

Comparing COMO-CMA-ES to well-known multiobjective evolutionary algorithms

Contents

Note: the content of this chapter is presented in [61]:
Paul Dufosse and Cheikh Toure, *Benchmarking MO-CMA-ES and COMO-CMA-ES on the Bi-objective bbob-biobj Testbed*, Proceedings of the Genetic and Evolutionary Computation Conference Companion, 2019, pp 1920–1927.

In this paper, we propose a comparative benchmark of MO-CMA-ES, COMO-CMA-ES (recently introduced in [185]) and NSGA-II, using the COCO framework for performance assessment and the Bi-objective test suite `bbob-biobj`. For a fixed number of points p, COMO-CMA-ES approximates an optimal p-distribution of the Hypervolume Indicator. While not designed to perform on archive-based assessment, $i.e.$ with respect to all points evaluated so far by the algorithm, COMO-CMA-ES behaves well on the COCO platform. The experiments are done in a true Black-Blox spirit by using a

minimal setting relative to the information shared by the 55 problems of the `bbob-biobj` Testbed.

8.1 Introduction

As the COCO platform assesses bi-objective algorithms by computing the covered Hypervolume of evaluated points in a black-box optimization framework, it is natural that well-performing benchmarked algorithms on COCO are designed to tackle these two issues (covering the front and black-box paradigm). Hence [128] uses a hybrid algorithm and parameter tuning to handle the black-box difficulty, and [136] develops an unbounded population size variant of MO-CMA-ES [104, 193] for the Pareto front covering issue.
MO-CMA-ES and COMO-CMA-ES, however, are non-hybrid algorithms with a population size p (number of kernels for COMO-CMA-ES) fixed *a priori*.

8.2 Algorithms Description

NSGA-II is a popular choice among Evolutionary Multi-Objective (EMO) algorithms [56], based on non dominated sorting of candidates and crowding distance comparisons. It has already been benchmarked on the `bbob-biobj` Testbed [17]. The archive data, $i.e.$ the data for all points evaluated so far, is shown under the name NSGA-II along with the best2016 reference, as a baseline for the reader.

The elitist Multiobjective CMA-ES, MO-CMA-ES [104], is based on a two-way ranking: the Pareto ranking and the hypervolume contribution among individuals with the same Pareto rank. Each parent is not only a point in the search space but a 5-tuple of data representing a (1+1)-CMA-ES, which is a Single Objective (SO) optimizer defined in [104]. We use here the improved step-size adaptation designed in [193]. The algorithm also uses greedy mating, which means that parents are selected only among non-dominated solutions. We choose the default settings for the λ_{MO}-(1+1)-CMA-ES presented in [104]. In this paper, the term population size refers to the MO population size, namely the parameter λ_{MO}.

A recent EMO algorithm using the non-elitist CMA-ES [88] is the Comma Multiobjective CMA-ES (COMO-CMA-ES) defined in [185], and is designed to approximate the optimal p-distribution of the Hypervolume indicator. It is the instantiation of a wider framework called Sofomore, on the non-elitist CMA-ES [185]. We also take the default setup for the standard CMA-ES presented in [88]. The *Single-objective Optimization FOr Optimizing Multiobjective Optimization pRoblEms* framework (Sofomore) finds p points approximating the optimal p-distribution of the Hyper-

volume indicator, by iteratively maximizing an extended notion of the hypervolume contribution (hypervolume improvement) called Uncrowded Hypervolume Improvement UHVI [185]. As the two-way ranking in [104], UHVI unflattens the hypervolume improvement's level sets in dominated regions, but the novelty is inherent to the fact that this unflattening operation still preserves diversity among the solutions, by ensuring them to be uncrowded.

8.3 Implementation and Experimental Procedure

The code for MO-CMA-ES was written in Matlab, 2014, using routines from the Shark library for hypervolume computation. COMO-CMA-ES is implemented in Python and uses NumPy. Note that the hypervolume is computed in pure Python and does not call any NumPy method. Implementation details for NSGA-II can be found in [17].

For MO-CMA-ES, the algorithm starts with random starting point uniformly sampled in $I = [-5, 5]^n$ with an initial stepsize σ_0 equals to 1. From this starting point a population of p (1+1)-CMA-ES is evolved, with p the population size of the algorithm. Any (1+1)-CMA-ES can send a warning, meaning that it has somehow terminated. When any warning is met, the MO algorithm stops. Then a restart is performed with a random starting point in I, with the same settings. The population size is a core parameter fixed *a priori* for each run (including restarts), which is why MO-CMA-ES is benchmarked with different population sizes, namely 10, 32 and 100. The reference point used for computing the contributing hypervolume is iteratively chosen as the nadir point among the current population, and adding 1 to each coordinate.

COMO-CMA-ES also starts with a random point sampled in I, with an initial stepsize $\sigma_0 = \sqrt{n}$. No restart is performed and no parameter tuning is done on purpose to observe the behaviour of the algorithm with default settings. In the same spirit, the only stopping criteria is given by the allocated budget of function evaluations. As COMO-CMA-ES is designed to approximate the optimal p-distribution of the Hypervolume indicator, it is then necessary to fix one and for all a reference point for each run; which is done by setting an attribute of each COCO problem called `1`argest_fvalues_of_interest [81] as reference point.

We benchmark MO-CMA-ES (with population sizes equal to $10, 32$ and 100), COMO-CMA-ES (with 3, 10, 32, 100, 316 and 1000 kernels), and NSGA-II. The latter is run with a population size of 100 as in [17]. COCO is run with the common settings for the `bbob-biobj` test suite; 10 instances with dimension $n \in \{2, 3, 5, 10, 20\}$. The allocated budget is $10^5 n$ function evaluations for each algorithm.

Algorithm	p	2-D	3-D	5-D	10-D	20-D
MO-	10	2.8	3.3	3.0	3.6	3.3
CMA-ES	32	2.7	2.9	2.8	2.9	2.9
	100	3.0	3.2	3.5	3.4	3.0
COMO-	10	5.9	6.0	5.0	4.7	4.8
CMA-ES	32	5.8	5.2	5.8	4.1	4.9
	100	12	8.9	7.7	7.5	4.3

Table 8.1: CPU times per function evaluation of MO-CMA-ES and COMO-CMA-ES (in 10^{-4} seconds) for varying dimensions. p is the population size (or number of kernels).

8.4 CPU Timing

In order to evaluate the duration of each algorithm, we have reported here the timing results when running the full budget experiment. Both codes were run on a linux machine with 64 cores, Intel®Xeon ®E7 to E3 v4 processors. We are interested in CPU time per function evaluation for varying dimensions and population sizes. To account for Matlab internal parallelization we use the function **cputime** and not real timing. The results can be found in Table 8.1.

We are benchmarking algorithms written in different languages, therefore comparisons should be made carefully. COMO-CMA-ES computation per function evaluation takes 1.5 to 4 times longer than MO-CMA-ES, and this ratio is less important when the dimension increases.

8.5 Results

Results from experiments according to [81], [79] and [40] on the benchmark functions given in [187] for the three algorithms called MO-*, NSGA-II or COMO-* (with * being the fixed population size) are presented in Figures 8.1, 8.2, 8.3 and 8.4. In Tables 8.2 and 8.3 we display results for each algorithm only with a population size of 100.

The **average runtime (aRT)** used in the tables depends on a given quality indicator value, $I_{\text{target}} = I_{\text{ref}} + \Delta I_{\text{HV}}^{\text{COCO}}$, and is computed over all relevant trials as the number of function evaluations executed during each trial while the best indicator value did not reach I_{target}, summed over all trials and divided by the number of trials that actually reached I_{target} [81]. **Statistical significance** is tested with the rank-sum test for a given target I_{target} using, for each trial, either the number of needed function evalua-

tions to reach I_{target} (inverted and multiplied by -1), or, if the target was not reached, the best $\Delta I_{\text{HV}}^{\text{COCO}}$-value achieved, measured only up to the smallest number of overall function evaluations for any unsuccessful trial under consideration.

The experiments were performed with COCO [80], version 2.2, the plots were produced with version 2.3.1.

For either COMO-CMA-ES or MO-CMA-ES, one can compare the shapes of the empirical cumulative distribution functions (ECDFs) for different population sizes. Note that for each algorithm and each COCO problem, the function is evaluated first with a zero-vector, since we know this produces a point with better f-values than a random sampled point. This allows rigorous comparison of the different algorithms ECDFs.

A glance on the data allows to surmise the following: the larger the population size, the better the reached target precisions are in the long run. However for small budgets, algorithms with smaller population size reach better target precisions. For example in Figure 8.1 on function **f28**, MO-10 is the first to solve 35% of the targets, MO-32 is the first to solve 45% for a larger budget, and finally MO-100 has the best overall performance with more than 50% of the targets solved. This result is not new and a MO-CMA-ES adapting the population size has been studied for example in [133]. It is still interesting to note that this tendency appears for almost all functions in the test suite.

The same comment can be made on the same function **f28** for COMO-CMA-ES with 3, 10 and 32 kernels: among the COMO-CMA-ES variants, COMO-3 is the first one to solve 40% of the targets, COMO-10 the first to solve 55%, and COMO-32 the first to solve 70%. Note that COMO-32 performs well at a budget of $10^4 n$, compared to the other COMO-CMA-ES. And for a budget of $10^5 n$, COMO-100 and COMO-316 are systematically better. We can expect the variants with more kernels (1000 is proposed here) to solve more targets for longer runs, where the ones with smaller number of kernels will stagnate sooner.

There are some functions where MO-CMA-ES outperforms the best 2016 data for small budgets, say up to $10^2 n$, see **f7, f17, f27, f32** on Figure 8.1. In contrast the COMO-CMA-ES' ECDFs dynamics are different: compared to the MO-CMA-ES with the same population size, it solves less targets on small budgets; there is a longer-lasting starting phase. This is partly due to the elitist nature of the (1+1)-CMA-ES used by MO-CMA-ES and the non-elitist nature of the standard CMA-ES used by COMO-CMA-ES. However COMO-CMA-ES performs better in the long run, for instance with a budget of $10^4 n$ on functions **f2, f4, f19, f53** in Figures 8.1, 8.2. Note that this effect (starting phase duration and fraction of targets solved) increases with the number of kernels. Remark that the MO-CMA-ES algorithms perform particularly well on problems where one of the functions is the Gallagher 101 (**f40, f45, f49, f52, f54, f55**) where a MO-* systematically outperforms the best2016 reference.

In 5-D on Figure 8.3, if we look at the fraction of targets found for a budget of

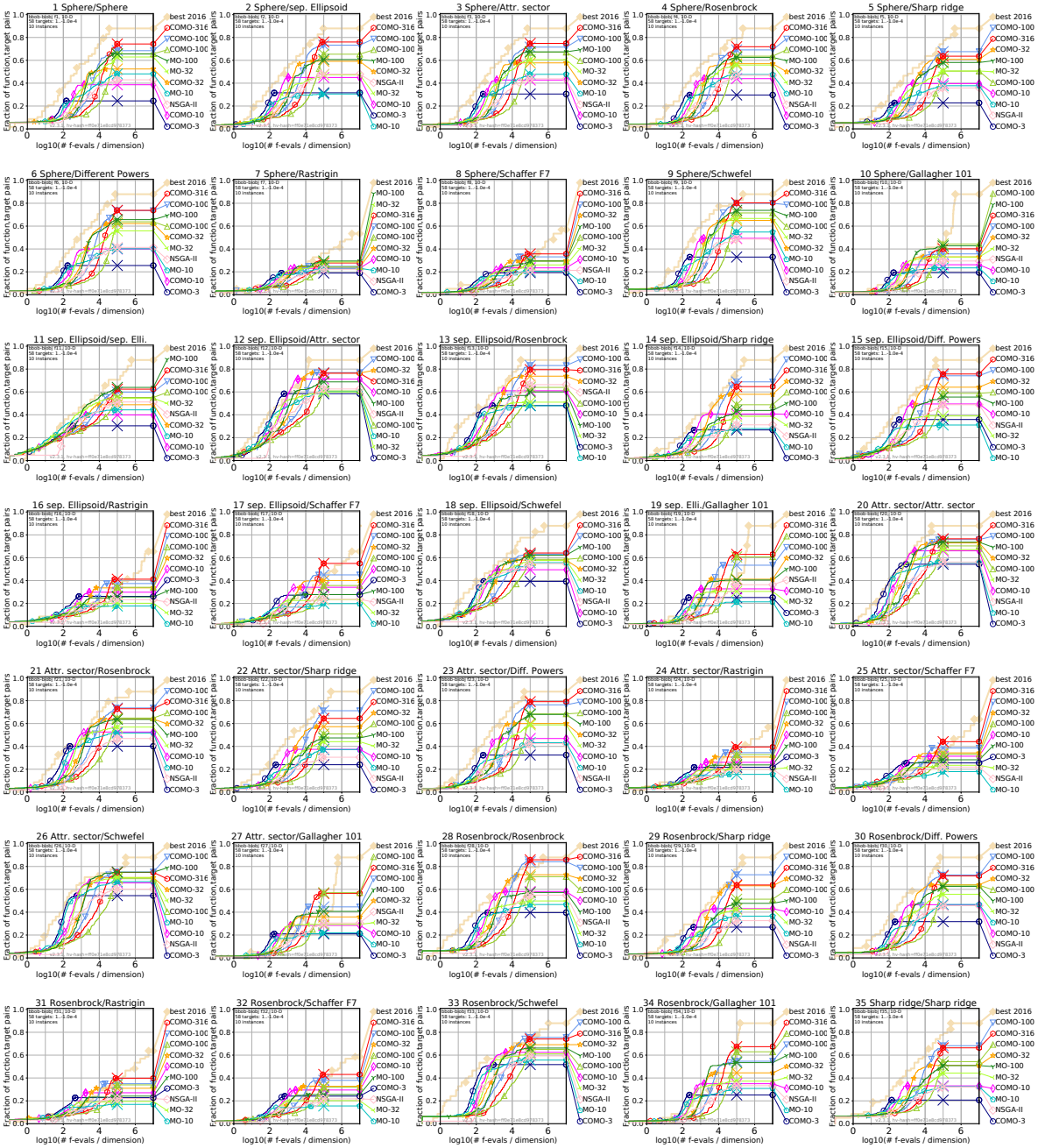


Figure 8.1: Empirical cumulative distribution of simulated (bootstrapped) runtimes, measured in number of objective function evaluations, divided by dimension (FEvals/DIM) for the 58 targets $\{-10^{-4}, -10^{-4.2}, -10^{-4.4}, -10^{-4.6}, -10^{-4.8}, -10^{-5}, 0, 10^{-5}, 10^{-4.9}, 10^{-4.8}, \ldots, 10^{-0.1}, 10^{0}\}$ in dimension 10.

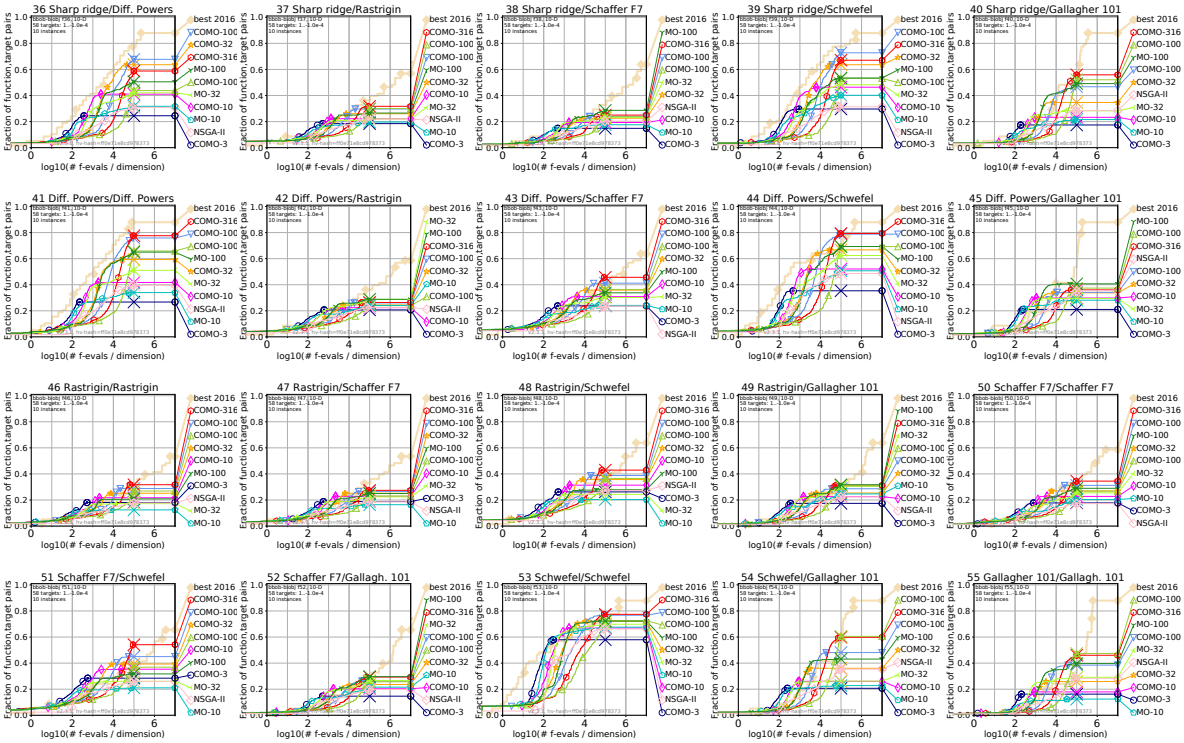


Figure 8.2: Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/DIM) as in Fig. 8.1 but for functions f_{36} to f_{55} in 10-D.

$10^4 n$, a global statement is that NSGA-II is ranking below either MO-100 or COMO-100 (which have the same population size), but it can be above the variants with a population size far from 100. In 20-D, as we can see on Figure 8.4, for a budget of $10^4 n$ and considering all subgroups of functions, COMO-100 solves more targets than MO-100 which in turn performs better than NSGA-II. As an example, on **ill-cond, ill-cond** subgroup, COMO-100 solves 50% of the target precisions versus roughly 40% for MO-100 and 20% for the NSGA-II. This kind of gap appears in most subgroups with one ill conditioned objective function. COMO-3 and COMO-10 also perform better than best2016 on subgroups with one multimodal objective function for small budgets, between $10^2 n$ and $10^4 n$.

8.6 Conclusion

We have benchmarked COMO-CMA-ES and MO-CMA-ES with different population sizes to particularly test the influence of this parameter on the performances. We also used NSGA-II as a baseline. COMO-CMA-ES performs well although it is *a priori* designed to approximate the optimal p-distribution of the Hypervolume Indicator, for

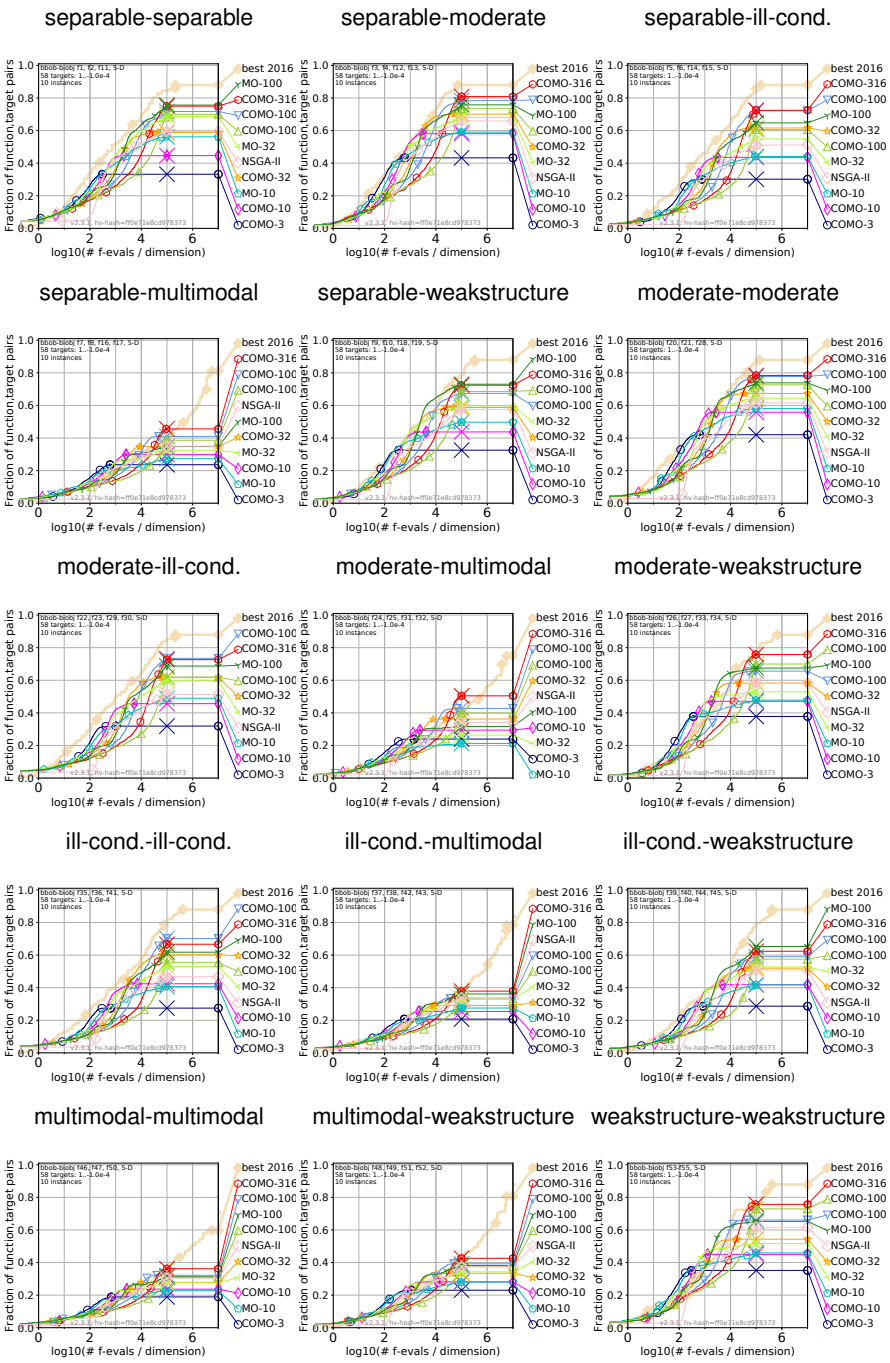


Figure 8.3: Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/DIM) for 58 targets with target precision in $\{-10^{-4}, -10^{-4.2}, -10^{-4.4}, -10^{-4.6}, -10^{-4.8}, -10^{-5}, 0, 10^{-5}, 10^{-4.9}, 10^{-4.8}, \ldots, 10^{-0.1}, 10^{0}\}$ for all functions and subgroups in 5-D.

p the population size (number of kernels). As observed in [185], this performance is mainly due to the the large stationary variance obtained with non-elitist (comma) evolution strategies and with the use of UHVI which guides the evolution towards the uncrowded space in the non-dominated region. The overall results for MO-CMA-ES and COMO-CMA-ES show that a smaller population size performs better for smaller budget, while a larger population size end up performing better for a budget large enough. Hence as done in [136] with MO-CMA-ES, a population size adaptation for COMO-CMA-ES should guarantee better performance on COCO.

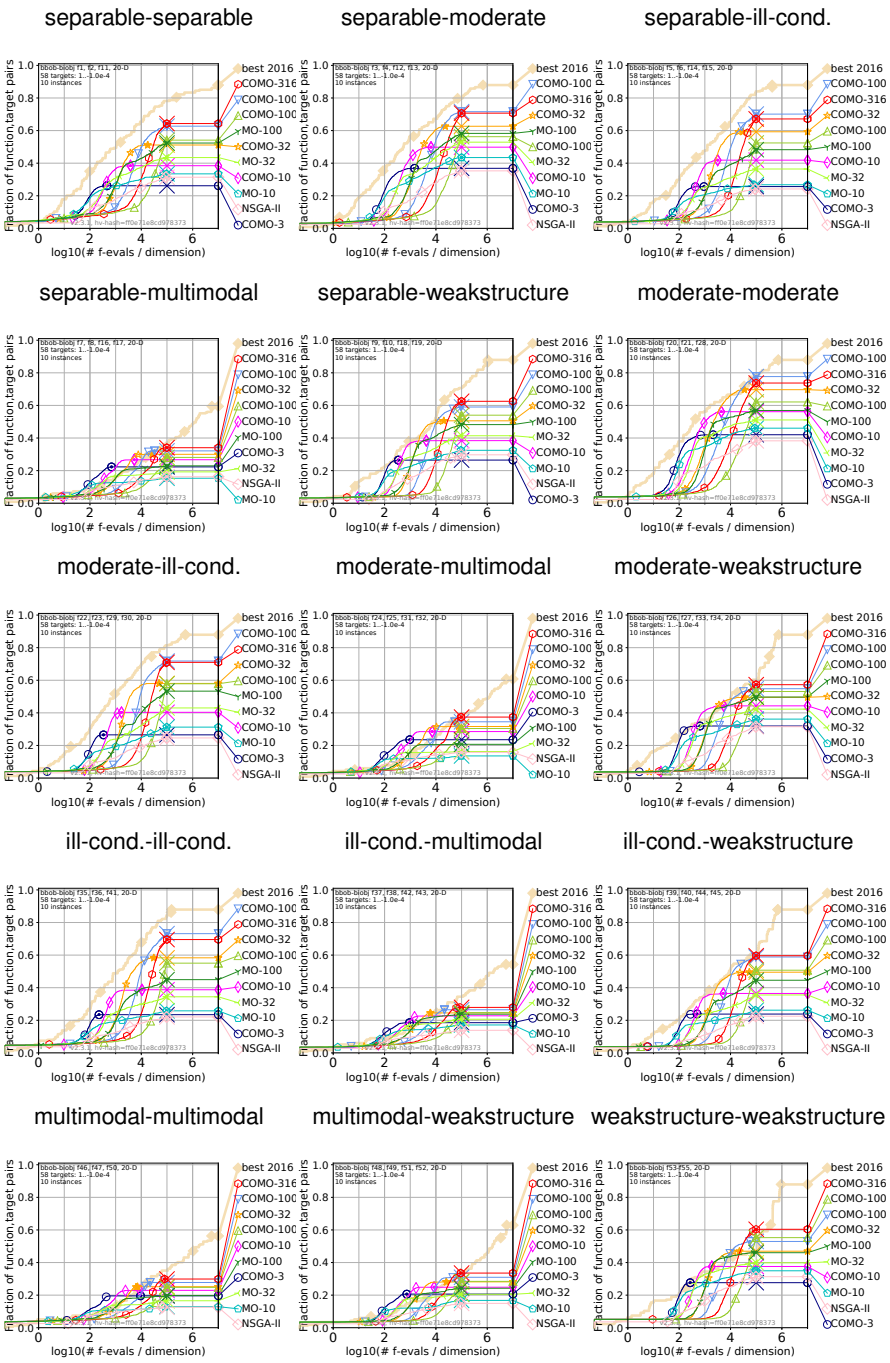


Figure 8.4: Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/DIM) for 58 targets with target precision in $\{-10^{-4}, -10^{-4.2}, -10^{-4.4}, -10^{-4.6}, -10^{-4.8}, -10^{-5}, 0, 10^{-5}, 10^{-4.9}, 10^{-4.8}, \ldots, 10^{-0.1}, 10^{0}\}$ for all functions and subgroups in 20-D.

Δf	1e0	1e-1	1e-2	1e-3	#succ
f1	1	75	584	3660	10/10
COMO	**1.2**(0)	20(14)	20(2)	8.7(0.2)	0/10
MO	3.2(6)	11(16)	12(2)	**4.0**(0.1)	0/10
NSGA	39(188)	14(2)	11(5)	35(6)	0/10
f2	5.0	105	601	3715	10/10
COMO	5.1(4)	34(52)	22(11)	8.7(3)	0/10
MO	10(22)	17(15)	11(4)	**3.7**(0.9)	0/10
NSGA	63(83)	11(2)	**4.1**(2)	36(16)	0/10
f3	3.0	115	665	5170	10/10
COMO	3.1(6)	21(7)	20(11)	6.5(3)	0/10
MO	**2.8**(6)	11(15)	10(3)	**2.8**(0.8)	0/10
NSGA	89(132)	22(20)	14(36)	93(111)	0/10
f4	2.0	109	571	2669	10/10
COMO	4.8(0)	15(8)	20(6)	11(1)	0/10
MO	**1.6**(0)	10(11)	12(3)	5.0(0.5)	0/10
NSGA	171(178)	10(2)	13(0.8)	46(63)	0/10
f5	2.0	120	1277	21889	10/10
COMO	**1.1**(3)	27(8)	17(2)	**3.4**(1)	0/10
MO	**1.2**(2)	16(12)	7.4(1)	3.7(4)	0/10
NSGA	86(76)	11(2)	19(14)	∞ *5e5*	0/10
f6	3.0	73	688	3976	10/10
COMO	3.5(0.2)	34(26)	20(8)	8.5(2)	0/10
MO	12(28)	27(22)	11(4)	**3.7**(0.8)	0/10
NSGA	104(122)	18(8)	11(8)	114(66)	0/10
f7	2.0	2763	1.2e5	3.5e6	0/10
COMO	**1.9**(0)	3.3(5)	6.6(9)	∞ *5e5*	0/10
MO	**1.5**(2)	**1.7**(1)	5.2(14)	∞ *5e5*	0/10
NSGA	80(198)	3.2(2)	**4.2**(10)	∞ *5e5*	0/10
f8	3.0	2167	1.6e5	2.1e6	0/10
COMO	19(43)	4.7(2)	**1.9**(2)	∞ *5e5*	0/10
MO	**1.8**(2)	**2.6**(1)	**1.3**(1)	∞ *5e5*	0/10
NSGA	142(44)	**1.0**(0.2)	4.6(3)	∞ *5e5*	0/10
f9	4.0	96	521	1986	10/10
COMO	7.9(2)	18(15)	22(3)	14(1)	0/10
MO	14(38)	14(2)	12(5)	6.3(0.5)	0/10
NSGA	67(92)	25(73)	11(24)	30(10)	0/10
f10	4.0	323	9839	52107	10/10
COMO	16(48)	13(5)	**2.2**(0.3)	11(5)	0/10
MO	15(6)	8.4(6)	**0.96**(0.1)	**4.4**(14)	0/10
NSGA	95(111)	10(26)	6.1(5)	10(9)	0/10
f11	5.0	56	436	9189	9/10
COMO	4.5(12)	10(37)	12(22)	3.1(2)	0/10
MO	**1.8**(0.6)	5.4(12)	4.3(3)	**1.5**(2)	0/10
NSGA	28(23)	11(5)	3.2(0.8)	5.7(11)	0/10
f12	5.0	44	641	3991	10/10
COMO	**1**(0.8)	29(64)	11(26)	**4.6**(5)	6/10
MO	5.0(2)	16(7)	4.6(9)	**7.3**(21)	4/10
NSGA	80(102)	23(24)	11(24)	36(63)	5/10
f13	7.0	60	560	5582	10/10
COMO	6.2(4)	15(16)	11(9)	3.6(2)	6/10
MO	7.6(15)	5.5(6)	5.4(4)	**1.5**(0.5)	0/10
NSGA	65(40)	15(6)	9.4(28)	10(3)	3/10
f14	5.0	247	1713	12801	10/10
COMO	**0.54**(0.4)	47(12)	17(2)	**5.4**(3)	0/10
MO	**0.60**(0.5)	36(4)	202(147)	108(78)	0/10
NSGA	66(60)	13(17)	45(50)	378(537)	0/10
f15	6.0	74	677	6559	10/10
COMO	10(21)	59(72)	24(16)	5.8(1)	0/10
MO	17(56)	30(45)	11(4)	**2.7**(1)	0/10
NSGA	80(93)	32(25)	11(17)	28(28)	0/10

Δf	1e0	1e-1	1e-2	1e-3	#succ
f20	4.0	70	1162	5724	10/10
COMO	33(80)	26(26)	11(5)	**6.5**(4)	2/10
MO	15(35)	17(29)	5.2(4)	**3.7**(5)	2/10
NSGA	761(106)	63(9)	12(5)	153(201)	0/10
f21	5.0	86	3249	9924	10/10
COMO	**3.4**(8)	21(23)	3.5(3)	**3.3**(3)	3/10
MO	8.7(15)	14(18)	**1.5**(1)	**1.1**(0.9)	1/10
NSGA	56(80)	13(6)	19(3)	20(31)	1/10
f22	3.0	97	1168	12608	9/10
COMO	16(35)	43(62)	22(10)	**5.5**(3)	0/10
MO	10(5)	21(25)	13(25)	14(36)	0/10
NSGA	41(102)	16(5)	25(23)	∞ *5e5*	0/10
f23	1	59	618	4410	10/10
COMO	**1.9**(0)	21(25)	21(12)	**7.5**(2)	0/10
MO	**2.2**(0)	15(24)	10(5)	**3.7**(2)	0/10
NSGA	202(318)	23(7)	11(4)	143(133)	0/10
f24	5.0	2347	1.8e5	4.1e6	0/10
COMO	**1.3**(3)	4.0(3)	**0.36**(0.1)	∞ *5e5*	0/10
MO	**2.0**(4)	**2.1**(1)	27(27)	∞ *5e5*	0/10
NSGA	71(25)	5.2(1)	5.7(8)	∞ *5e5*	0/10
f25	9.0	3143	1.2e5	2.5e6	3/10
COMO	14(35)	**2.8**(3)	**0.99**(0.4)	**0.36**(0.5)	0/10
MO	**0.81**(3)	**1.5**(2)	3.3(6)	∞ *5e5*	0/10
NSGA	47(87)	10(29)	7.2(7)	∞ *5e5*	0/10
f26	7.0	59	2182	13673	8/10
COMO	13(30)	21(13)	5.6(4)	5.9(9)	0/10
MO	11(7)	13(13)	**2.0**(2)	10(18)	0/10
NSGA	88(76)	31(29)	**1.5**(2)	7.5(19)	0/10
f27	6.0	2631	21971	44576	10/10
COMO	27(124)	**1.2**(2)	**0.83**(0.5)	5.5(9)	0/10
MO	15(37)	**0.97**(0.9)	**0.36**(0.1)	**3.1**(3)	2/10
NSGA	386(61)	5.5(6)	12(24)	15(12)	0/10
f28	2.0	20	145	1230	9/10
COMO	7.4(1)	36(35)	35(23)	16(5)	8/10
MO	**1.9**(6)	18(6)	25(11)	7.4(0.3)	0/10
NSGA	95(63)	46(7)	16(9)	27(32)	0/10
f29	3.0	114	1413	13660	10/10
COMO	41(201)	54(29)	18(6)	**5.8**(2)	0/10
MO	12(29)	33(18)	36(70)	55(50)	0/10
NSGA	140(118)	36(6)	29(32)	∞ *5e5*	0/10
f30	1	33	366	2294	10/10
COMO	**2.5**(8)	41(62)	34(14)	13(3)	2/10
MO	**1.2**(0.5)	37(45)	17(7)	6.3(1)	0/10
NSGA	155(177)	92(147)	42(71)	71(148)	0/10
f31	3.0	2166	50028	3.2e6	0/10
COMO	7.4(10)	4.2(3)	**1.1**(0.4)	∞ *5e5*	0/10
MO	4.8(16)	**2.2**(1)	90(217)	∞ *5e5*	0/10
NSGA	71(117)	7.0(14)	10(18)	∞ *5e5*	0/10
f32	3.0	2131	1.1e5	2.4e6	1/10
COMO	5.3(11)	**2.7**(2)	**0.53**(0.2)	∞ *5e5*	0/10
MO	**1.1**(2)	**1.4**(0.9)	7.0(4)	∞ *5e5*	0/10
NSGA	73(70)	5.1(11)	8.2(9)	∞ *5e5*	0/10
f33	6.0	627	1214	3730	7/10
COMO	10(41)	90(200)	50(108)	20(36)	2/10
MO	22(68)	89(1.0)	48(2)	**17**(1)	1/10
NSGA	64(67)	90(399)	47(104)	18(37)	1/10

Δf	1e0	1e-1	1e-2	1e-3	#succ
f38	4.0	4366	2.7e5	5.4e6	1/10
COMO	**1.0**(0.6)	4.6(2)	17(13)	∞ *5e5*	0/10
MO	**2.5**(2)	**1.8**(0.5)	**1.9**(3)	∞ *5e5*	0/10
NSGA	88(71)	**2.2**(2)	5.2(4)	∞ *5e5*	0/10
f39	2.0	215	1160	47472	8/10
COMO	5.0(3)	19(7)	22(7)	**1.9**(0.8)	0/10
MO	**1.9**(0.2)	12(7)	87(126)	10(11)	0/10
NSGA	95(144)	6.8(1)	19(17)	31(35)	0/10
f40	2.0	998	34442	2.0e5	8/10
COMO	**2.9**(2)	9.3(2)	**2.6**(7)	11(8)	0/10
MO	**1.8**(2)	4.6(0.8)	**0.41**(0.1)	**1.4**(2)	0/10
NSGA	161(169)	**2.7**(1)	4.1(9)	∞ *5e5*	0/10
f41	2.0	48	708	6343	10/10
COMO	**4.0**(4)	53(63)	20(10)	5.5(2)	0/10
MO	5.3(9)	41(42)	11(4)	**2.3**(0.4)	0/10
NSGA	173(185)	29(16)	12(16)	81(144)	0/10
f42	2.0	2525	3.4e5	4.3e6	0/10
COMO	**3.9**(13)	26(103)	13(13)	∞ *5e5*	0/10
MO	7.0(16)	**2.2**(2)	**2.9**(2)	∞ *5e5*	0/10
NSGA	220(230)	4.3(15)	**2.8**(4)	∞ *5e5*	0/10
f43	4.0	2468	1.5e5	3.8e6	2/10
COMO	**2.2**(3)	3.1(2)	**1.8**(1.0)	∞ *5e5*	0/10
MO	7.1(14)	**1.4**(1)	**1.4**(0.6)	∞ *5e5*	0/10
NSGA	65(99)	6.8(3)	4.3(4)	∞ *5e5*	0/10
f44	6.0	86	513	1853	10/10
COMO	23(46)	34(17)	28(20)	17(6)	1/10
MO	21(1)	20(27)	13(8)	**7.7**(4)	0/10
NSGA	80(113)	14(9)	7.4(2)	34(65)	0/10
f45	4.0	816	5730	53653	10/10
COMO	3.7(0.4)	5.8(4)	14(2)	23(24)	0/10
MO	6.9(15)	3.2(2)	**1.6**(0.2)	**2.8**(5)	0/10
NSGA	53(62)	**2.3**(0.8)	**10**(5)	90(77)	0/10
f46	4.0	11925	4.9e5	5.6e7	0/10
COMO	**3.0**(2)	**2.3**(1)	**2.7**(5)	∞ *5e5*	0/10
MO	**3.5**(6)	5.4(0.4)	∞	∞ *5e5*	0/10
NSGA	88(65)	3.5(4)	∞	∞ *5e5*	0/10
f47	3.0	4172	2.7e5	4.4e6	0/10
COMO	**1.2**(2)	4.6(3)	**7.7**(9)	∞ *5e5*	0/10
MO	**0.73**(0.7)	**1.6**(0.9)	8.9(15)	∞ *5e5*	0/10
NSGA	67(105)	6.8(14)	17(17)	∞ *5e5*	0/10
f48	9.0	2160	1.1e5	2.1e6	2/10
COMO	8.8(30)	6.7(9)	**2.3**(4)	∞ *5e5*	0/10
MO	6.7(14)	**1.9**(1)	39(17)	∞ *5e5*	0/10
NSGA	35(36)	11(52)	**2.7**(4)	∞ *5e5*	0/10
f49	9.0	3737	2.0e5	1.3e6	1/10
COMO	10(2)	18(35)	10(16)	∞ *5e5*	0/10
MO	**1.6**(3)	**1.5**(0.8)	**4.5**(5)	∞ *5e5*	0/10
NSGA	42(50)	7.9(15)	5.2(4)	**3.5**(5)	0/10
f50	6.0	3913	2.2e5	2.6e6	1/10
COMO	**0.93**(2)	3.7(2)	**1.4**(2)	∞ *5e5*	0/10
MO	**1.2**(2)	**1.4**(1)	**1.0**(0.7)	∞ *5e5*	0/10
NSGA	39(68)	**2.4**(3)	5.8(6)	∞ *5e5*	0/10
f51	8.0	1251	32465	2.5e6	1/10
COMO	3.3(1)	5.4(5)	**1.7**(0.8)	**0.86**(1)	1/10
MO	**2.5**(6)	**3.0**(3)	**3.7**(4)	∞ *5e5*	0/10
NSGA	53(50)	**2.6**(5)	11(13)	∞ *5e5*	0/10

Δf	1e0	1e-1	1e-2	1e-3	#succ
f16	3.0	2810	2.0e5	4.2e6	3/10
COMO	**0.63**(0.5)	**2.5**(2)	**0.27**(0.1)	**0.50**(1)	1/10
MO	**2.0**(5)	**1.6**(1)	6.0(8)	*∞ 5e5*	0/10
NSGA	135(134)	4.0(14)	6.5(9)	*∞ 5e5*	0/10
f17	29	2935	48624	2.4e6	3/10
COMO	51(127)	3.3(5)	**2.2**(3)	**0.34**(0.4)	0/10
MO	12(0.7)	**1.7**(4)	25(21)	*∞ 5e5*	0/10
NSGA	16(23)	3.4(1)	10(19)	**2.0**(2)	0/10
f18	2.0	56	557	6912	8/10
COMO	4.2(8)	5.6(5)	7.6(2)	4.7(3)	0/10
MO	9.0(17)	3.2(5)	3.6(4)	**1.2**(0.5)	0/10
NSGA	158(215)	15(5)	**2.4**(0.5)	41(56)	0/10
f19	9.0	1292	6164	88464	9/10
COMO	4.9(10)	**2.5**(1)	**2.4**(1)	**0.39**(0.2)	1/10
MO	**2.6**(1)	**1.4**(1)	10(0.3)	**0.77**(0.0)	4/10
NSGA	67(22)	18(0.6)	11(18)	**1.8**(1)	0/10

Δf	1e0	1e-1	1e-2	1e-3	#succ
f34	9.0	1376	15575	54195	10/10
COMO	**1.5**(3)	**2.2**(2)	**1.1**(0.4)	**3.0**(5)	0/10
MO	**0.82**(0.7)	**1.4**(1)	**0.51**(0.1)	6.4(14)	2/10
NSGA	21(36)	**1.6**(0.4)	5.2(6)	15(15)	0/10
f35	1	141	2268	51388	8/10
COMO	3.7(2)	28(23)	10(4)	**1.7**(0.3)	0/10
MO	3.9(6)	22(15)	35(50)	14(14)	0/10
NSGA	64(158)	9.5(3)	29(37)	*∞ 5e5*	0/10
f36	2.0	204	4640	41237	10/10
COMO	4.2(1)	32(13)	8.3(3)	**3.0**(1)	0/10
MO	**2.7**(2)	17(6)	4.9(10)	5.4(5)	0/10
NSGA	91(185)	12(14)	20(8)	*∞ 5e5*	0/10
f37	2.0	3956	1.5e5	2.5e6	1/10
COMO	**1**(2)	3.2(2)	**2.0**(2)	*∞ 5e5*	0/10
MO	**0.70**(1)	**1.6**(1)	6.2(5)	*∞ 5e5*	0/10
NSGA	73(121)	**1.0**(0.9)	5.0(3)	*∞ 5e5*	0/10

Δf	1e0	1e-1	1e-2	1e-3	#succ
f52	2.0	3027	1.4e5	2.1e6	1/10
COMO	**2.1**(3)	**2.6**(3)	**1.2**(3)	**2.2**(4)	0/10
MO	3.8(6)	**1.5**(1)	**0.42**(0.3)	**1.0**(2)	0/10
NSGA	133(181)	3.2(6)	3.0(2)	*∞ 5e5*	0/10
f53	2.0	13	42	1443	2/10
COMO	9.1(0)	12(19)	29(53)	42(173)	1/10
MO	6.9(32)	14(11)	21(42)	41(87)	0/10
NSGA	116(229)	62(13)	29(6)	40(88)	1/10
f54	98	1514	12856	45502	10/10
COMO	**2.1**(5)	**1.8**(4)	**1.2**(0.2)	**0.72**(0.2)	0/10
MO	**2.2**(0.1)	**0.94**(1)	**0.61**(0.2)	7.6(11)	0/10
NSGA	23(52)	**2.0**(0.1)	4.5(4)	7.3(7)	0/10
f55	3.0	1415	18086	51245	9/10
COMO	**2.2**(0.9)	5.2(3)	**1.2**(0.5)	7.4(13)	0/10
MO	**1**(1)	**1.6**(1)	**0.51**(0.1)	10(15)	1/10
NSGA	98(121)	7.7(7)	10(8)	27(25)	0/10

Table 8.2: Average runtime (aRT in number of function evaluations) divided by the respective best aRT measured during BBOB-2016 in dimension 5. This aRT ratio and, in braces as dispersion measure, the half difference between 10 and 90%-tile of bootstrapped run lengths appear for each algorithm and target, the corresponding reference aRT in the first row. The different target $\Delta I_{\mathrm{HV}}^{\mathrm{COCO}}$-values are shown in the top row. #succ is the number of trials that reached the (final) target $I_{\mathrm{ref}} + 10^{-5}$. The median number of conducted function evaluations is additionally given in *italics*, if the target in the last column was never reached. Entries, succeeded by a star, are statistically significantly better (according to the rank-sum test) when compared to all other algorithms of the table, with $p = 0.05$ or $p = 10^{-k}$ when the number k following the star is larger than 1, with Bonferroni correction by the number of functions (55). A ↓ indicates the same tested against the best algorithm from BBOB 2016. Best results are printed in bold.
Data produced with COCO v2.3.1, hv-hash=ff0e71e8cd978373

Δf	1e0	1e-1	1e-2	1e-3	#succ
f1	1	157	1468	8244	10/10
COMO	**1**(0)	189(15)	48(2)	**17**(0.4)	0/10
MO	**1**(0)	100(12)	24(0.9)	90(11)	0/10
NSGA	**1**(0)	702(1356)	∞	∞ *2e6*	0/10
f2	5.0	163	2170	21153	10/10
COMO	**2.0**(8)	237(91)	42(8)	**9.2**(2)	0/10
MO	12(30)	90(12)	17(3)	∞ *2e6*	0/10
NSGA	50(94)	654(762)	2459(2e4)	∞ *2e6*	0/10
f3	1	216	2384	11104	10/10
COMO	**1**(0)	163(32)	32(4)	**14**(2)	0/10
MO	**1**(0)	71(6)	16(6)	34(30)	0/10
NSGA	**1**(0)	663(648)	∞	∞ *2e6*	0/10
f4	1	172	1682	10200	9/10
COMO	**1**(0)	210(22)	45(2)	**14**(0.5)	0/10
MO	**1**(0)	91(8)	23(5)	34(9)	0/10
NSGA	**1**(0)	504(408)	1.1e4(9216)	∞ *2e6*	0/10
f5	1	190	3207	32860	10/10
COMO	**1**(0)	245(16)	35(3)	**7.9**(0.9)	0/10
MO	**1**(0)	101(9)	18(2)	72(53)	0/10
NSGA	**1**(0)	1635(650)	∞	∞ *2e6*	0/10
f6	3.0	236	2134	16568	10/10
COMO	**2.2**(10)	150(38)	39(5)	**10**(0.7)	0/10
MO	3.1(7)	64(15)	16(2)	77(22)	0/10
NSGA	72(94)	459(583)	∞	∞ *2e6*	0/10
f7	1	24981	6.8e5	3.8e7	0/10
COMO	**1**(0)	3.7(3)	27(18)	∞ *2e6*	0/10
MO	**1**(0)	**1.1**(0.4)	∞	∞ *2e6*	0/10
NSGA	**1**(0)	326(260)	∞	∞ *2e6*	0/10
f8	4.0	16448	2.4e6	2.8e7	0/10
COMO	162(0.4)	6.0(3)	∞	∞ *2e6*	0/10
MO	81(201)	4.3(8)	∞	∞ *2e6*	0/10
NSGA	27(68)	1164(1216)	∞	∞ *2e6*	0/10
f9	1	144	1468	5944	10/10
COMO	**1**(0)	248(20)	49(2)	**22**(1)	8/10
MO	**1**(0)	105(11)	22(0.9)	36(6)	0/10
NSGA	111(275)	375(294)	654(1946)	∞ *2e6*	0/10
f10	1	6124	2.0e5	2.4e5	10/10
COMO	**1**(0)	9.2(0.9)	**1.7**(0.1)	**20**(29)	0/10
MO	**1**(0)	3.8(0.7)	4.8(12)	22(21)	0/10
NSGA	94(0)	223(303)	∞	∞ *2e6*	0/10
f11	3.0	246	3177	1.1e5	1/10
COMO	**1.3**(2)	89(98)	48(21)	**16**(11)	0/10
MO	**2.9**(13)	14(13)	**8.1**(6)	**13**(10)	0/10
NSGA	84(140)	19(27)	215(256)	∞ *2e6*	0/10
f12	11	131	1483	11435	10/10
COMO	**0.98**(0.6)	61(94)	76(25)	24(8)	0/10
MO	**0.87**(1)	48(21)	13(4)	65(166)	0/10
NSGA	38(40)	142(223)	236(556)	779(1093)	1/10
f13	7.0	139	1087	17899	7/10
COMO	318(3)	197(98)	70(7)	**13**(4)	0/10
MO	45(220)	60(34)	20(2)	136(185)	0/10
NSGA	77(77)	128(232)	782(1800)	∞ *2e6*	0/10
f14	3.0	349	6102	63789	10/10
COMO	**0.63**(2)	180(62)	23(5)	**14**(9)	0/10
MO	**0.63**(0)	66(19)	117(87)	∞ *2e6*	0/10
NSGA	112(157)	1054(3715)	∞	∞ *2e6*	0/10
f15	6.0	210	2099	25094	9/10
COMO	6.0(1)	224(99)	53(5)	**8.7**(2)	0/10
MO	**5.8**(0)	52(12)	37(51)	381(259)	0/10
NSGA	74(81)	192(387)	∞	∞ *2e6*	0/10

Δf	1e0	1e-1	1e-2	1e-3	#succ
f20	3.0	136	1870	8641	10/10
COMO	**0.57**(0)	125(50)	30(14)	26(15)	5/10
MO	**2.3**(0)	70(21)	10(5)	**8.2**(4)	0/10
NSGA	101(159)	56(52)	570(540)	330(442)	0/10
f21	5.0	331	1753	9785	8/10
COMO	39(128)	76(31)	39(12)	**22**(16)	3/10
MO	29(58)	33(11)	14(4)	71(102)	0/10
NSGA	126(559)	84(111)	1993(1148)	∞ *2e6*	0/10
f22	1	313	4469	23797	10/10
COMO	**1**(0)	160(33)	**26**(3)	**11**(6)	0/10
MO	**1**(0)	68(11)	58(27)	119(182)	0/10
NSGA	**1**(0)	6470(6329)	∞	∞ *2e6*	0/10
f23	3.0	332	2660	16975	10/10
COMO	**1.6**(3)	130(24)	37(3)	**13**(15)	0/10
MO	**1.7**(0)	39(10)	**27**(36)	67(50)	0/10
NSGA	110(84)	710(394)	7301(1e4)	∞ *2e6*	0/10
f24	1	61504	8.1e5	6.5e7	0/10
COMO	963(4808)	**1.4**(0.8)	23(38)	∞ *2e6*	0/10
MO	59(0)	5.6(4)	∞	∞ *2e6*	0/10
NSGA	589(202)	293(195)	∞	∞ *2e6*	0/10
f25	1	34160	9.2e5	1.2e7	0/10
COMO	**1**(0)	**1.7**(2)	**1.2**(3)	∞ *2e6*	0/10
MO	**1**(0)	**0.64**(0.5)	∞	∞ *2e6*	0/10
NSGA	193(402)	42(44)	∞	∞ *2e6*	0/10
f26	3.0	136	1117	4425	9/10
COMO	615(3072)	195(98)	56(30)	88(122)	1/10
MO	180(0)	73(23)	19(9)	**11**(13)	0/10
NSGA	53(214)	62(107)	315(914)	307(1260)	0/10
f27	1	60235	1.3e6	6.4e6	7/10
COMO	**2.3**(0)	4.8(0.4)	**2.4**(2)	**1.3**(0.5)	0/10
MO	9.1(40)	8.7(17)	**2.6**(2)	**2.9**(6)	0/10
NSGA	400(568)	58(51)	∞	∞ *2e6*	0/10
f28	1	62	539	6898	8/10
COMO	**1**(0)	387(91)	102(7)	**17**(2)	5/10
MO	**1**(0)	185(34)	45(4)	314(235)	0/10
NSGA	195(506)	441(329)	2960(3264)	∞ *2e6*	0/10
f29	1	359	3841	30739	8/10
COMO	**1**(0)	154(23)	31(2)	**7.2**(0.8)	0/10
MO	**1**(0)	64(9)	48(9)	205(391)	0/10
NSGA	126(0)	1327(503)	∞	∞ *2e6*	0/10
f30	1	220	1777	10156	9/10
COMO	7.4(32)	158(54)	47(7)	**15**(1)	0/10
MO	8.1(36)	57(18)	20(6)	109(131)	0/10
NSGA	183(458)	293(433)	5082(7316)	∞ *2e6*	0/10
f31	3.0	27722	8.6e5	3.0e7	0/10
COMO	92(277)	3.9(1)	**21**(30)	∞ *2e6*	0/10
MO	93(231)	32(36)	∞	∞ *2e6*	0/10
NSGA	45(140)	657(902)	∞	∞ *2e6*	0/10
f32	6.0	32985	1.1e6	2.6e7	0/10
COMO	289(604)	**2.5**(1)	**1.1**(2)	∞ *2e6*	0/10
MO	92(208)	61(76)	∞	∞ *2e6*	0/10
NSGA	99(98)	149(166)	∞	∞ *2e6*	0/10
f33	1	33	278	3348	6/10
COMO	**1**(0)	662(203)	184(41)	47(31)	0/10
MO	**1**(0)	272(56)	69(9)	27(39)	0/10
NSGA	106(262)	578(500)	2412(1712)	∞ *2e6*	0/10

Δf	1e0	1e-1	1e-2	1e-3	#succ
f38	3.0	36705	3.9e6	2.2e8	0/10
COMO	**1.0**(0)	**5.3**(2)	∞	∞ *2e6*	0/10
MO	**0.47**(0.7)	**1.6**(0.5)	∞	∞ *2e6*	0/10
NSGA	64(167)	∞	∞	∞ *2e6*	0/10
f39	1	457	3876	25958	10/10
COMO	**1**(0)	113(16)	30(3)	**8.5**(3)	0/10
MO	**1**(0)	48(9)	99(136)	115(39)	0/10
NSGA	**1**(0)	2584(2559)	∞	∞ *2e6*	0/10
f40	1	26134	2.1e5	2.6e6	7/10
COMO	**1**(0)	**2.3**(0.4)	**7.0**(10)	**1.9**(2)	0/10
MO	**1**(0)	**1.0**(0.2)	13(25)	∞ *2e6*	0/10
NSGA	**1**(0)	144(241)	∞	∞ *2e6*	0/10
f41	2.0	245	2389	15818	10/10
COMO	42(207)	144(37)	37(4)	**10**(1.0)	1/10
MO	31(72)	49(15)	14(2)	1198(1770)	0/10
NSGA	107(252)	456(1001)	∞	∞ *2e6*	0/10
f42	1	34578	2.9e6	2.3e7	0/10
COMO	**1**(0)	3.3(2)	∞	∞ *2e6*	0/10
MO	**1**(0)	**0.70**(0.2)	∞	∞ *2e6*	0/10
NSGA	191(432)	149(111)	∞	∞ *2e6*	0/10
f43	2.0	55957	2.3e6	3.9e7	0/10
COMO	**1.3**(0.9)	**1.4**(0.5)	**2.1**(3)	∞ *2e6*	0/10
MO	4.4(3)	**0.90**(0.5)	∞	∞ *2e6*	0/10
NSGA	122(214)	162(274)	∞	∞ *2e6*	0/10
f44	3.0	168	1485	6584	10/10
COMO	**31**(48)	200(65)	52(10)	**21**(3)	4/10
MO	56(267)	69(31)	25(7)	84(101)	0/10
NSGA	93(130)	261(971)	1477(3126)	∞ *2e6*	0/10
f45	1	49797	2.4e6	5.2e6	7/10
COMO	**1.1**(0.2)	**1.3**(0.8)	**0.88**(0.8)	**3.5**(4)	0/10
MO	4.1(0)	4.9(0.2)	**1.3**(0.8)	∞ *2e6*	0/10
NSGA	170(188)	32(40)	∞	∞ *2e6*	0/10
f46	1	55141	1.0e6	1.1e8	0/10
COMO	**1**(0)	**2.8**(0.8)	∞	∞ *2e6*	0/10
MO	**1**(0)	16(9)	∞	∞ *2e6*	0/10
NSGA	81(0)	∞	∞	∞ *2e6*	0/10
f47	2.0	54618	2.2e6	5.2e7	0/10
COMO	**0.60**(0.2)	3.3(0.6)	∞	∞ *2e6*	0/10
MO	**2.5**(10)	42(64)	∞	∞ *2e6*	0/10
NSGA	45(112)	331(220)	∞	∞ *2e6*	0/10
f48	1	22301	3.5e5	3.3e7	0/10
COMO	**1**(0)	3.6(2)	**23**(15)	∞ *2e6*	0/10
MO	**1**(0)	**1.1**(0.5)	∞	∞ *2e6*	0/10
NSGA	91(0)	220(184)	∞	∞ *2e6*	0/10
f49	2.0	1.6e5	1.1e7	7.1e7	0/10
COMO	**1.1**(3)	**0.84**(0.4)	∞	∞ *2e6*	0/10
MO	3.5(15)	**1.3**(1)	∞	∞ *2e6*	0/10
NSGA	52(128)	113(107)	∞	∞ *2e6*	0/10
f50	3.0	22169	1.3e6	2.0e7	0/10
COMO	**0.67**(0.8)	4.8(2)	**6.7**(14)	∞ *2e6*	0/10
MO	7.3(32)	26(25)	∞	∞ *2e6*	0/10
NSGA	72(101)	∞	∞	∞ *2e6*	0/10
f51	1	8574	9.9e5	2.2e7	0/10
COMO	**1**(0)	7.4(3)	**0.28**(0.1)	∞ *2e6*	0/10
MO	**1**(0)	29(118)	∞	∞ *2e6*	0/10
NSGA	269(243)	566(1279)	∞	∞ *2e6*	0/10

Δf	1e0	1e-1	1e-2	1e-3	#succ
f16	6.0	32242	1.0e6	1.9e7	0/10
COMO	**1.1**(2)	3.5(3)	**2.4**(3)	∞ *2e6*	0/10
MO	**1.3**(2)	7.9(0.8)	∞	∞ *2e6*	0/10
NSGA	85(106)	84(17)	∞	∞ *2e6*	0/10
f17	1	33890	9.3e5	2.7e7	0/10
COMO	**1**(0)	**2.4**(1)	**1.7**(3)	**0.69**(0.7)	0/10
MO	**1**(0)	40(30)	∞	∞ *2e6*	0/10
NSGA	227(70)	139(208)	∞	∞ *2e6*	0/10
f18	6.0	141	1507	17541	10/10
COMO	22(94)	221(84)	54(21)	**17**(4)	0/10
MO	**1.8**(0.5)	52(24)	12(3)	70(84)	0/10
NSGA	62(87)	27(17)	54(18)	∞ *2e6*	0/10
f19	7.0	18889	2.4e5	5.1e6	6/10
COMO	**1.7**(0.0)	3.3(0.7)	**1.5**(2)	3.6(3)	0/10
MO	3.5(8)	6.9(0.4)	20(19)	∞ *2e6*	0/10
NSGA	69(114)	42(29)	∞	∞ *2e6*	0/10

Δf	1e0	1e-1	1e-2	1e-3	#succ
f34	3.0	45588	1.9e5	2.3e5	8/10
COMO	477(1192)	**1.2**(0.2)	**5.2**(5)	36(50)	0/10
MO	95(238)	**0.48**(0.1)	3(27)	41(37)	0/10
NSGA	49(121)	17(22)	∞	∞ *2e6*	0/10
f35	1	338	4782	41429	10/10
COMO	**1**(0)	127(17)	**22**(2)	**5.2**(0.6)	0/10
MO	**1**(0)	60(8)	88(42)	∞ *2e6*	0/10
NSGA	**1**(0)	1968(1326)	∞	∞ *2e6*	0/10
f36	1	454	4824	48159	10/10
COMO	**1**(0)	142(35)	30(5)	**6.8**(3)	0/10
MO	**1**(0)	48(6)	61(106)	∞ *2e6*	0/10
NSGA	**1**(0)	4570(2112)	∞	∞ *2e6*	0/10
f37	1	10528	5.2e5	3.5e7	0/10
COMO	**1**(0)	10(5)	**∞**	∞ *2e6*	0/10
MO	**1**(0)	**3.3**(1)	**∞**	∞ *2e6*	0/10
NSGA	**1**(0)	∞	**∞**	∞ *2e6*	0/10

Δf	1e0	1e-1	1e-2	1e-3	#succ
f52	1	82166	4.9e6	2.4e7	1/10
COMO	**1**(0)	**1.5**(0.5)	∞	∞ *2e6*	0/10
MO	**1**(0)	**0.80**(0.1)	**3.7**(5)	∞ *2e6*	0/10
NSGA	39(96)	240(280)	∞	∞ *2e6*	0/10
f53	1	28	159	1952	8/10
COMO	**1**(0)	649(131)	250(24)	46(5)	6/10
MO	**1**(0)	234(90)	94(4)	13(1)	0/10
NSGA	**1**(0)	123(160)	147(120)	124(84)	0/10
f54	7.0	6049	1.6e5	2.5e6	8/10
COMO	293(1143)	8.5(1)	5.9(9)	**2.0**(3)	0/10
MO	59(122)	3.3(0.5)	7.0(9)	∞ *2e6*	0/10
NSGA	66(118)	155(208)	∞	∞ *2e6*	0/10
f55	1	8.4e5	8.8e6	8.8e6	7/10
COMO	**1**(0)	**0.37**(1)	**0.93**(0.8)	**2.1**(3)	0/10
MO	**1**(0)	**1.1**(3)	**0.92**(0.4)	**2.1**(2)	0/10
NSGA	**1**(0)	∞	∞	∞ *2e6*	0/10

Table 8.3: Average runtime (aRT in number of function evaluations) divided by the respective best aRT measured during BBOB-2016 in dimension 20. This aRT ratio and, in braces as dispersion measure, the half difference between 10 and 90%-tile of bootstrapped run lengths appear for each algorithm and target, the corresponding reference aRT in the first row. The different target $\Delta I_{\mathrm{HV}}^{\mathrm{COCO}}$-values are shown in the top row. #succ is the number of trials that reached the (final) target $I_{\mathrm{ref}} + 10^{-5}$. The median number of conducted function evaluations is additionally given in *italics*, if the target in the last column was never reached. Entries, succeeded by a star, are statistically significantly better (according to the rank-sum test) when compared to all other algorithms of the table, with $p = 0.05$ or $p = 10^{-k}$ when the number k following the star is larger than 1, with Bonferroni correction by the number of functions (55). A ↓ indicates the same tested against the best algorithm from BBOB 2016. Best results are printed in bold.
Data produced with COCO v2.3.1, hv-hash=ff0e71e8cd978373

Chapter 9

Implementations

Contents

We present in this chapter different implementations done with respect to this PhD project. Mainly, we introduce the ask-and-tell pattern [51] in all the implementations. We present an implementation of the ask-and-tell paradigm for the weighted multi-recombination step-size adaptive evolution

strategies, specifically applied to CSA without cumulation and xNES without covariance matrix adaptation. We then present the `pycomocma` Python package that implements COMO-CMA-ES [185] by using the ask-and-tell interface along with the CMA-ES Python package [89]. Finally we present the Matlab interface of COMO-CMA-ES, that we derive from the `pycomocma` Python package.

9.1 The ask-and-tell optimization interface

Often in practice, optimization algorithms are implemented with a functional paradigm placing the optimization method at the top of the hierarchy, $i.e.$ where a single call to a procedure does all the optimization process [44]. This technique has the advantage to be easy to use. However it does not allow any prototyping from the user, specifically if one wants to interact consistently with the optimizer, step by step.

This is why the ask-and-tell paradigm introduced in [51] is particularly adapted in our context. It allows to create an optimization algorithm independently to the objective function that we want to optimize. The optimization algorithm implementation is mainly based on the search space, instead of the objective space. This independence is helpful in practice to, for instance, implement the optimization algorithm in Python and apply it to an objective function implemented in Matlab.

As indicated by its nomenclature, the ask-and-tell paradigm contains mainly the two methods of the optimizer `ask` and `tell`. The method `ask` asks the optimizer to generate search space points to be evaluated, based on its mechanism, and returns the asked points. Then the method `tell` tells to the optimizer's instance the asked points along with their corresponding objective function values, and returns the new instance. It can be summarized in Algorithm 11.

Algorithm 11 The ask-and-tell interface

Given: optimizer opt
while not stopping criterion **do**
    $X = \mathrm{ask}(\mathrm{opt})$
    $Y = (f(x) \text{ for } x \in X)$
    $\mathrm{opt} = \mathrm{tell}(\mathrm{opt}, \mathrm{X}, \mathrm{Y})$
end while

At the end of the loop, the favorite solution $\mathrm{x}_{\mathrm{opt}}$ is given by the method best :

$$\mathrm{x}_{\mathrm{opt}} = \mathrm{best}(\mathrm{opt}).$$

A direct observation from Algorithm 11 is that for the optimization of an expensive problem (where the evaluation of the objective function is too expensive with respect to the standard operations of the optimizer), the procedure is easily parallelizable by distributing on each iteration the evaluations of the objective functions, between the method `ask` and the method `tell`.

9.1.1 An ask-and-tell Python interface for the step-size adaptive evolution strategies with recombination

We introduce here a generic implementation of step-size adaptive evolution strategies with recombination, where only the step-size is adapted, without adaptations that use the cumulative path. The main goal is to observe in practice how the ask-and-tell interface works.

```
from cma import interfaces


class Saes(interfaces.OOOptimizer):
    """
    Step-size adaptive Evolution Strategy framework that inherits from the
    cma `interfaces.OOOptimizer`, with the ask-and-tell interface available.

    Attributes and Properties
    =========================
    - `incumbent`: current solution of the optimizer.
    - `dimension`: dimension of the search space.
    - `sigma`: current step-size of the optimizer.
    - `countevals`: number of function evaluations.
    - `countiter`: number of iterations.
    - `popsize`: population size.
    - `mu`: number of selected offspring.
    - `ksigma`: learning rate for the step-size.
    - `km`: learning rate for the incumbnent.
    - `weights`: weights given to the `mu` selected offspring for the
    recombination.
    - `step_size_factor`: ratio between the current and latter step-size
    values.
    - `ask`: generates new candidate solutions.
    - `tell`: passes the objective values and updates the states of the
            optimizer.
    """
    def __init__(self, x0, sigma0, inopts = {
                    'popsize': None, 'mu': None, 'weights': None}):
        """
        Initialization:
            - `x0` is the initial solution
            - `sigma0` is the initial step-size
            - `inopts` is a dictionary setting the population  size (with the
            key 'popsize') and the number of selected offspring ('mu').
        """

```

```
        self.incumbent = np.array(x0)
        self.dimension = len(x0)
        self.sigma = sigma0
        self.countevals = 0
        self.countiter = 0
        popsize = inopts['popsize']
        if popsize is None:
            self.popsize = int(4 + np.floor(3 * np.log(self.dimension)))
        else:
            self.popsize = popsize
        mu = inopts['mu']
        if mu is None:
            self.mu = int(self.popsize // 4)
        else:
            self.mu = mu
        self.km = 1
        self.ksigma = 1
        weights = inopts['weights']
        if weights is not None:
            self.weights = weights
        else:
            self.weights = 1/self.mu * np.ones(self.mu)
        self.step_size_factor = 0.0
        self.x0 = x0
        self.sigma0 = sigma0
    def ask(self):
        """
        ask `popsize` offspring normally distributed with
        mean `incumbent` and standard deviation `sigma`.
        """
        return [np.array(self.incumbent) + self.sigma * np.array(u) for u in
                np.random.randn(self.popsize, self.dimension)]

    def tell(self, solutions, objective_values):
        """
        pass function values to update the variables' states.
        """
        _old_incumbent = self.incumbent

        sorted_indices = sorted(range(len(solutions)),
                                key = lambda i: objective_values[i])
        selected_indices = sorted_indices[:self.mu]
        selection = np.array([solutions[i] for i in selected_indices])

        normal_selection = (selection - self.incumbent) / self.sigma


        self.incumbent = (1 - self.km) * np.array(self.incumbent) + (
                         self.km * np.dot(self.weights, selection))

        self.step_size_factor = self.step_size_update(normal_selection)
        self.sigma = self.sigma * self.step_size_factor

        self.dist = np.linalg.norm(self.incumbent - _old_incumbent)

        self.countevals += self.popsize
```

```
        self.countiter += 1

    def stop(self):
        """
        """
        pass

    def step_size_update(self, selection):
        """
        """
        pass


class XNes(Saes):
    """
    inherits from the `Saes` class to implement the 'xNES' Evolution Strategy
    without covariance matrix adaptation.
    """
    def __init__(self, x0, sigma0, inopts = {
                    'popsize': None, 'mu': None, 'weights': None}):
        """
        """

        Saes.__init__(self, x0, sigma0, inopts)



    def step_size_update(self, selection, expo=True):
        """
        return the step-size factor for the 'xNes' Evolution Strategy.
        """
        norm_square_selection = np.array([np.linalg.norm(u)**2 for u in
                                          selection]) - self.dimension
        log_factor = 1/(2 * self.dimension) * self.ksigma * np.dot(
                self.weights, norm_square_selection)
        if expo:
            return np.exp(log_factor)
        else:
            return log_factor

class Csaes(Saes):
    """
    inherits from the `Saes` class to implement the 'Csa-es' Evolution
    Strategy
    without covariance matrix adaptation.
    """
    def __init__(self, x0, sigma0, inopts = {
                    'popsize': None, 'mu': None, 'weights': None}):
        """
        """
        Saes.__init__(self, x0, sigma0, inopts)

    def step_size_update(self, selection, expo=True):
        """
        return the step-size factor for the 'xNes' Evolution Strategy.
        """
```

```
148         norm_weights_square = (np.linalg.norm(self.weights))**2
149         log_factor = 1/2*self.ksigma * (-1 + np.linalg.norm(np.dot(
150                 self.weights, selection))**2/(self.dimension *
    norm_weights_square))
151         if expo:
152             return np.exp(log_factor)
153         else:
154             return log_factor
```

Based on this ask-and-tell implementation, we illustrate the main results of Chapter 4 in Section 1.5 of the appendix. In Figure A.1, we present for $(1,3)$-CSA-ES and for $(5,11)$-CSA-ES a linear behavior of the distance of the favorite solution to a reference point and of the step-size. The objective functions are the Euclidean norm square and a nontrivial linear function. With the same settings, the stability of the σ-normalized chain is illustrated in Figure A.2, and the linear behavior of the expected log-progress is illustrated in Figure A.3. For $(1,3)$-xNES and for $(5,11)$-xNES, we observe in Figure A.6 a linear behavior of the expected log-progress on the Euclidean norm square and on a nontrivial linear function. In Figure A.5, the ergodicity of the σ-normalized chain is presented. In Figure A.4, we observe with the same problems a linear behavior of the distance of the favorite solution to a reference point (usually the optimum) and of the step-size.

9.1.2 Use case: Exponential Natural Evolution Strategies without covariance matrix adaptation

We test the above code of Section 9.1.1 with a specific Optimization algorithm. We create in the following a simple calling sequence example for the Exponential Natural Evolution Strategy (xNES) [70], without covariance matrix adaptation, minimizing the following linear function $l^\star : x \longmapsto x_1$. We then expect the final favorite solution having its first coordinate diverging towards $-\infty$.

[1]:
```
import cma
import saes
%pylab nbagg
from matplotlib import pyplot as plt
import numpy as np
import scipy.stats as sps
```

```
Populating the interactive namespace from numpy and matplotlib
```

[2]:
```
lam = 11
mu = 5
weights = np.arange(mu, 0, -1)
```

```
weights = weights / sum(weights)

es = saes.XNes(10 * [1], 0.2, inopts={'popsize': lam, 'mu': mu,␣
 ↪'weights': weights}) # es with non-uniform weights

function = lambda x: x[0]
#function = cma.ff.sphere

num_steps = 10000
```

[3]:
```
for i in range(num_steps):
    X = es.ask()
    objective_values = [function(x) for x in X]
    es.tell(X, objective_values)
```

[4]:
```
es.incumbent # gives the solution at the last step
```

[4]:
```
array([-2.27683548e+113,  7.92314104e+111, -4.38439697e+111,
        2.45857432e+111,  5.40496049e+111, -1.71161266e+111,
        1.79033942e+112, -2.83903187e+111,  1.10629799e+112,
       -9.41423910e+110])
```

[5]:
```
function(es.incumbent) # gives the value at the last step
```

[5]:
```
-2.276835476814582e+113
```

9.2 The pycomocma Python package

The `pycomocma` package is a Python implementation of COMO-CMA-ES [185] which is a Multiobjective Evolution Strategy, based upon the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) single optimizer [95].

For the time being, only the bi-objective case is tested and functional. The package is located in the following link: https://github.com/CMA-ES/pycomocma. It contains the module called `comocma`.

The ask-and-tell paradigm is used for the implementation of this package. A number of points p is fixed before the optimization process, and the goal of the multiobjective optimizer is to obtain the optimal p-distribution of the points with respect to the hypervolume indicator [21]. In `pycomocma`, each point of those p points is represented

by the favorite solution of a CMA-ES instance. Those p CMA-ES evolve sequentially until a stopping criteria is met [185]. The parallelizable version of the multiobjective optimizer is available with a slight different search dynamic, wherein all the CMA-ES instances are updated at the same time.

9.2.1 Installation

The `pycomocma` Python package can be installed either via

```
pip install git+https://github.com/CMA-ES/pycomocma.git@master
```

or simply via

```
pip install comocma
```

9.2.2 Links

The following links give detailed information with respect to the `pycomocma` package.

- Code on Github: https://github.com/CMA-ES/pycomocma
- Documentation in
  - apidocs format: https://cma-es.github.io/pycomocma/comocma-apidocs/index.html
  - epydocs format: https://cma-es.github.io/pycomocma/comocma-epydocs/index.html

9.2.3 Testing of the `comocma` module

In order to test if everything is well installed, the script

```
python -m comocma
```

runs the test written in the `__main__` file, which allows to test all the doctest scripts included inside the `comocma` module.

9.2.4 Instantiating a multiobjective solver

To create a multiobjective solver instance, we use beforehand a factory function within the `comocma` module called `get_cmas`. It allows to create the adequate number of CMA-ES instances with additional attributes, called in this context *kernels*, whose favorite solutions represent the estimated Pareto set. The following process can be used to initialize a COMO-CMA-ES instance, and to create a callable multiobjective problem based on callable single-objective problems.

Importing necessary packages:

```
import cma, comocma
```

Setting parameters:

```
dimension = 10  # dimension of the search space
num_kernels = 5
 # number of single-objective solvers (number of points on the front)
sigma0 = 0.2    # initial step-sizes
```

Instantiate a multiobjective solver

```
list_of_solvers = comocma.get_cmas(num_kernels * [dimension * [0]], sigma0)
# produce `num_kernels cma instances`
moes = comocma.Sofomore(list_of_solvers, reference_point=[11, 11])
# create a bi-objective como-cma-es instance
moes3 = comocma.Sofomore(list_of_solvers, reference_point=[11, 11, 11])
# create a multiobjective como-cma-es instance
```

Setting a callable multiobjective function

```
fitness = comocma.FitFun(cma.ff.sphere, lambda x: cma.ff.sphere(x-1))
 # a callable bi-objective function
fitness3 = comocma.FitFun(cma.ff.sphere, lambda x: cma.ff.sphere(x-1),
 lambda x: cma.ff.sphere(x+1)) # a callable multiobjective function
```

Single-objective options: a use case with few cma-es' options

```
list_of_solvers = comocma.get_cmas(num_kernels * [dimension * [0]], 0.2,
inopts={'bounds': [0.2, 0.9], 'tolx': 10**-7,'popsize': 32})
# produce `num_kernels cma instances`
moes = comocma.Sofomore(list_of_solvers, [1.1, 1.1])
# create a como-cma-es instance
```

Use case with some Multiobjective options

```
list_of_solvers = comocma.get_cmas(num_kernels * [dimension * [0]], 0.2)
moes = comocma.Sofomore(list_of_solvers, [1.1, 1.1],
opts={'archive': True, 'restart': None, 'update_order': None})
# create a como-cma-es instance
```

9.2.5 The `Optimize` interface

This interface is inherited from the `pycma` package [89]. It is a method of the multiobjective optimizer, taking as required argument the multiobjective function to optimize. The following sequences are the baselines to use in practice.

Initialization

```
import cma, comocma

dimension = 10  # dimension of the search space
num_kernels = 5 # number of single-objective solvers
 # (number of points on the front)
sigma0 = 0.2    # initial step-sizes

list_of_solvers = comocma.get_cmas(num_kernels * [dimension * [0]], sigma0)
# produce `num_kernels cma instances`
moes = comocma.Sofomore(list_of_solvers, [11,11])
# create a como-cma-es instance

fitness = comocma.FitFun(cma.ff.sphere, lambda x: cma.ff.sphere(x-1))
# a callable bi-objective function
```

Optimizing `fitness` until default stopping criteria

```
moes.optimize(fitness)
```

```
Iterat #Fevals   Hypervolume   axis ratios   sigmas   min&max stds
                                 (median)  (median)    (median)
    1      10 1.210000000000000e+00 1.0e+00 2.00e-01  2e-01  2e-01
    2      20 1.210000000000000e+00 1.0e+00 2.00e-01  2e-01  2e-01
    3      30 1.210000000000000e+00 1.0e+00 1.85e-01  2e-01  2e-01
  100    1000 1.207601015381810e+00 1.6e+00 3.40e-02  3e-02  3e-02
```

```
  200   2000 1.209903687756354e+00 1.7e+00 7.74e-03  5e-03  6e-03
  300   3000 1.209997694077156e+00 1.8e+00 2.03e-03  1e-03  1e-03
  400   4000 1.209999800600613e+00 1.8e+00 4.90e-04  2e-04  3e-04
  480   4800 1.209999979594839e+00 1.9e+00 2.02e-04  7e-05  9e-05
```

Optimizing `fitness` with a limited number of iterations

```
moes.optimize(fitness, iterations=300)
```

```
Iterat #Fevals   Hypervolume   axis ratios   sigmas   min&max stds
                                (median)  (median)    (median)
  1     10 1.100000000000000e+01 1.0e+00 2.00e-01  2e-01  2e-01
  2     20 2.158412269365152e+01 1.0e+00 2.00e-01  2e-01  2e-01
  3     30 2.896035267829712e+01 1.0e+00 1.98e-01  2e-01  2e-01
100   1000 9.512982413314423e+01 1.7e+00 1.01e-01  8e-02  9e-02
200   2000 9.703624875547615e+01 1.9e+00 4.27e-02  3e-02  4e-02
300   3000 9.722958234416403e+01 1.9e+00 1.63e-02  9e-03  1e-02
```

Optimizing `fitness` with a maximum number of evaluations

```
moes.optimize(fitness, maxfun=3000)
```

```
Iterat #Fevals   Hypervolume   axis ratios   sigmas   min&max stds
                                (median)  (median)    (median)
  1     10 1.100000000000000e+01 1.0e+00 2.00e-01  2e-01  2e-01
  2     20 2.158412269365152e+01 1.0e+00 2.00e-01  2e-01  2e-01
  3     30 2.896035267829712e+01 1.0e+00 1.98e-01  2e-01  2e-01
100   1000 9.512982413314423e+01 1.7e+00 1.01e-01  8e-02  9e-02
200   2000 9.703624875547615e+01 1.9e+00 4.27e-02  3e-02  4e-02
300   3000 9.722958234416403e+01 1.9e+00 1.63e-02  9e-03  1e-02
```

9.2.6 The ask-and-tell interface

```
while not moes.stop():
    solutions = moes.ask("all")
    objective_values = [fitness(x) for x in solutions]
    moes.tell(solutions, objective_values)
    moes.disp()          # display datas during the optimization
    moes.logger.add()
 # logging data after each `ask` and `tell` call
```

```
Iterat #Fevals   Hypervolume   axis ratios   sigmas   min&max stds
                                 (median)  (median)    (median)
    1     180 1.990425600000000e-01 1.0e+00 1.88e-01  2e-01  2e-01
    2     360 2.279075246432772e-01 1.1e+00 1.87e-01  2e-01  2e-01
    3     540 2.436105134581627e-01 1.2e+00 1.90e-01  2e-01  2e-01
  100   18000 3.607157703968831e-01 2.1e+00 1.80e-02  1e-02  2e-02
  200   35172 3.635275131024869e-01 2.1e+00 5.95e-03  4e-03  5e-03
  300   49788 3.637412031970786e-01 2.2e+00 1.29e-03  8e-04  1e-03
  320   50784 3.637421277015990e-01 2.2e+00 1.26e-03  7e-04  9e-04
```

Argument of `moes.ask`

```
solutions = moes.ask()
# we generate offspring for only one kernel (sequential)
solutions = moes.ask("all")
# we generate offspring simultaneously for all kernels (parallel)
solutions = moes.ask(number_asks)
# we generate offspring for `number_asks` kernels
```

9.2.7 Picklable object: saving and resuming a MO optimization with the ask-and-tell interface

With our collaboration with Storengy, the produced implementation is used in production, with all the human constraints that come with it. For example the implemented optimizer should be robust enough with respect to server failures or to periodic maintenances of the systems that require the interruption of the running machines. Therefore it is very useful to serialize the optimizer (seen as Python objects). Alongside the ask-and-tell interface, the serialization allows the user, after each iteration (step), to save the current optimizer's instance to a file *per se*, and then load it in a program later on. These serialization and de-serialization methods are what Python's `pickle` module does [190]. We present in the following simple use cases of multiobjective optimization with pickled COMO-CMA-ES instances.

Initialization

```
import cma, como, pickle

dimension = 10  # dimension of the search space
num_kernels = 5
```

```
 # number of single-objective solvers (number of points on the front)
sigma0 = 0.2    # initial step-sizes

list_of_solvers = como.get_cmas(num_kernels * [dimension * [0]], sigma0)
# produce `num_kernels cma instances`
moes = como.Sofomore(list_of_solvers, reference_point = [11,11])
# create a como-cma-es instance

fitness = como.FitFun(cma.ff.sphere, lambda x: cma.ff.sphere(x-1))
# a callable bi-objective function
```

Saving an optimization

```
for i in range(100):
    solutions = moes.ask()
    objective_values = [fitness(x) for x in solutions]
    moes.tell(solutions, objective_values)
    moes.disp()

pickle.dump(moes, open('saved-mocma-object.pkl', 'wb'))
# we save the instance
print('saved')
del moes  # deleting completely the Sofomore instance
```

Output

```
Iterat #Fevals   Hypervolume   axis ratios   sigmas   min&max stds
                                  (median)  (median)    (median)
    1     10 1.100000000000000e+01 1.0e+00 2.00e-01  2e-01  2e-01
    2     20 2.845200549045931e+01 1.0e+00 2.00e-01  2e-01  2e-01
    3     30 3.440089785096067e+01 1.0e+00 2.00e-01  2e-01  2e-01
  100   1000 9.562953505152342e+01 1.9e+00 1.13e-01  9e-02  1e-01
saved
```

Resuming an optimization

```
moes = pickle.load(open('saved-mocma-object.pkl', 'rb'))
# we load the saved file here

moes.optimize(fitness, iterations=400)
```

Output

```
200    2000 9.716644477685412e+01 1.9e+00 3.33e-02  2e-02  3e-02
300    3000 9.723550009906029e+01 2.0e+00 1.13e-02  6e-03  8e-03
400    4000 9.724067117112808e+01 1.9e+00 2.95e-03  1e-03  2e-03
500    5000 9.724107479961819e+01 2.0e+00 9.38e-04  4e-04  5e-04
```

9.2.8 Example of plots

At the end of a multiobjective optimization process with the `comocma` module in Python, some data visualization tools are handy to assess in some images the behavior of the multiobjective optimizer's instance. We have within the Python `pycomocma` package some visualization tools specific to COMO-CMA-ES, and we also preserve the data visualization tools from the `pycma` package [89], since each point of the estimated Pareto set is in fact the favorite solution of an underlying CMA-ES instance (called a kernel in our context).

COMO-CMA-ES data plottings

```
moes.logger.plot_front()
```

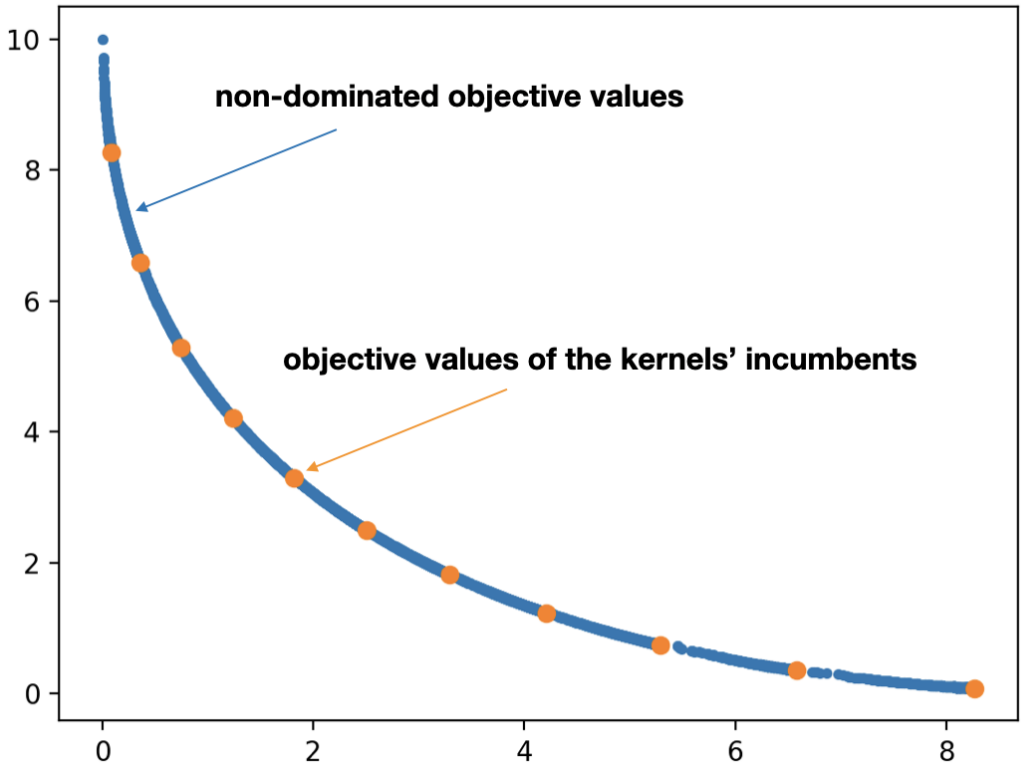


Figure 9.1: Estimated Pareto front of (f_1, f_2) where $f_1 : x \mapsto \|x\|^2$ and $f_2 : x \mapsto \|x - \mathbb{1}\|^2$
and $\mathbb{1}$ is the all-ones vector. The points in blue represents the non-dominated points found throughout the optimization process, and the points in orange are the objective values of the favorite solutions of the single-objective solvers' instances.

For the visualization of various and generic metrics related to the COMO-CMA-ES algorithm, the following calling sequence line can be used.

```
moes.logger.plot_divers()
```

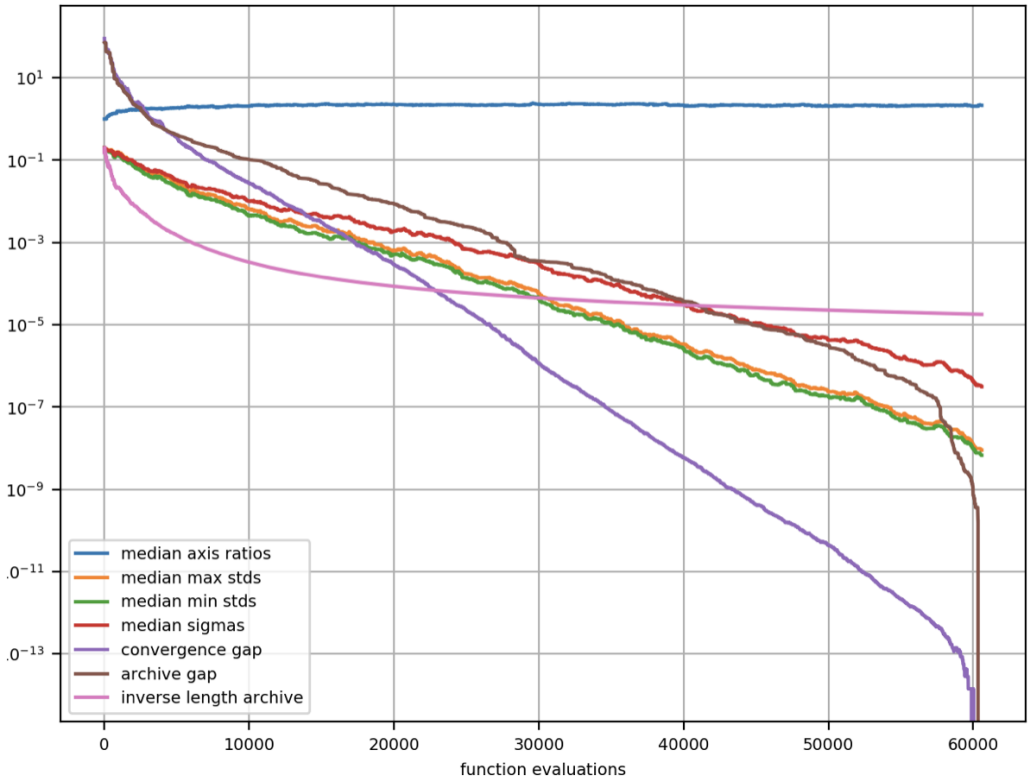


Figure 9.2: Various graphs representing the medians of some parameters of the different CMA-ES instances that constitute the COMO-CMA-ES instance, the convergence gap, the archive gap and the inverse length of the archive, all with respect to the function evaluations.

CMA-ES plots of written data

As stated above, the standard data plots of the CMA-ES instances that form the COMO-CMA-ES instance can still be visualized. Those data are stored in a folder named by default `cma_kernels`. The index of a kernel is the name of the file within that folder where the corresponding CMA-ES data are stored. For example for kernel number 0, we have the following.

```
cma.plot("cma_kernels/0")
```

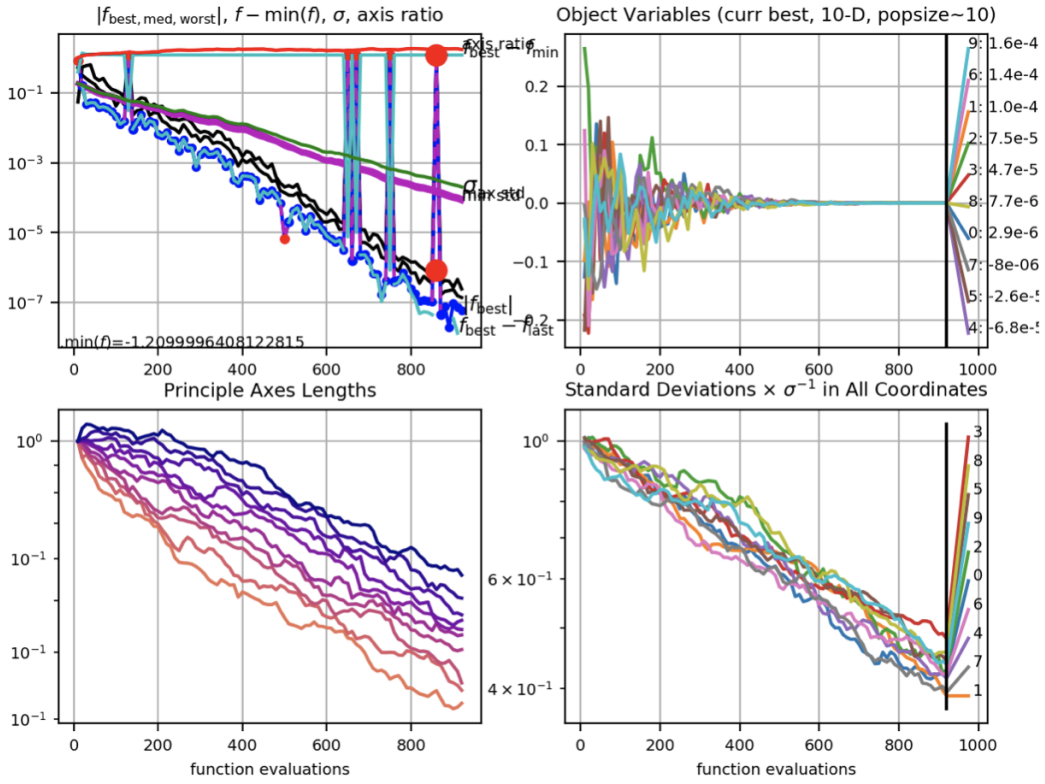


Figure 9.3: Standard visualization of a CMA-ES instance plot.

9.3 The Matlab interface of COMO-CMA-ES

We descrtibe in this section a Matlab interface implementation of the COMO-CMA-ES algorithm [185]. The interface typically calls the Python methods from the `pycomocma` Python package. It exploits the ask-and-tell paradigm adopted in that Python package and the adaptability of some specific data structures from Python to Matlab, and from Matlab to Python too. Note that this interface is only functional for Matlab 2014b or later, and the installation of the `pycomocma` Python package is necessary as it is mainly used.

More concretely at each iteration, the Matlab interface calls the method `ask` of the Python module `comocma`, then evaluates the generated solutions with the Matlab multiobjective function to optimize (after adaptation of the data structure), and finally call the `tell` method of `comocma` with the adequate Python data structure this time.

One additional constraint is that the interface should be the same as Matlab implementations of well-known multiobjective evolutionary algorithms such as MO-CMA-ES and SMS-EMOA, developed by Dimo Brockhoff (for the MO-CMA-ES part) and Timo

Wagner (for the SMS-EMOA part) [41, 186].

9.3.1 The Matlab interface

With the latter constraint, the Matlab interface for the COMO-CMA-ES algorithm produces the following calling sequence.

```
[paretoFront, paretoSet, out] = COMOCMAES(problem, nObj, xstart,
    insigma, inopts)
```

Input arguments

- `PROBLEM` is a string function name. Calling a problem with a matrix as first parameter interprets the columns as individual solutions and computes the objective vectors and the repaired variables for all solutions in parallel.
- `nObj` gives the number of objectives of the multiobjective problem.
- `xstart` indicates the initial sample points that will be used to initialize the individual means of COMO-CMA-ES's sample distributions. The number of rows thereby gives the number of variables and the number of columns the population size. If the number of columns does not match the population size given in `opts.nPop`, the first column of `xstart` is used to initialize the mean vectors of all sample distributions. Note also that a string can be given that will be evaluated as MATLAB code such as `'rand(10, nPop)'`.
- `insigma` is the initial step-sizes for the single-objective cma-es.
- `inopts` (an optional argument) is a struct holding additional input options.

Output

- `paretoFront` is a matrix holding the objectives in rows. Each column holds the objective vector of one solution.
- `paretoSet` is a matrix holding the parameters in rows. Each column holds one solution.
- `out` a struct with additional information about the run.

9.3.2 Options of the Matlab interface

In addition, several options are inherited from the Python package. More specifically the `picklable` property of the Python COMO-CMA-ES instance is diversified into finer

options very specific to industrial uses. More generally, the user can have access to most of the options in Python, provided that the correct wrapping is done to handle the changes of the data structures. Here are some example of the options' usage.

- `opts = COMOCMAES()` sets the default options.
- `opts.nPop` is the number of kernels for the algorithm.
- `opts.popsize` defines the number of offspring per kernel.
- `opts.tolx` is the tolerance in x-values for stopping criterion purpose.
- `opts.number_asks` is the number of kernels from which we generate offspring simultaneously. By default all "active" kernels are asked simultaneously.
- `opts.maxiter` is the maximum number of iterations allowed.
- `opts.maxEval` is the maximum number of evaluations allowed.
- `opts.logger` is the log data if 1. If 0, no data is logged.
- `opts.okresume` is a boolean. If 1 then the optimization is resumed from a saved one. Otherwise it is 0 and the optimization starts from scratch.
- `opts.resumefile` is a string. if the string is empty, then the optimization is not saved. Else, the optimization is saved in the given path. If in addition `opts.okresume = 1`, then the optimization is resumed from the file `opts.resumefile`.
- `opts.verb_disp` allows to display plots every `opts.verb_disp` iterations.
- `opts.abscissa` If 1, plot with number of function evaluations as axis, if 0 plot with number of iterations.
- `opts.bounds` is the search space boundary constraints.
- `opts.restart` states whether or not we do COMO-CMA-ES with restart. The default value is `'False'`.
- `opts.incrementer` is the incrementer of the population size in the case where `opts.restart` is true. The default value is 2.

Conclusion and discussion

In this work, we have approached various topics in designing multiobjective optimizers and analyzing single-objective evolution strategies. We have posed questions in the introduction that we all answer to some extent.

In single-objective optimization, we have analyzed a subclass of step-size adaptive evolution strategies with recombination on scaling-invariant functions. This class includes well-known algorithms, such as CSA-ES without cumulation and xNES without covariance matrix adaptation.

This analysis starts by characterizing *continuous* scaling-invariant functions in Chapter 3: they are composites of homeomorphisms with *continuous* positively homogeneous functions. We also present necessary and sufficient conditions for a scaling-invariant function to be a strictly increasing transformation of a positively homogeneous function. Surprisingly, we present real scaling-invariant functions that are not monotonic on any nontrivial interval.

Various properties of the sublevel sets of scaling-invariant functions are also developed in Chapter 3. We have shown conditions for a lower semi-continuous scaling-invariant function with a unique global argmin to have compact sublevel sets. For a C^1 scaling-invariant function with a unique global argmin, we have crafted a geometric object that is a compact neighborhood of a level set with non-vanishing gradient, intersecting any half-line from the reference point to a unique point.

That geometric object is key in the way that we prove in Chapter 4 the convergence in distribution of the step-size multiplicative factor on an increasing transformation of a C^1 scaling-invariant function with a unique global argmin, towards the step-size change on a nontrivial linear function. This convergence is observed when the σ-normalized chain—the difference between the favorite solution and a reference point, divided by the step-size—goes to ∞. Based on the invariance under rotation of the step-size update function that we consider, we derive that the step-size multiplicative factors have the same distribution on nontrivial linear functions. The convergence in distribution above is what leads us to the geometric ergodicity of the σ-normalized chain. We use existing results from deterministic control models to deduce φ-irreducibility, aperiodicity and T-chain property for the σ-normalized chain.

Overall, that chain is geometrically ergodic.

We develop generic results for the ergodicity of an abstract double chain of the form $\{W_k = (Z_k, U_{k+2}) ; k \in \mathbb{N}\}$ with $Z_{k+1} = G(Z_k, U_{k+1})$ where G is a measurable function. Under suitable conditions, we transfer the stability of the chain $\{Z_k; k \in \mathbb{N}\}$ to the double chain $\{W_k; k \in \mathbb{N}\}$. This transfer is crucial in the way that we apply the LLN and obtain linear behaviors for our class of step-size adaptive evolution strategies with recombination on increasing transformations of either C^1 scaling-invariant functions with a unique global argmin, or nontrivial linear functions.

The main condition for linear behavior is that the logarithm of the step-size increases on expectation. We show that this is the tightest condition we can get (also tighter than previous conditions in the literature for ES with one parent population), since it is equivalent to the geometric divergence of the step-size on nontrivial linear functions.

A notable limitation in our proofs is that we do not give the sign of the rate on increasing transformations of C^1 scaling-invariant functions with a unique global argmin. It is an expectation taken over an unknown invariant probability measure. Yet we give a central limit theorem to efficiently approximate that rate, via Monte Carlo simulations. Another major limitation is the omission of cumulation in our methodology, whilst step-size adaptive evolution strategies with cumulation are what is used in practice. Therefore adding the cumulative stochastic process in the algorithm framework is a natural next step of this work.

In multiobjective optimization, we design COMO-CMA-ES in Chapter 7, a multiobjective optimizer based on the non-elitist CMA-ES with recombination, that converges geometrically fast on bi-objective strictly convex-quadratic functions thoroughly studied in Chapter 6. The convergence notion is built upon the hypervolume indicator.

COMO-CMA-ES is an instance of the designed Sofomore framework, that allows to build a multiobjective optimizer based on single-objective optimization algorithms. In Sofomore, we create a new performance metric based on the hypervolume indicator, that we call UHVI: Uncrowded Hypervolume Improvement. That performance metric improves the limitations of the traditional two-way ranking that consists of using a Pareto ranking followed by a hypervolume Improvement or a hypervolume contribution. The two-way ranking tends to steer dominated points towards regions already occupied by a non-dominated point. This provokes an undesirable crowdedness of the final solutions. With the UHVI, the opposite of the distance to the so-called empirical Pareto front is added for dominated points. This mechanism guides the dominated solutions towards regions of the Pareto front unexplored by non-dominated points.

COMO-CMA-ES is experimentally studied on various classes of bi-objective convex quadratic problems constructed in Chapter 6. These subclasses test the bahavior of the multiobjective optimizer with respect to non-separability, ill-conditioning of the objective functions, and curvature of the Pareto front. COMO-CMA-ES converges

linearly on these subclasses of problems. We also observe that COMO-CMA-ES performs better than well-known multiobjective optimizers on these problems, that are MO-CMA-ES and NSGA-II.

In Chapter 8, COMO-CMA-ES is benchmarked in the COCO platform that compares continuous optimizers on a variety of problems. Yet COCO has a performance metric that depends on all the non-dominated points observed so far by the algorithm, while COMO-CMA-ES is designed to approximate the optimal distribution of a fixed number of solutions p on the Pareto set, in order to maximize the hypervolume of p points. Nonetheless, COMO-CMA-ES performs good on this platform for large budgets. We conjecture that it is due to two things: the UHVI that allows to visit more unexplored regions, and the recombination effect of the CMA-ES that produces a large stationary variance.

COMO-CMA-ES has not yet been tested on a problem with three or more objective functions. This is therefore a natural next step to our work, to observe especially whether or not the UHVI remains relevant in this framework. We have presented the implementation of COMO-CMA-ES and the Sofomore framework in Chapter 9. A python package $\mathrm{pycomocma}$ is developed for this purpose. The implementation is quite compatible with the pycma package of the CMA-ES. This allows to transfer the regular updates of the CMA-ES so that the underlying single-objective optimizers used in COMO-CMA-ES will stay up to date.

For industrial usage by our partner Storengy, we have implemented a Matlab interface for COMO-CMA-ES. That version incorporates various options often needed in the industry. Especially the function evaluations of the multiobjective optimizer can be fully parallelized during an optimization. An instance of the Sofomore framework is also picklable, to help saving and resuming efficiently an optimization problem, after each iteration. This is much needed to cope with server failures and periodic maintenances that happen in industrial use cases.

Bibliography

[1] Joseph Aczél. *Lectures on functional equations and their applications*. Academic press, 1966.

[2] Youhei Akimoto, Anne Auger, and Tobias Glasmachers. Drift theory in continuous search spaces: expected hitting time of the (1+ 1)-ES with 1/5 success rule. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 801–808, 2018.

[3] Youhei Akimoto, Anne Auger, Tobias Glasmachers, and Daiki Morinaga. Global linear convergence of evolution strategies on more than smooth strongly convex functions. *arXiv preprint arXiv:2009.08647*, 2020.

[4] Youhei Akimoto, Anne Auger, and Nikolaus Hansen. Convergence of the continuous time trajectories of isotropic evolution strategies on monotonic $\mathcal{C}^2$-composite functions. In *International Conference on Parallel Problem Solving from Nature*, pages 42–51. Springer, 2012.

[5] Youhei Akimoto, Anne Auger, and Nikolaus Hansen. An ode method to prove the geometric convergence of adaptive stochastic algorithms. *arXiv preprint arXiv:1811.06703*, 2018.

[6] Youhei Akimoto, Anne Auger, and Nikolaus Hansen. Quality gain analysis of the weighted recombination evolution strategy on general convex quadratic functions. *Theoretical Computer Science*, 2018.

[7] Youhei Akimoto, Yuichi Nagata, Isao Ono, and Shigenobu Kobayashi. Bidirectional relation between cma evolution strategies and natural evolution strategies. In *International Conference on Parallel Problem Solving from Nature*, pages 154–163. Springer, 2010.

[8] Youhei Akimoto, Yuichi Nagata, Isao Ono, and Shigenobu Kobayashi. Theoretical analysis of evolutionary computation on continuously differentiable functions. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pages 1401–1408, 2010.

[9] Youhei Akimoto, Yuichi Nagata, Isao Ono, and Shigenobu Kobayashi. Theoretical foundation for cma-es from information geometry perspective. *Algorithmica*, 64(4):698–716, 2012.

[10] Youhei Akimoto and Yann Ollivier. Objective improvement in information-geometric optimization. In *Proceedings of the twelfth workshop on Foundations of genetic algorithms XII*, pages 1–10, 2013.

[11] Shun-ichi Amari and Hiroshi Nagaoka. *Methods of information geometry*, volume 191. American Mathematical Soc., 2007.

[12] Dirk V Arnold. Optimal weighted recombination. In *International Workshop on Foundations of Genetic Algorithms*, pages 215–237. Springer, 2005.

[13] Dirk V Arnold. Weighted multirecombination evolution strategies. *Theoretical computer science*, 361(1):18–37, 2006.

[14] Dirk V Arnold and H-G Beyer. Performance analysis of evolutionary optimization with cumulative step length adaptation. *IEEE Transactions on Automatic Control*, 49(4):617–622, 2004.

[15] Dirk V Arnold and Hans-Georg Beyer. Random dynamics optimum tracking with evolution strategies. In *International Conference on Parallel Problem Solving from Nature*, pages 3–12. Springer, 2002.

[16] A. Auger and D. Brockhoff. Theoretically Investigating Optimal μ-Distributions for the Hypervolume Indicator: First Results For Three Objectives. Poster presentation at Parallel Problem Solving from Nature (PPSN XI), Krakow, Poland, 2010.

[17] A. Auger, D. Brockhoff, N. Hansen, D. Tušar, T. Tušar, and T. Wagner. Benchmarking MATLAB's gamultiobj (NSGA-II) on the Bi-objective BBOB-2016 Test Suite. In *GECCO (Companion) workshop on Black-Box Optimization Benchmarking (BBOB'2016)*, pages 1233–1239. ACM, 2016.

[18] A. Auger and N. Hansen. Theory of Evolution Strategies: a New Perspective. In *Theory of Randomized Search Heuristics: Foundations and Recent Developments*. World Scientific Publishing, 2011. in press.

[19] Anne Auger. Convergence results for the (1, λ)-SA-ES using the theory of ϕ-irreducible markov chains. *Theoretical Computer Science*, 334(1-3):35–69, 2005.

[20] Anne Auger. *Thèse d'habilitation à diriger des recherches" Analysis of Comparison-based Stochastic Continuous Black-Box Optimization Algorithms"*. PhD thesis, University Paris Sud, 2016.

[21] Anne Auger, Johannes Bader, Dimo Brockhoff, and Eckart Zitzler. Theory of the hypervolume indicator: optimal μ-distributions and the choice of the reference point. In *Foundations of Genetic Algorithms (FOGA 2009)*, pages 87–102, Orlando, Florida, USA, 2009. ACM.

[22] Anne Auger, Johannes Bader, Dimo Brockhoff, and Eckart Zitzler. Hypervolume-based multiobjective optimization: Theoretical foundations and practical implications. *Theoretical Computer Science*, 425:75–103, 2012.

[23] Anne Auger and Benjamin Doerr. *Theory of randomized search heuristics: Foundations and recent developments*, volume 1. World Scientific, 2011.

[24] Anne Auger and Nikolaus Hansen. Linear convergence on positively homogeneous functions of a comparison based step-size adaptive randomized search: the (1+ 1)-ES with generalized one-fifth success rule. *arXiv preprint arXiv:1310.8397*, 2013.

[25] Anne Auger and Nikolaus Hansen. Linear convergence of comparison-based step-size adaptive randomized search via stability of markov chains. *SIAM Journal on Optimization*, 26(3):1589–1624, 2016.

[26] Thomas Back. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press, 1996.

[27] Michel Benaïm, Josef Hofbauer, and Sylvain Sorin. Stochastic approximations and differential inclusions. *SIAM Journal on Control and Optimization*, 44(1):328–348, 2005.

[28] Michel Benaïm, Josef Hofbauer, and Sylvain Sorin. Stochastic approximations and differential inclusions, part ii: Applications. *Mathematics of Operations Research*, 31(4):673–695, 2006.

[29] R. Berghammer, T. Friedrich, and F. Neumann. Set-based Multi-objective Optimization, Indicators, and Deteriorative Cycles. In *Genetic and Evolutionary Computation Conference (GECCO 2010)*, pages 495–502, Portland, Oregon, 2010. ACM.

[30] Nicola Beume, Boris Naujoks, and Michael Emmerich. Sms-emoa: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, 2007.

[31] Hans-Georg Beyer and Dirk V Arnold. Qualms regarding the optimality of cumulative path length control in csa/cma-evolution strategies. *Evolutionary computation*, 11(1):19–28, 2003.

[32] Hans-Georg Beyer and Hans-Paul Schwefel. Evolution strategies–a comprehensive introduction. *Natural computing*, 1(1):3–52, 2002.

[33] Alexis Bienvenüe and Olivier François. Global convergence for evolution strategies in spherical problems: some simple proofs and difficulties. *Theoretical Computer Science*, 306(1-3):269–289, 2003.

[34] Patrick Billingsley. *Convergence of probability measures*. Wiley, 1999.

[35] Ihor O Bohachevsky, Mark E Johnson, and Myron L Stein. Generalized simulated annealing for function optimization. *Technometrics*, 28(3):209–217, 1986.

[36] Vivek S Borkar. *Stochastic approximation: a dynamical systems viewpoint*, volume 48. Springer, 2009.

[37] Zyed Bouzarkouna, Didier Yu Ding, and Anne Auger. Well placement optimization with the covariance matrix adaptation evolution strategy and meta-models. *Computational Geosciences*, 16(1):75–92, 2012.

[38] Karl Bringmann and Tobias Friedrich. Convergence of hypervolume-based archiving algorithms i: Effectiveness. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 745–752, Dublin, Ireland, 2011. ACM.

[39] D. Brockhoff, T.-D. Tran, and N. Hansen. Benchmarking Numerical Multiobjective Optimizers Revisited. In *Genetic and Evolutionary Computation Conference (GECCO 2015)*, pages 639–646. ACM, 2015.

[40] D. Brockhoff, T. Tušar, D. Tušar, T. Wagner, N. Hansen, and A. Auger. Biobjective performance assessment with the COCO platform. *ArXiv e-prints*, arXiv:1605.01746, 2016.

[41] Dimo Brockhoff, Tobias Wagner, and Heike Trautmann. 2 indicator-based multiobjective search. *Evolutionary Computation*, 23(3):369–395, 2015.

[42] Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

[43] Gerard Buskes and Arnoud van Rooij. Topological spaces. In *Topological Spaces*, pages 187–201. Springer, 1997.

[44] SL Campbell, Jean-Philippe Chancelier, and Ramine Nikoukhah. Modeling and simulation in scilab/scicos. *Google Scholar*, 2006.

[45] Rachid Chelouah and Patrick Siarry. A continuous genetic algorithm designed for the global optimization of multimodal functions. *Journal of Heuristics*, 6(2):191–213, 2000.

[46] Alexandre Chotard. *Markov chain Analysis of Evolution Strategies*. PhD thesis, Universite Paris Sud, 2015.

[47] Alexandre Chotard and Anne Auger. Verifiable conditions for the irreducibility and aperiodicity of markov chains by analyzing underlying deterministic models. *Bernoulli*, 25(1):112–147, 2019.

[48] Alexandre Chotard, Anne Auger, and Nikolaus Hansen. Cumulative step-size adaptation on linear functions. In *International Conference on Parallel Problem Solving from Nature*, pages 72–81. Springer, 2012.

[49] Carlos A Coello Coello, Gary B Lamont, David A Van Veldhuizen, et al. *Evolutionary algorithms for solving multi-objective problems*, volume 5. Springer, 2007.

[50] G. Collange, N. Delattre, N. Hansen, I. Quinquis, and M. Schoenauer. Multidisciplinary Optimisation in the Design of Future Space Launchers. In *Multidisciplinary Design Optimization in Computational Mechanics*, pages 487–496. Wiley, 2010.

[51] Yann Collette, Nikolaus Hansen, Gilles Pujol, Daniel Salazar Aponte, and Rodolphe Le Riche. On object-oriented programming of optimizers - examples in scilab. In Rajan Filomeno Coelho and Piotr Breitkopf, editors, *Multidisciplinary Design Optimization in Computational Mechanics*, pages 499–538. Wiley, New Jersey, 2010.

[52] Andrew R Conn, Nicholas IM Gould, and Philippe L Toint. *Trust region methods*. SIAM, 2000.

[53] Andrew R Conn, Katya Scheinberg, and Luis N Vicente. *Introduction to derivative-free optimization*. SIAM, 2009.

[54] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley, Chichester, UK, 2001.

[55] Kalyanmoy Deb and Himanshu Jain. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints. *IEEE transactions on evolutionary computation*, 18(4):577–601, 2013.

[56] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

[57] Anton Dekkers and Emile Aarts. Global optimization and simulated annealing. *Mathematical programming*, 50(1-3):367–393, 1991.

[58] Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. Sequential monte carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, 2006.

[59] Arnaud Denjoy. Sur les fonctions dérivées sommables. *Bulletin de la Société Mathématique de France*, 43:161–248, 1915.

[60] Youssef Diouane, Serge Gratton, and Luís Nunes Vicente. Globally convergent evolution strategies. *Mathematical Programming*, 152(1):467–490, 2015.

[61] Paul Dufossé and Cheikh Touré. Benchmarking MO-CMA-ES and COMO-CMA-ES on the bi-objective bbob-biobj Testbed. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1920–1927, 2019.

[62] J Dutta, JE Martinez-Legaz, and AM Rubinov. Monotonic analysis over cones: I. *Optimization*, 53(2):129–146, 2004.

[63] Michael Emmerich, Nicola Beume, and Boris Naujoks. An emo algorithm using the hypervolume measure as selection criterion. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 62–76, Guanajuato, Mexico, 2005. Springer.

[64] Michael Emmerich and Jan-willem Klinkenberg. The computation of the expected improvement in dominated hypervolume of pareto front approximations. Technical Report 4-2008, Leiden Institute of Advanced Computer Science, LIACS, 2008.

[65] Michael TM Emmerich and André H Deutz. Test problems based on Lamé superspheres. In *Evolutionary Multi-Criterion Optimization (EMO 2007)*, pages 922–936. Springer, 2007.

[66] Mark Fleischer. The measure of pareto optima applications to multi-objective metaheuristics. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 519–533. Springer, 2003.

[67] C. M. Fonseca and Peter J. Fleming. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In Stephanie Forrest, editor, *Conference on Genetic Algorithms*, pages 416–423, San Mateo, California, 1993. Morgan Kaufmann.

[68] Hervé Fournier and Olivier Teytaud. Lower bounds for comparison based evolution strategies using vc-dimension and sign patterns. *Algorithmica*, 59(3):387–408, 2011.

[69] Tobias Glasmachers. Global convergence of the (1+ 1) evolution strategy to a critical point. *Evolutionary computation*, 28(1):27–53, 2020.

[70] Tobias Glasmachers, Tom Schaul, Sun Yi, Daan Wierstra, and Jürgen Schmidhuber. Exponential natural evolution strategies. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pages 393–400, 2010.

[71] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, Massachusetts, 1989.

[72] David E Goldberg. *Genetic algorithms*. Pearson Education India, 2006.

[73] David E Goldenberg. Genetic algorithms in search, optimization and machine learning, 1989.

[74] Valentin V Gorokhovik and Marina Trafimovich. Positively homogeneous functions revisited. *Journal of Optimization Theory and Applications*, 171(2):481–503, 2016.

[75] Valentin V Gorokhovik and Marina Trafimovich. Saddle representations of positively homogeneous functions by linear functions. *Optimization Letters*, 12(8):1971–1980, 2018.

[76] Oleg Grodzevich and Oleksandr Romanko. Normalization and other topics in multi-objective optimization. In *Proceedings of the Fields-MITACS Industrial Problems Workshop, 2006*, pages 89–101. Fields-MITACS, 2006.

[77] Bruce Hajek. Hitting-time and occupation-time bounds implied by drift analysis with applications. *Advances in Applied probability*, pages 502–525, 1982.

[78] M. P. Hansen and A. Jaszkiewicz. Evaluating The Quality of Approximations of the Non-Dominated Set. Technical report, Institute of Mathematical Modeling, Technical University of Denmark, 1998. IMM Technical Report IMM-REP-1998-7.

[79] N. Hansen, A Auger, D. Brockhoff, D. Tušar, and T. Tušar. COCO: Performance assessment. *ArXiv e-prints*, arXiv:1605.03560, 2016.

[80] N. Hansen, A. Auger, O. Mersmann, T. Tušar, and D. Brockhoff. COCO: A platform for comparing continuous optimizers in a black-box setting. *ArXiv e-prints*, arXiv:1603.08785, 2016.

[81] N. Hansen, T. Tušar, O. Mersmann, A. Auger, and D. Brockhoff. COCO: The experimental procedure. *ArXiv e-prints*, arXiv:1603.08776, 2016.

[82] Nikolaus Hansen. Verallgemeinerte individuelle schrittweitenregelung in der evolutionsstrategie. *Mensch & Buch Verlag, Berlin*, 1998.

[83] Nikolaus Hansen. Invariance, self-adaptation and correlated mutations in evolution strategies. In *International Conference on Parallel Problem Solving from Nature*, pages 355–364. Springer, 2000.

[84] Nikolaus Hansen. An analysis of mutative σ-self-adaptation on linear fitness functions. *Evolutionary computation*, 14(3):255–275, 2006.

[85] Nikolaus Hansen. The cma evolution strategy: a comparing review. In *Towards a new evolutionary computation*, pages 75–102. Springer, 2006.

[86] Nikolaus Hansen. CMA-ES with two-point step-size adaptation. *arXiv preprint arXiv:0805.0231*, 2008.

[87] Nikolaus Hansen. Cma-es: A function value free second order optimization method, 2014.

[88] Nikolaus Hansen. The cma evolution strategy: A tutorial. *arXiv preprint arXiv:1604.00772*, 2016.

[89] Nikolaus Hansen, Youhei Akimoto, and Petr Baudis. CMA-ES/pycma on Github. Zenodo, DOI:10.5281/zenodo.2559634, February 2019.

[90] Nikolaus Hansen, Dirk V Arnold, and Anne Auger. Evolution strategies. In *Springer handbook of computational intelligence*, pages 871–898. Springer, Berlin, 2015.

[91] Nikolaus Hansen and Anne Auger. Principled design of continuous stochastic search: From theory to practice. In *Theory and principled methods for the design of metaheuristics*, pages 145–180. Springer, 2014.

[92] Nikolaus Hansen, Fabian Gemperle, Anne Auger, and Petros Koumoutsakos. When do heavy-tail distributions help? In *Parallel Problem Solving from Nature-PPSN IX*, pages 62–71. Springer, 2006.

[93] Nikolaus Hansen and Stefan Kern. Evaluating the cma evolution strategy on multimodal test functions. In *International Conference on Parallel Problem Solving from Nature*, pages 282–291. Springer, 2004.

[94] Nikolaus Hansen, Sibylle D Müller, and Petros Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary computation*, 11(1):1–18, 2003.

[95] Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation*, 9(2):159–195, 2001.

[96] Nikolaus Hansen, Raymond Ros, Nikolas Mauny, Marc Schoenauer, and Anne Auger. Impacts of invariance in search: When CMA-ES and PSO face ill-conditioned and non-separable problems. *Applied Soft Computing*, 11(8):5755–5769, 2011.

[97] Godefroy Harold Hardy. Weierstrass's non-differentiable function. *Trans. Amer. Math. Soc*, 17(3):301–325, 1916.

[98] W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. *Oxford University Press*, 1970.

[99] VAS Hernandez, O Schutze, H Wang, A Deutz, and M Emmerich. The set-based hypervolume newton method for bi-objective optimization. *IEEE transactions on cybernetics*, in print, 2018. (in print).

[100] Ashley Hill, Eric Lucet, and Roland Lenain. Neuroevolution with CMA-ES for real-time gain tuning of a car-like robot controller. In *ICINCO (1)*, pages 311–319, 2019.

[101] John H Holland. Genetic algorithms. *Scientific american*, 267(1):66–73, 1992.

[102] J. Horn, N. Nafpliotis, and D. E. Goldberg. A Niched Pareto Genetic Algorithm for Multiobjective Optimization. In *Congress on Evolutionary Computation (CEC 1994*, pages 82–87. IEEE Press, 1994.

[103] Yuguang Huang and WF McColl. An improved simplex method for function minimization. In *1996 IEEE International Conference on Systems, Man and Cybernetics. Information Intelligence and Systems (Cat. No. 96CH35929)*, volume 3, pages 1702–1705. IEEE, 1996.

[104] C. Igel, N. Hansen, and S. Roth. Covariance matrix adaptation for multi-objective optimization. *Evolutionary Computation*, 15(1):1–28, 2007.

[105] Christian Igel, Thorsten Suttorp, and Nikolaus Hansen. A computational efficient covariance matrix update and a (1+ 1)-CMA for evolution strategies. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 453–460, 2006.

[106] Jens Jägersküpper. Analysis of a simple evolutionary algorithm for minimization in euclidean spaces. In *International Colloquium on Automata, Languages, and Programming*, pages 1068–1079. Springer, 2003.

[107] Jens Jägersküpper. Rigorous runtime analysis of the (1+ 1)-ES: 1/5-rule and ellipsoidal fitness landscapes. In *International Workshop on Foundations of Genetic Algorithms*, pages 260–281. Springer, 2005.

[108] Jens Jägersküpper. How the (1+ 1)-ES using isotropic mutations minimizes positive definite quadratic forms. *Theoretical Computer Science*, 361(1):38–56, 2006.

[109] Jens Jägersküpper. Probabilistic runtime analysis of (1+, λ)-ES using isotropic mutations. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 461–468, 2006.

[110] Jens Jägersküpper. Algorithmic analysis of a basic evolutionary algorithm for continuous optimization. *Theoretical Computer Science*, 379(3):329–347, 2007.

[111] Jens Jägersküpper. Lower bounds for randomized direct search with isotropic sampling. *Operations research letters*, 36(3):327–332, 2008.

[112] J Jahn. Vector optimization: Theory, applications and extensions. 2004.

[113] Himanshu Jain and Kalyanmoy Deb. An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part ii: handling constraints and extending to an adaptive approach. *IEEE Transactions on evolutionary computation*, 18(4):602–622, 2013.

[114] Kevin G Jamieson, Robert D Nowak, and Benjamin Recht. Query complexity of derivative-free optimization. *arXiv preprint arXiv:1209.2434*, 2012.

[115] Grahame A Jastrebski and Dirk V Arnold. Improving evolution strategies through active covariance matrix adaptation. In *2006 IEEE international conference on evolutionary computation*, pages 2814–2821. IEEE, 2006.

[116] Mohamed Jebalia, Anne Auger, and Nikolaus Hansen. Log-linear convergence and divergence of the scale-invariant (1+ 1)-ES in noisy environments. *Algorithmica*, 59(3):425–460, 2011.

[117] Mohamed Jebalia, Anne Auger, and Pierre Liardet. Log-linear convergence and optimal bounds for the (1+ 1)-ES. In *International Conference on Artificial Evolution (Evolution Artificielle)*, pages 207–218. Springer, 2007.

[118] Søren Tolver Jensen and Anders Rahbek. On the law of large numbers for (geometrically) ergodic markov chains. *Econometric Theory*, pages 761–766, 2007.

[119] Cornelia Kappler. Are evolutionary algorithms improved by large mutations? In *International Conference on Parallel Problem Solving from Nature–PPSN IV*, pages 346–355. Springer, 1996.

[120] Andy J. Keane. Statistical improvement criteria for use in multiobjective design optimization. *AIAA journal*, 44(4):879–891, 2006.

[121] Stefan Kern, Sibylle D Müller, Nikolaus Hansen, Dirk Büche, Jiri Ocenasek, and Petros Koumoutsakos. Learning probability distributions in continuous evolutionary algorithms–a comparative review. *Natural Computing*, 3(1):77–112, 2004.

[122] P Kerschke, H Wang, M Preuss, C Grimme, AH Deutz, H Trautmann, and MTM Emmerich. Search dynamics on multimodal multi-objective problems. *Evolutionary computation*, pages 1–30, 2018.

[123] Pascal Kerschke, Hao Wang, Mike Preuss, Christian Grimme, André Deutz, Heike Trautmann, and Michael Emmerich. Towards analyzing multimodality of continuous multiobjective landscapes. In *Parallel Problem Solving from Nature (PPSN 2016)*, pages 962–972. Springer, 2016.

[124] Nazan Khan, David E Goldberg, and Martin Pelikan. Multi-objective bayesian optimization algorithm. In *Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation*, pages 684–684. Citeseer, 2002.

[125] J. Knowles and D. Corne. Properties of an Adaptive Archiving Algorithm for Storing Nondominated Vectors. *IEEE Transactions on Evolutionary Computation*, 7(2):100–116, 2003.

[126] J. Knowles, L. Thiele, and E. Zitzler. A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers. TIK Report 214, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, February 2006.

[127] J. D. Knowles, D. W. Corne, and M. Fleischer. Bounded Archiving using the Lebesgue Measure. In *Congress on Evolutionary Computation (CEC 2003)*, pages 2490–2497. IEEE Press, 2003.

[128] Oswin Krause, Tobias Glasmachers, Nikolaus Hansen, and Christian Igel. Unbounded population mo-cma-es for the bi-objective bbob test suite. In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion*, pages 1177–1184. ACM, 2016.

[129] Oswin Krause, Tobias Glasmachers, and Christian Igel. Qualitative and quantitative assessment of step size adaptation rules. In *Proceedings of the 14th ACM/SIGEVO Conference on Foundations of Genetic Algorithms*, pages 139–148, Copenhagen, Denmark, 2017. ACM.

[130] Marek Kuczma. *An introduction to the theory of functional equations and inequalities: Cauchy's equation and Jensen's inequality*. Springer Science & Business Media, 2009.

[131] Harold Kushner and G George Yin. *Stochastic approximation and recursive algorithms and applications*, volume 35. Springer Science & Business Media, 2003.

[132] JB Lasserre and JB Hiriart-Urruty. Mathematical properties of optimization problems defined by positively homogeneous functions. *Journal of optimization theory and applications*, 112(1):31–52, 2002.

[133] Steffen Limmer and Dietmar Fey. Investigation of strategies for an increasing population size in multi-objective CMA-ES. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 476–483, Vancouver, BC, Canada, July 2016. IEEE.

[134] Lennart Ljung. Analysis of recursive stochastic algorithms. *IEEE transactions on automatic control*, 22(4):551–575, 1977.

[135] Marco Locatelli. Simulated annealing algorithms for continuous global optimization: convergence conditions. *Journal of Optimization Theory and applications*, 104(1):121–133, 2000.

[136] Ilya Loshchilov and Tobias Glasmachers. Anytime bi-objective optimization with a hybrid multi-objective cma-es (hmo-cma-es). In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion*, pages 1169–1176. ACM, 2016.

[137] Ken IM McKinnon. Convergence of the nelder–mead simplex method to a nonstationary point. *SIAM Journal on optimization*, 9(1):148–158, 1998.

[138] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

[139] Sean P Meyn and PE Caines. Asymptotic behavior of stochastic systems possessing markovian realizations. *SIAM journal on control and optimization*, 29(3):535–561, 1991.

[140] Sean P Meyn and Richard L Tweedie. *Markov chains and stochastic stability*. Springer Science & Business Media, 2012.

[141] K. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer, Boston, MA, USA, 1999.

[142] Jorge J Moré. The levenberg-marquardt algorithm: implementation and theory. In *Numerical analysis*, pages 105–116. Springer, 1978.

[143] Daiki Morinaga and Youhei Akimoto. Generalized drift analysis in continuous domain: linear convergence of (1+1)-ES on strongly convex functions with lipschitz continuous gradients. In *Proceedings of the 15th ACM/SIGEVO Conference on Foundations of Genetic Algorithms*, pages 13–24, 2019.

[144] Daiki Morinaga, Kazuto Fukuchi, Jun Sakuma, and Youhei Akimoto. Convergence rate of the (1+ 1)-evolution strategy with success-based step-size adaptation on convex quadratic functions. *arXiv preprint arXiv:2103.01578*, 2021.

[145] Marian Muresan. *A concrete approach to classical analysis*, volume 14. Springer, 2009.

[146] John A Nelder and Roger Mead. A simplex method for function minimization. *The computer journal*, 7(4):308–313, 1965.

[147] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2003.

[148] Jorge Nocedal and Stephen J Wright. Numerical optimization, 2006.

[149] Yann Ollivier, Ludovic Arnold, Anne Auger, and Nikolaus Hansen. Information-geometric optimization algorithms: A unifying picture via invariance principles. *The Journal of Machine Learning Research*, 18(1):564–628, 2017.

[150] Andreas Ostermeier, Andreas Gawelczyk, and Nikolaus Hansen. A derandomized approach to self-adaptation of evolution strategies. *Evolutionary Computation*, 2(4):369–380, 1994.

[151] Ali Ouni, Marouane Kessentini, Houari Sahraoui, Mel Ó Cinnéide, Kalyanmoy Deb, and Katsuro Inoue. A multi-objective refactoring approach to introduce design patterns and fix anti-patterns. In *North American Search Based Software Engineering Symposium (NASBaSE)*, pages 1–16, 2015.

[152] Wolfgang Ponweiser, Tobias Wagner, Dirk Biermann, and Markus Vincze. Multiobjective optimization on a limited budget of evaluations using model-assisted -metric selection. In *Parallel Problem Solving from Nature (PPSN 2008)*, pages 784–794, Dortmund, Germany, 2008. Springer.

[153] Michael JD Powell. Uobyqa: unconstrained optimization by quadratic approximation. *Mathematical Programming*, 92(3):555–582, 2002.

[154] Michael JD Powell. On trust region methods for unconstrained minimization without derivatives. *Mathematical programming*, 97(3):605–623, 2003.

[155] Michael JD Powell. Least frobenius norm updating of quadratic models that satisfy interpolation conditions. *Mathematical Programming*, 100(1):183–215, 2004.

[156] Michael J.D. Powell. The NEWUOA software for unconstrained optimization without derivatives. Technical Report DAMTP 2004/NA05, CMS, University of Cambridge, Cambridge CB3 0WA, UK, November 2004.

[157] Michael JD Powell. The newuoa software for unconstrained optimization without derivatives. In *Large-scale nonlinear optimization*, pages 255–297. Springer, 2006.

[158] MJD Powell. On the lagrange functions of quadratic models that are defined by interpolation. *Optimization Methods and Software*, 16(1-4):289–309, 2001.

[159] I. Rechenberg. Optimierung technischer Systeme nach Prinzipien der biologischen Evolution Dr.-Ing. Dissertation. Technical report, Verlag Frommann-Holzboog, Stuttgart-Bad Cannstatt, 1973.

[160] Ingo Rechenberg. *Evolutionsstrategie'94*. frommann-holzboog, 1994.

[161] Raymond Ros. Comparison of newuoa with different numbers of interpolation points on the bbob noisy testbed. In *Proceedings of the 12th annual conference companion on Genetic and evolutionary computation*, pages 1495–1502, 2010.

[162] Raymond Ros and Nikolaus Hansen. A simple modification in CMA-ES achieving linear time and space complexity. In *International Conference on Parallel Problem Solving from Nature*, pages 296–305. Springer, 2008.

[163] AM Rubinov and RN Gasimov. Strictly increasing positively homogeneous functions with application to exact penalization. *Optimization*, 52(1):1–28, 2003.

[164] AM Rubinov and BM Glover. Duality for increasing positively homogeneous functions and normal sets. *RAIRO-Operations Research-Recherche Opérationnelle*, 32(2):105–123, 1998.

[165] Günter Rudolph. *Convergence properties of evolutionary algorithms*. Verlag Dr. Kovač, 1997.

[166] Pole In C Ryan et al. References to CMA-ES applications. *Strategies*, 4527(467), 2007.

[167] Ralf Salomon. Re-evaluating genetic algorithm performance under coordinate rotation of benchmark functions. a survey of some theoretical and practical aspects of genetic algorithms. *BioSystems*, 39(3):263–278, 1996.

[168] Ralf Salomon. Evolutionary algorithms and gradient search: similarities and differences. *IEEE Transactions on Evolutionary Computation*, 2(2):45–55, 1998.

[169] JD SCHAFFER. Some experiments in machini learing using vector evaluated genetic algorithms. *doctoral dissertation, Vanderbilt University*, 1984.

[170] Tom Schaul. Natural evolution strategies converge on sphere functions. In *Proceedings of the 14th annual conference on Genetic and evolutionary computation*, pages 329–336, 2012.

[171] Tom Schaul, Tobias Glasmachers, and Jürgen Schmidhuber. High dimensions and heavy tails for natural evolution strategies. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 845–852, 2011.

[172] H.-P. Schwefel. *Evolution and Optimum Seeking*. Sixth-Generation Computer Technology Series. John Wiley & Sons, Inc., New York, 1995.

[173] Hans-Paul Schwefel. *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie: mit einer vergleichenden Einführung in die Hill-Climbing-und Zufallsstrategie*, volume 1. Springer, 1977.

[174] Hans-Paul Schwefel and Günter Rudolph. Contemporary evolution strategies. In *European conference on artificial life*, pages 891–907. Springer, 1995.

[175] Andrew Sohn, Randal S Olson, and Jason H Moore. Toward the automated analysis of complex diseases in genome-wide association studies using genetic programming. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 489–496, 2017.

[176] Ralf Solomon and J Leo Van Hemmen. Accelerating backpropagation through dynamic self-adaptation. *Neural Networks*, 9(4):589–601, 1996.

[177] Danny C Sorensen. Newton?s method with a model trust region modification. *SIAM Journal on Numerical Analysis*, 19(2):409–426, 1982.

[178] Nidamarthi Srinivas and Kalyanmoy Deb. Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary computation*, 2(3):221–248, 1994.

[179] William F Stout and William F Stout. *Almost sure convergence*, volume 24. Academic press, 1974.

[180] Thorsten Suttorp, Nikolaus Hansen, and Christian Igel. Efficient covariance matrix update for variable metric evolution strategies. *Machine Learning*, 75(2):167–197, 2009.

[181] Ryoji Tanabe and Hisao Ishibuchi. Non-elitist evolutionary multi-objective optimizers revisited. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 612–619, 2019.

[182] Olivier Teytaud and Sylvain Gelly. General lower bounds for evolutionary algorithms. In *Parallel Problem Solving from Nature-PPSN IX*, pages 21–31. Springer, 2006.

[183] Cheikh Toure, Anne Auger, Dimo Brockhoff, and Nikolaus Hansen. On bi-objective convex-quadratic problems. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 3–14, Lansing, Michigan, USA, 2019. Springer.

[184] Cheikh Toure, Armand Gissler, Anne Auger, and Nikolaus Hansen. Scaling-invariant functions versus positively homogeneous functions. *Journal of Optimization Theory and Applications*, 191(1):363–383, 2021.

[185] Cheikh Touré, Nikolaus Hansen, Anne Auger, and Dimo Brockhoff. Uncrowded hypervolume improvement: COMO-CMA-ES and the Sofomore framework. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 638–646, 2019.

[186] Heike Trautmann, Tobias Wagner, and Dimo Brockhoff. R2-emoa: Focused multiobjective search using r2-indicator-based selection. In *International Conference on Learning and Intelligent Optimization*, pages 70–74. Springer, 2013.

[187] T. Tušar, D. Brockhoff, N. Hansen, and A. Auger. COCO: The bi-objective black-box optimization benchmarking (bbob-biobj) test suite. *ArXiv e-prints*, arXiv:1604.00359, 2016.

[188] T. Tušar, D. Brockhoff, N. Hansen, and A. Auger. COCO: the bi-objective black box optimization benchmarking (bbob-biobj) test suite. *CoRR*, abs/1604.00359, 2016.

[189] RL Tweedie. The existence of moments for stationary markov chains. *Journal of Applied Probability*, 20(1):191–196, 1983.

[190] Guido van Rossum and Fred L Drake. pickle–python object serialization. *Available from World Wide Web: http://docs. python. org/lib/module-pickle. html*, 2009.

[191] Konstantinos Varelas, Anne Auger, Dimo Brockhoff, Nikolaus Hansen, Ouassim Ait ElHara, Yann Semet, Rami Kassab, and Frédéric Barbaresco. A comparative study of large-scale variants of cma-es. In *International Conference on Parallel Problem Solving from Nature*, pages 3–15. Springer, 2018.

[192] Yash Vesikar, Kalyanmoy Deb, and Julian Blank. Reference point based nsga-iii for preferred solutions. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1587–1594. IEEE, 2018.

[193] T. Voß, N. Hansen, and C. Igel. Improved Step Size Adaptation for the MO-CMA-ES. In J. Branke et al., editors, *Genetic and Evolutionary Computation Conference (GECCO 2010)*, pages 487–494, Portland, OR, USA, 2010. ACM.

[194] Tobias Wagner, Michael Emmerich, André Deutz, and Wolfgang Ponweiser. On expected-improvement criteria for model-based multi-objective optimization. In *International Conference on Parallel Problem Solving from Nature*, pages 718–727, Krakow, Poland, 2010. Springer.

[195] Tobias Wagner and Heike Trautmann. Online convergence detection for evolutionary multi-objective algorithms revisited. In *IEEE Congress on Evolutionary Computation*, pages 1–8, Barcelona, Spain, 2010. IEEE.

[196] Simon Wessing. evoalgos: Modular evolutionary algorithms. python package version 1, 2017. [Online; accessed 31-January-2019].

[197] Margaret H Wright. Direct search methods: Once scorned, now respectable. *Pitman Research Notes in Mathematics Series*, pages 191–208, 1996.

[198] Kaifeng Yang, Michael Emmerich, André Deutz, and Thomas Bäck. Multi-objective bayesian global optimization using expected hypervolume improvement gradient. *Swarm and evolutionary computation*, 44:945–956, 2019.

[199] Xin Yao and Yong Liu. Fast evolution strategies. In *International Conference on Evolutionary Programming*, pages 149–161. Springer, 1997.

[200] Xinjie Yu and Mitsuo Gen. *Introduction to evolutionary algorithms*. Springer Science & Business Media, 2010.

[201] Q. Zhang and H. Li. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, 2007.

[202] Anatoly Zhigljavsky and Antanasz Zilinskas. *Stochastic global optimization*, volume 9. Springer Science & Business Media, 2007.

[203] E. Zitzler, J. Knowles, and L. Thiele. Quality Assessment of Pareto Set Approximations. In J. Branke, K. Deb, K. Miettinen, and R. Slowinski, editors, *Multiobjective Optimization: Interactive and Evolutionary Approaches*, pages 373–404. Springer, 2008.

[204] Eckart Zitzler and Simon Künzli. Indicator-based selection in multiobjective search. In *International Conference on Parallel Problem Solving from Nature*, pages 832–842, Birmingham, UK, 2004. Springer.

[205] Eckart Zitzler and Lothar Thiele. Multiobjective optimization using evolutionary algorithms - a comparative case study. In *International conference on parallel problem solving from nature*, pages 292–301, Amsterdam, The Netherlands, 1998. Springer.

[206] Eckart Zitzler, Lothar Thiele, Marco Laumanns, Carlos M Fonseca, and Viviane Grunert Da Fonseca. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on evolutionary computation*, 7(2):117–132, 2003.

Appendix A

Supplements in single-objective analysis

1.1 Bijection Theorem

This standard theorem is reminded for the sake of completeness.
Theorem 14 (Bijection theorem, [145, Theorem 2.20])**.** *Let $I \subset \mathbb{R}$ be a nontrivial interval, $J \subset \mathbb{R}$ and $\varphi : I \to J$ be a continuous bijection (and therefore strictly monotonic). Then J is an interval and φ is a homeomorphism, $i.e.$ $\varphi^{-1} : J \to I$ is also a continuous bijection, and if φ is strictly increasing (respectively strictly decreasing), then φ^{-1} is strictly increasing (respectively strictly decreasing).*

1.2 Proof of Proposition 17

With Lemma 5, we assume without loss of generality that f is a nontrivial linear function. Let us remark beforehand that the random variable $\alpha_f(z, U_1)$ does not depend on z thanks to Lemma 8. Let $\varphi : \Gamma(\mathbb{R}^{n\mu}) \to \mathbb{R}$ be a continuous and bounded function, it is then enough to prove that $\mathbb{E}_{U_1 \sim \mathcal{N}_{n\lambda}} \left[\varphi \left(\Gamma(\alpha_f(z, U_1)) \right) \right] = \mathbb{E}_{U_1 \sim \mathcal{N}_{n\lambda}} \left[\varphi \left(\Gamma(\alpha_{l^\star}(0, U_1)) \right) \right]$. Denote by e_1 the unit vector $(1, 0, \ldots, 0)$, then for all $x \in \mathbb{R}^n$, $l^\star(x) = e_1^\intercal x$. Denote by $\tilde{e}_1$ the σ-normalized gradient of f at some point. Then there exists $K > 0$ such that for all $x \in \mathbb{R}^n$, $f(x) = K \tilde{e}_1^\intercal x$. And by the Gram-Schmidt process, there exist $(e_2, \ldots, e_n)$ and $(\tilde{e}_2, \ldots, \tilde{e}_n)$ such that $(e_1, e_2, \ldots, e_n)$ and $(\tilde{e}_1, \tilde{e}_2, \ldots, \tilde{e}_n)$ are orthonormal bases. Denote by T the linear function defined as $T(e_i) = \tilde{e}_i$ for $i = 1, \ldots, n$. Then T is an orthogonal matrix. For all $x \in \mathbb{R}^n$,

$$\tilde{e}_1^\intercal T(x) = e_1^\intercal x, \text{ and } \|T(x)\| = \|x\|. \tag{A.1}$$

Denote $A = \mathbb{E}_{U_1 \sim \mathcal{N}_{n\lambda}} \left[\varphi \left(\Gamma(\alpha_f(z, U_1)) \right) \right]$.

We do a change of variable $u \mapsto \left(T(u^1), \ldots, T(u^\mu) \right)$ and incorporate the results of (A.1): $\frac{(\lambda-\mu)!}{\lambda!} A = \int \varphi \left(\Gamma(u) \right) \mathbb{1}_{\tilde{e}_1^\top (u^2 - u^1) > 0, \ldots, \tilde{e}_1^\top (u^\mu - u^{\mu-1}) > 0} P \left(\tilde{e}_1^\top \mathcal{N}_n > \tilde{e}_1^\top u^\mu \right)^{\lambda - \mu} p_{\mathcal{N}_n}(u^1) \ldots p_{\mathcal{N}_n}(u^\mu) \mathrm{d}u^1 \ldots \mathrm{d}u^\mu = \int \varphi \left(\Gamma \left(T(u^1), \ldots, T(u^\mu) \right) \right) \mathbb{1}_{e_1^\top (u^2 - u^1) > 0, \ldots, e_1^\top (u^\mu - u^{\mu-1}) > 0} P \left(e_1^\top \mathcal{N}_n > e_1^\top u^\mu \right)^{\lambda - \mu} p_{\mathcal{N}_n}(T(u^1)) \ldots p_{\mathcal{N}_n}(T(u^\mu)) \mathrm{d}u^1 \ldots \mathrm{d}u^\mu$, thanks to the fact that $e_1^\top \mathcal{N}_n \sim \tilde{e}_1^\top \mathcal{N}_n \sim \mathcal{N}(0, 1)$. Since Γ and $p_{\mathcal{N}_n}$ are invariant under rotation, it follows that

$$\mathbb{E}_{U_1 \sim \mathcal{N}_{n\lambda}} \left[\varphi \left(\Gamma(\alpha_f(z, U_1)) \right) \right] = \mathbb{E}_{U_1 \sim \mathcal{N}_{n\lambda}} \left[\varphi \left(\Gamma(\alpha_{l^\star}(0, U_1)) \right) \right].$$

1.3 Proof of Proposition 20

We have $Z_{k+1} = G(Z_k, U_{k+1})$ and U_{k+3} is independent from $\{W_t ; t \leq k\}$, therefore $\{W_k ; k \in \mathbb{N}\}$ is a Markov chain on $\mathcal{B}(\mathcal{Z}) \otimes \mathcal{B}(\mathbb{R}^m)$. Let $(A, B) \in \mathcal{B}(\mathcal{Z}) \times \mathcal{B}(\mathbb{R}^m)$ and $(z, u) \in \mathcal{Z} \times \mathbb{R}^m$. Then by independence

$$P \left((Z_{t+1}, U_{t+3}) \in A \times B | (Z_t, U_{t+2}) = (z, u) \right) = P \left(Z_{t+1} \in A | Z_t = z \right) P \left(U_{t+3} \in B \right).$$

For $(A, B) \in \mathcal{B}(\mathcal{Z}) \times \mathcal{B}(\mathbb{R}^m)$, for $(z, u) \in \mathcal{Z} \times \mathbb{R}^m$, $\sum_{k=1}^\infty P^k((z, u), A \times B) = \Psi(B) \sum_{k=1}^\infty P^k(z, A)$. Therefore $\sum_{k=1}^\infty P^k((z, u), \cdot)$ is a product measure.

Let φ be an irreducible measure of $\{Z_k ; k \in \mathbb{N}\}$ and let $E \in \mathcal{B}(\mathcal{Z}) \otimes \mathcal{B}(\mathbb{R}^m)$. By definition of a product measure,

$$(\varphi \times \Psi)(E) = \int \varphi(E^v) \Psi(\mathrm{d}v)$$
$$\sum_{k=1}^\infty P^k((z, u), E) = \int \sum_{k=1}^\infty P^k(z, E^v) \Psi(\mathrm{d}v)$$

where $E^v = \{z \in \mathcal{Z} ; (z, v) \in E\}$.

If $\sum_{k=1}^\infty P^k((z, u), E) = 0$, then $0 = \sum_{k=1}^\infty P^k(z, E^v)$ for almost all v and then $\varphi(E^v) = 0$ for almost all v. Then $(\varphi \times \Psi)(E) = \int \varphi(E^v) \Psi(\mathrm{d}v) = 0$, hence the $(\varphi \times \Psi)$-irreducibility of $\{W_k ; k \in \mathbb{N}\}$.

Let us show that $\pi \times \Psi$ is an invariant probability measure of $\{W_k ; k \in \mathbb{N}\}$ when π is an invariant measure of $\{Z_k ; k \in \mathbb{N}\}$. Assume that $(A, B) \in \mathcal{B}(\mathcal{Z}) \times \mathcal{B}(\mathbb{R}^m)$. Then

$$\int P \left((Z_1, U_3) \in A \times B | (Z_0, U_2) = (z, u) \right) (\pi \times \Psi) \left(\mathrm{d}(z, u) \right) =$$
$$\int P_z \left(Z_1 \in A \right) \Psi(B) \pi(\mathrm{d}z) \Psi(\mathrm{d}u) = \Psi(B) \pi(A) = (\pi \times \Psi)(A \times B).$$

Hence $\pi \times \Psi$ is an invariant probability of $\{W_k ; k \in \mathbb{N}\}$.

Assume that $\{W_k\,;k\in\mathbb{N}\}$ has a d-cycle $(D_i)_{i=1,\dots,d}\in(\mathcal{B}(\mathcal{Z})\otimes\mathcal{B}(\mathbb{R}^m))^d$. Define for $i=1,\dots,d$, $\widetilde{D}_i=\{z\in\mathcal{Z}|\exists\, u\in\mathbb{R}^m\,;(z,u)\in D_i\}$ and let us prove that $\left(\widetilde{D}_i\right)_{i=1,\dots,d}$ is a d-cycle of $\{Z_k\,;k\in\mathbb{N}\}$.

Let $z\in\widetilde{D}_i$ and $i=0,\dots,d-1$ (mod d). There exists $u\in\mathbb{R}^m$ such that $(z,u)\in D_i$. Then $1=P((z,u),D_{i+1})=P\left((Z_1,U_3)\in D_{i+1}|Z_0=z\right)\leq P\left(Z_1\in\widetilde{D}_{i+1}|Z_0=z\right)$. Thereforer $P\left(Z_1\in\widetilde{D}_{i+1}|Z_0=z\right)=1$.

If Λ is an irreducible measure of $\{Z_k\,;k\in\mathbb{N}\}$, then we have proven above that $\Lambda\times\Psi$ is an irreducible measure of $\{W_k\,;k\in\mathbb{N}\}$. Then $0=(\Lambda\times\Psi)\left(\left(\bigcup_{i=1}^d D_i\right)^c\right)$. For $i=1,\dots,d$, $(\Lambda\times\Psi)(D_i)=\displaystyle\int\Lambda(D_i^v)\Psi(\mathrm{d}v)\leq\int\Lambda(\widetilde{D}_i)\Psi(\mathrm{d}v)=\Lambda(\widetilde{D}_i)$. Then $\Lambda\left(\bigcup_{i=1}^d\widetilde{D}_i\right)=\sum_{i=1}^d\Lambda(\widetilde{D}_i)\geq(\Lambda\times\Psi)\left(\bigcup_{i=1}^d D_i\right)$. Hence $\Lambda\left(\left(\bigcup_{i=1}^d\widetilde{D}_i\right)^c\right)=0$ and finally we have a d-cycle for $\{Z_k\,;k\in\mathbb{N}\}$.

Similarly we can show that if $\{Z_k\,;k\in\mathbb{N}\}$ has a d-cycle, then $\{W_k\,;k\in\mathbb{N}\}$ also has a d-cycle.

Now assume that C is a small set of $\{Z_k\,;k\in\mathbb{N}\}$. Then there exists a positive integer k and a nontrivial measure ν_k on $\mathcal{B}(\mathcal{Z})$ such that $P^k(z,A)\geq\nu_k(A)$ for all $z\in C$, $A\in\mathcal{B}(\mathcal{Z})$. If $(z,u)\in C\times\mathbb{R}^m$ and $E\in\mathcal{B}(\mathcal{Z})\otimes\mathcal{B}(\mathbb{R}^m)$, $P^k((z,u),E)\geq(\nu_k\times\Psi)(E)$ and therefore $C\times\mathbb{R}^m$ is a small set of $\{W_k\,;k\in\mathbb{N}\}$.

The drift condition for $\{W_k\,;k\in\mathbb{N}\}$ follows directly from the drift condition for $\{Z_k\,;k\in\mathbb{N}\}$.

1.4 Proof of Proposition 25

To prove the convergence in distribution of the step-size multiplicative factor for a function f that satisfies F1 or F2, we use the intermediate result given by Proposition 25, that asymptotically links $\Gamma\left(\alpha_f(x^\star+z,U_1)\right)$ to the random variable $\Gamma\left(\alpha_{l_z^f}(z,U_1)\right)$ where the nontrivial linear function l_z^f depends on z, ∇f, and is introduced in (4.35). Since $\alpha_f(x^\star+z,U_1)=\alpha_{\tilde{f}}(z,U_1)$, we assume without loss of generality that $x^\star=0$ and $f(0)=0$.

The next lemma is our fist step towards understanding the asymptotic behavior of $\alpha_f(z,U_1)$ for a C^1 scaling-invariant function f with a unique global argmin. For $\varphi:\mathbb{R}^{n\mu}\to\mathbb{R}$ continuous and bounded, we approximate $\mathbb{E}\left[\varphi(\alpha_f(z,U_1))\right]$ by using the explicit definition of p_z^f in (4.15), and observing the integrals in the balls $\overline{\mathbf{B}\left(0,\sqrt{\|z\|}\right)}$, such that the f-values we consider are relatively close to the f-values of $\frac{t_z^f}{\|z\|}z\in\mathcal{L}_{f,z_0}$.

Lemma 16. *Let f be a C^1 scaling-invariant function with a unique global argmin. Then*

for all $\varphi : \mathbb{R}^{n\mu} \to \mathbb{R}$ *continuous and bounded:*

$$\lim_{\|z\|\to\infty} \int_{\|u\|\le\sqrt{\|z\|}} \left(\int_{\|w\|\le\sqrt{\|z\|}} \mathbb{1}_{f\left(\frac{t_z^f}{\|z\|}z+\frac{t_z^f}{\|z\|}w\right)>f\left(\frac{t_z^f}{\|z\|}z+\frac{t_z^f}{\|z\|}u^\mu\right)} p_{\mathcal{N}_n}(w)\mathrm{d}w \right)^{\lambda-\mu}$$
$$\frac{\lambda!}{(\lambda-\mu)!}\varphi(u) \prod_{i=1}^{\mu-1} \mathbb{1}_{\left\{f\left(\frac{t_z^f}{\|z\|}z+\frac{t_z^f}{\|z\|}u^{i+1}\right)>f\left(\frac{t_z^f}{\|z\|}z+\frac{t_z^f}{\|z\|}u^{i}\right)\right\}} \prod_{i=1}^{\mu} p_{\mathcal{N}_n}(u^i)\mathrm{d}u$$
$$- \int \varphi(u) p_z^f(u)\mathrm{d}u = 0.$$

Proof. For $z \in \mathbb{R}^n$, define $A(z) = \left| \int \varphi(u) p_z^f(u)\mathrm{d}u - \frac{\lambda!}{(\lambda-\mu)!} \int_{\|u\|\le\sqrt{\|z\|}} \varphi(u) \right.$
$\left. P\left(f(z+\mathcal{N}_n) > f(z+u^\mu)\right)^{\lambda-\mu} \prod_{i=1}^{\mu-1} \mathbb{1}_{f(z+u^{i+1})>f(z+u^i)} \prod_{i=1}^{\mu} p_{\mathcal{N}_n}(u^i)\mathrm{d}u \right|$. It follows that $A(z) =$
$\frac{\lambda!}{(\lambda-\mu)!} \left| \int_{\|u\|>\sqrt{\|z\|}} \varphi(u) P\left(f(z+\mathcal{N}_n) > f(z+u^\mu)\right)^{\lambda-\mu} \prod_{i=1}^{\mu-1} \mathbb{1}_{f(z+u^{i+1})>f(z+u^i)} \prod_{i=1}^{\mu} p_{\mathcal{N}_n}(u^i)\mathrm{d}u \right|$.

Then $A(z) \le \frac{\lambda!}{(\lambda-\mu)!}\|\varphi\|_\infty \int_{\|u\|>\sqrt{\|z\|}} \prod_{i=1}^{\mu} p_{\mathcal{N}_n}(u^i)\mathrm{d}u = \frac{\lambda!}{(\lambda-\mu)!}\|\varphi\|_\infty$
$\int_{\|u\|>\sqrt{\|z\|}} p_{\mathcal{N}_{n\mu}}(u)\mathrm{d}u = \frac{\lambda!}{(\lambda-\mu)!}\|\varphi\|_\infty \left(1 - P\left(\|\mathcal{N}_{n\mu}\| \le \sqrt{\|z\|}\right)\right)$.

Then by scaling-invariance with a multiplication by $t_z^f/\|z\|$,

$$\lim_{\|z\|\to\infty} \int_{\|u\|\le\sqrt{\|z\|}} \varphi(u) P\left(f\left(\frac{t_z^f}{\|z\|}z+\frac{t_z^f}{\|z\|}\mathcal{N}_n\right) > f\left(\frac{t_z^f}{\|z\|}z+\frac{t_z^f}{\|z\|}u^\mu\right)\right)^{\lambda-\mu}$$
$$\prod_{i=1}^{\mu-1} \mathbb{1}_{f\left(\frac{t_z^f}{\|z\|}z+\frac{t_z^f}{\|z\|}u^{i+1}\right)>f\left(\frac{t_z^f}{\|z\|}z+\frac{t_z^f}{\|z\|}u^{i}\right)} \prod_{i=1}^{\mu} p_{\mathcal{N}_n}(u^i)\mathrm{d}u - \frac{(\lambda-\mu)!}{\lambda!} \int \varphi(u) p_z^f(u)\mathrm{d}u = 0. \quad \text{(A.2)}$$

In addition, $P\left(f\left(\frac{t_z^f}{\|z\|}z+\frac{t_z^f}{\|z\|}\mathcal{N}_n\right) > f\left(\frac{t_z^f}{\|z\|}z+\frac{t_z^f}{\|z\|}u^\mu\right)\right) -$
$\int_{\|w\|\le\sqrt{\|z\|}} \mathbb{1}_{f\left(\frac{t_z^f}{\|z\|}z+\frac{t_z^f}{\|z\|}w\right)>f\left(\frac{t_z^f}{\|z\|}z+\frac{t_z^f}{\|z\|}u^\mu\right)} p_{\mathcal{N}_n}(w)\mathrm{d}w =$
$\int_{\|w\|>\sqrt{\|z\|}} \mathbb{1}_{f\left(\frac{t_z^f}{\|z\|}z+\frac{t_z^f}{\|z\|}w\right)>f\left(\frac{t_z^f}{\|z\|}z+\frac{t_z^f}{\|z\|}u^\mu\right)} p_{\mathcal{N}_n}(w)\mathrm{d}w \le 1 - P\left(\|\mathcal{N}_n\| \le \sqrt{\|z\|}\right)$. Hence (A.2) along with the dominated convergence theorem proves the lemma. □

Proof of Proposition 25. Let $\varphi : \mathbb{R}^{n\mu} \to \mathbb{R}$ be continuous and bounded. Using Lemma 16, it is enough to prove that

$$\lim_{\|z\|\to\infty} \int_{\|u\|\le\sqrt{\|z\|}} \left(\int_{\|w\|\le\sqrt{\|z\|}} \mathbb{1}_{f\left(\frac{t_z^f}{\|z\|}z+\frac{t_z^f}{\|z\|}w\right)>f\left(\frac{t_z^f}{\|z\|}z+\frac{t_z^f}{\|z\|}u^\mu\right)} p_{\mathcal{N}_n}(w)\mathrm{d}w \right)^{\lambda-\mu}$$
$$\varphi(u) \prod_{i=1}^{\mu-1} \mathbb{1}_{f\left(\frac{t_z^f}{\|z\|}z+\frac{t_z^f}{\|z\|}u^{i+1}\right)>f\left(\frac{t_z^f}{\|z\|}z+\frac{t_z^f}{\|z\|}u^{i}\right)} \prod_{i=1}^{\mu} p_{\mathcal{N}_n}(u^i)\mathrm{d}u - \frac{(\lambda-\mu)!}{\lambda!} \int \varphi(u) p_z^{l_z^f}(u)\mathrm{d}u = 0.$$

Let us define the function g on the compact set $\left(\mathcal{L}_{f,z_0^f}+\overline{\mathbb{B}(0,\delta_f)}\right)\times[0,\delta_f]$ as follows. For $(x,\rho)\in\left(\mathcal{L}_{f,z_0^f}+\overline{\mathbb{B}(0,\delta_f)}\right)\times(0,\delta_f]$, $g(x,\rho)$ equals

$$\int_{\|u\|\leq\frac{1}{\sqrt{\rho}}}\left(\int_{\|w\|\leq\frac{1}{\sqrt{\rho}}}1_{(w-u^\mu)^\top\nabla f\left(x+t_x^f\rho(u^\mu+\tau_x^\rho(u^\mu,w)(w-u^\mu))\right)>0}p_{\mathcal{N}_n}(w)\mathrm{d}w\right)^{\lambda-\mu}$$
$$\varphi(u)\prod_{i=1}^{\mu-1}\mathbb{1}_{(u^{i+1}-u^i)^\top\nabla f\left(x+t_x^f\rho(u^i+\tau_x^\rho(u^i,u^{i+1})(u^{i+1}-u^i))\right)>0}\prod_{i=1}^{\mu}p_{\mathcal{N}_n}(u^i)\mathrm{d}u$$

with $\tau_x^\rho(v^1,v^2)\in(0,1)$ defined thanks to the mean value theorem by $f(x+t_x^f\rho v^2)-f(x+t_x^f\rho v^1)=t_x^f\rho(v^2-v^1)^\top\nabla f\left(x+t_x^f\rho(v^1+\tau_x^\rho(v^1,v^2)(v^2-v^1))\right)$. And for $x\in\mathcal{L}_{f,z_0^f}+\overline{\mathbb{B}(0,\delta_f)}$, $g(x,0)$ equals

$$\int\varphi(u)P\Big((\mathcal{N}_n-u^\mu)^\top\nabla f(x)>0\Big)^{\lambda-\mu}\prod_{i=1}^{\mu-1}\mathbb{1}_{(u^{i+1}-u^i)^\top\nabla f(x)>0}\prod_{i=1}^{\mu}p_{\mathcal{N}_n}(u^i)\mathrm{d}u.$$

Remark that $g\left(t_z^f\frac{z}{\|z\|},0\right)=\frac{(\lambda-\mu)!}{\lambda!}\int\varphi(u)p_z^{l_z^f}(u)\mathrm{d}u$. Then using Lemma 16,

$$\lim_{\|z\|\to\infty}g\left(t_z^f\frac{z}{\|z\|},\frac{1}{\|z\|}\right)-\frac{(\lambda-\mu)!}{\lambda!}\int\varphi(u)p_z^f(u)\mathrm{d}u=0.$$

Therefore it is enough to prove that g is uniformly continuous in order to obtain that

$$\frac{(\lambda-\mu)!}{\lambda!}\left(\lim_{\|z\|\to\infty}\int\varphi(u)p_z^f(u)\mathrm{d}u-\int\varphi(u)p_z^{l_z^f}(u)\mathrm{d}u\right)$$
$$=\lim_{\|z\|\to\infty}g\left(t_z^f\frac{z}{\|z\|},\frac{1}{\|z\|}\right)-g\left(t_z^f\frac{z}{\|z\|},0\right)=0.$$

For $(x,\rho)\in\left(\mathcal{L}_{f,z_0^f}+\overline{\mathbb{B}(0,\delta_f)}\right)\times(0,\delta_f]$, for $u\in\overline{\mathbb{B}(0,1/\sqrt{\rho})}$, $w\in\overline{\mathbb{B}(0,1/\sqrt{\rho})}$, $\nabla f\Big(x+t_x^f\rho(u^\mu+\tau_x^\rho(u^\mu,w)(w-u^\mu))\Big)\neq0$ since $x+t_x^f\rho(u^\mu+\tau_x^\rho(u^\mu,w)(w-u^\mu))\in\mathcal{L}_{f,z_0^f}+\overline{\mathbb{B}(0,2\delta_f)}$. Then the set

$$\left\{w\in\mathbb{R}^n;(w-u^\mu)^\top\nabla f\left(x+t_x^f\rho(u^\mu+\tau_x^\rho(u^\mu,w)(w-u^\mu))\right)=0\right\}$$

is Lebesgue negligible. In addition, the function $y\mapsto\mathbb{1}_{y>0}$ is continuous on $\mathbb{R}\setminus\{0\}$, therefore it follows that for almost all w, the function

$$(x,\rho,u^\mu)\mapsto\mathbb{1}_{\|w\|\leq\frac{1}{\sqrt{\rho}}}\mathbb{1}_{(w-u^\mu)^\top\nabla f\left(x+t_x^f\rho(u^\mu+\tau_x^\rho(u^\mu,w)(w-u^\mu))\right)}p_{\mathcal{N}_n}(w)$$

is continuous and bounded by the integrable function $p_{\mathcal{N}_n}$. Then by the dominated convergence theorem, for almost all u, the function $(x,\rho)\mapsto$

$$\mathbb{1}_{\|u\|\leq\frac{1}{\sqrt{\rho}}}\left(\int_{\|w\|\leq\frac{1}{\sqrt{\rho}}}\mathbb{1}_{(w-u^\mu)^\top\nabla f\left(x+t_x^f\rho(u^\mu+\tau_x^\rho(u^\mu,w)(w-u^\mu))\right)>0}p_{\mathcal{N}_n}(w)\mathrm{d}w\right)^{\lambda-\mu}$$

is continuous. The same tools allows to say that for almost all u, the function $(x, \rho) \mapsto \mathbb{1}_{\|u\| \leq \frac{1}{\sqrt{\rho}}} \prod_{i=1}^{\mu-1} \mathbb{1}_{(u^{i+1}-u^i)^\top \nabla f\left(x + t_x^f \rho(u^i + \tau_x^\rho(u^i, u^{i+1})(u^{i+1}-u^i))\right) > 0}$ is continuous. Therefore we can conclude that g is continuous on $\left(\mathcal{L}_{f,z_0^f} + \overline{\mathbb{B}(0,\delta_f)}\right) \times (0, \delta_f]$, and for all $x \in \mathcal{L}_{f,z_0^f} + \overline{\mathbb{B}(0,\delta_f)}$, $\lim_{\rho \to 0} g(x,\rho)$ exists, and is equal to:

$$\int \lim_{\rho \to 0} \mathbb{1}_{\|u\| \leq \frac{1}{\sqrt{\rho}}} \varphi(u) \prod_{i=1}^{\mu-1} \mathbb{1}_{(u^{i+1}-u^i)^\top \nabla f\left(x + t_x^f \rho(u^i + \tau_x^\rho(u^i, u^{i+1})(u^{i+1}-u^i))\right) > 0}$$
$$\left(\int_{\|w\| \leq \frac{1}{\sqrt{\rho}}} 1_{(w-u^\mu)^\top \nabla f\left(x + t_x^f \rho(u^\mu + \tau_x^\rho(u^\mu, w)(w-u^\mu))\right) > 0} \, p_{\mathcal{N}_n}(w) \mathrm{d}w \right)^{\lambda - \mu}$$
$$\prod_{i=1}^{\mu} p_{\mathcal{N}_n}(u^i) \mathrm{d}u = g(x, 0).$$

Finally g is continuous on the compact $\left(\mathcal{L}_{f,z_0^f} + \overline{\mathbb{B}(0,\delta_f)}\right) \times [0, \delta_f]$; it is thereby uniformly continuous on that compact. □

1.5 Linear convergence and divergence illustrations for CSA1-ES and for xNES

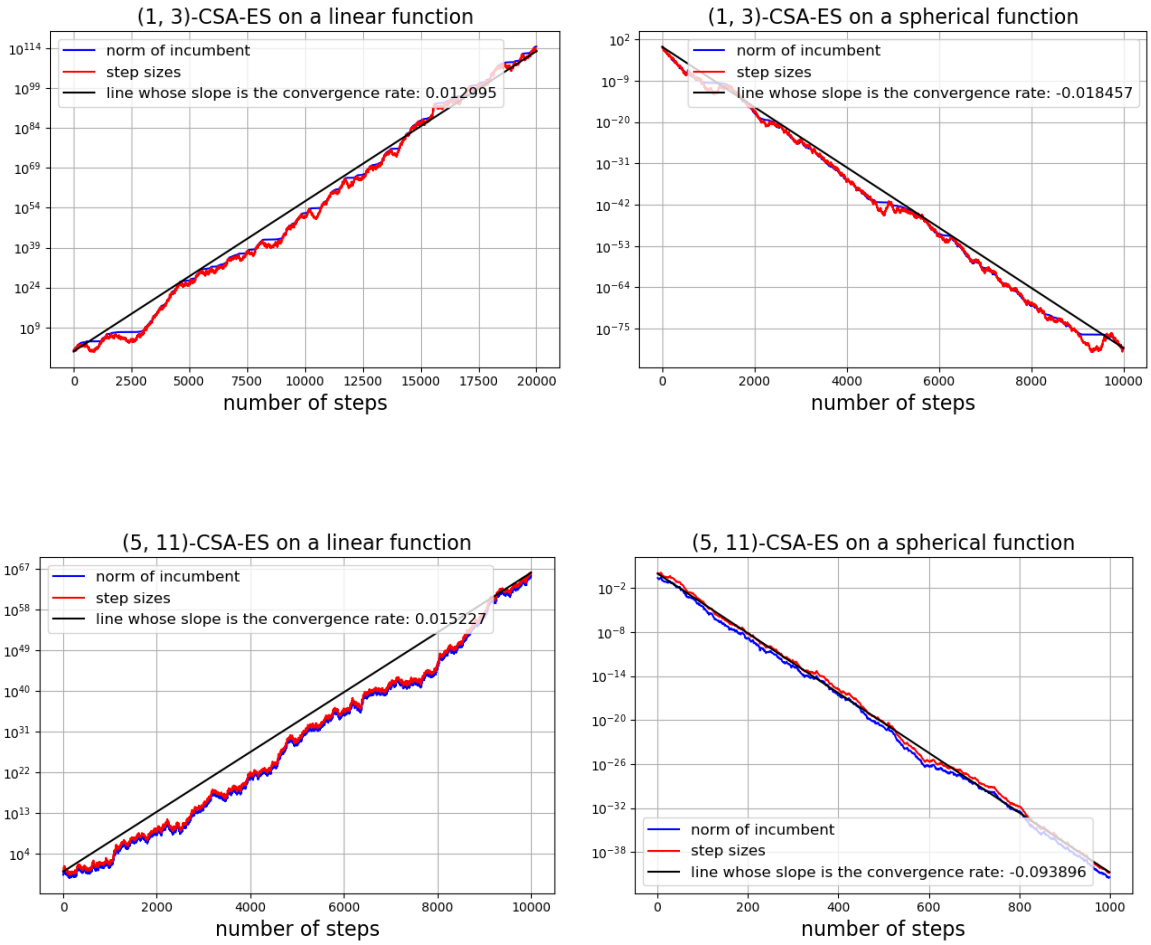


Figure A.1: Illustration of Theorem 5 for CSA1-ES algorithms. Left: f is the linear function $x \to x_1$. Right: f is the spherical function: $x \to \|x\|^2$. Top: $\mu = 1$, $\lambda = 3$ and the weight is 1. Bottom: $\mu = 5$, $\lambda = 11$ and the weights are set as $\text{weights} = \text{cma.recombination\_weights.RecombinationWeights}(5)$.

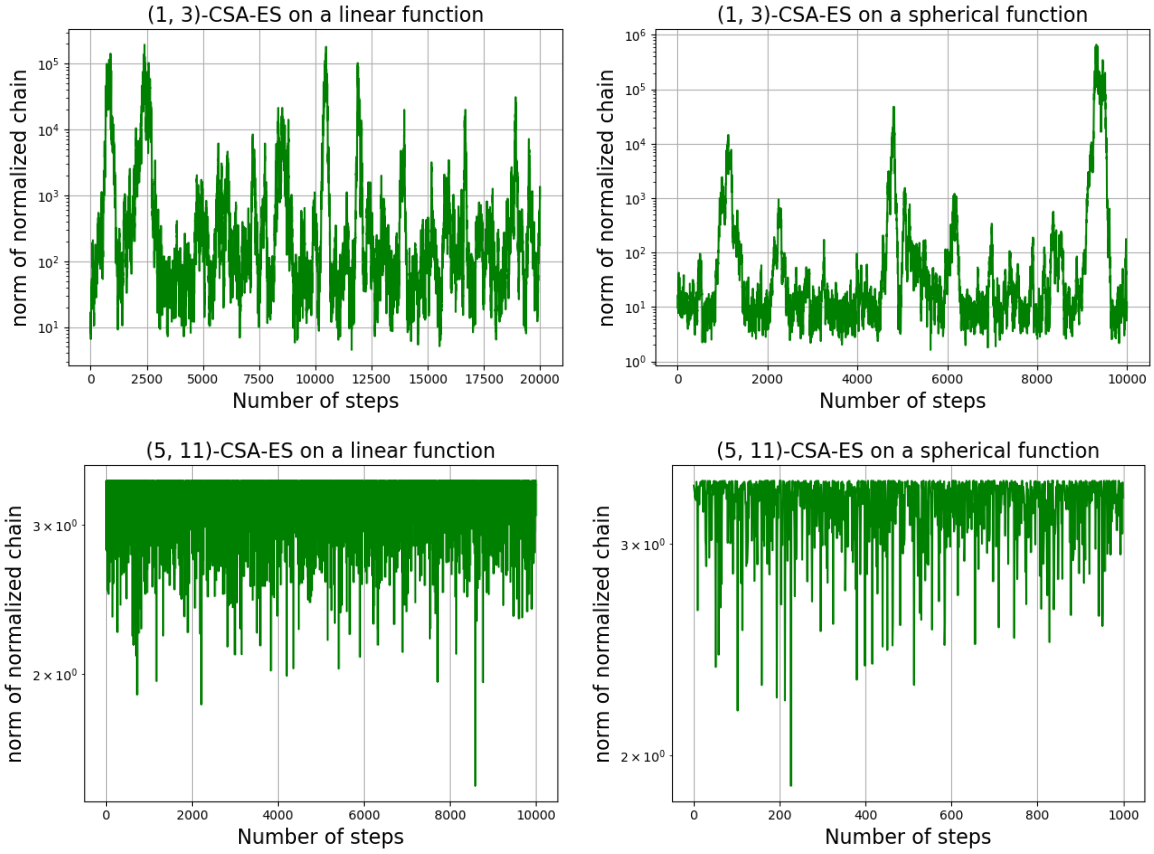


Figure A.2: Illustration of $\{\|Z_k\|; k \in \mathbb{N}\}$ from the main results in Section 4.3 on CSA1-ES. Left: f is the linear function $x \to x_1$. Right: f is the spherical function: $x \to \|x\|^2$. Top: $\mu = 1$, $\lambda = 3$ and the weight is 1. Bottom: $\mu = 5$, $\lambda = 11$ and the weights are set as $\mathrm{weights} = \mathrm{cma.recombination\_weights.RecombinationWeights}(5)$.

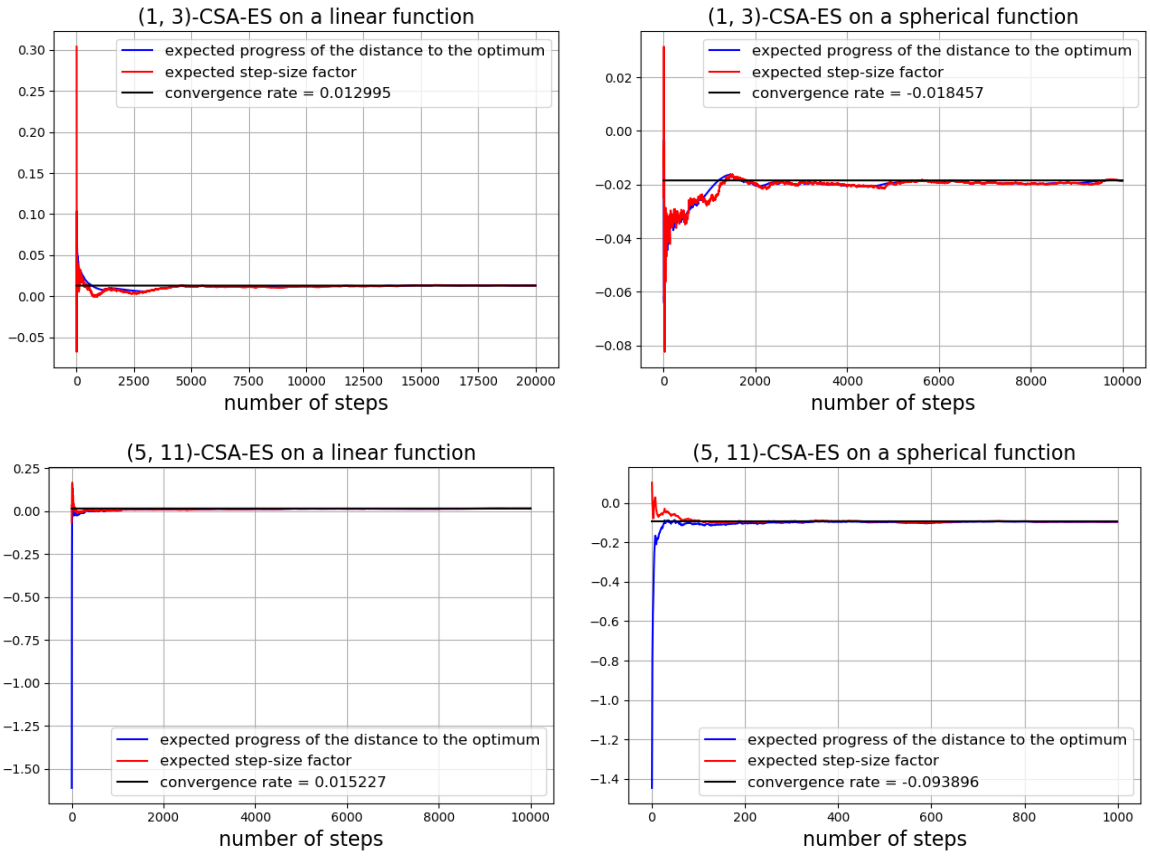


Figure A.3: Illustration of Theorem 5 for CSA1-ES algorithms. Left: f is the linear function $x \to x_1$. Right: f is the spherical function: $x \to \|x\|^2$. Top: $\mu = 1$, $\lambda = 3$ and the weight is 1. Bottom: $\mu = 5$, $\lambda = 11$ and the weights are set as $\text{weights} = \text{cma.recombination\_weights.RecombinationWeights}(5)$.

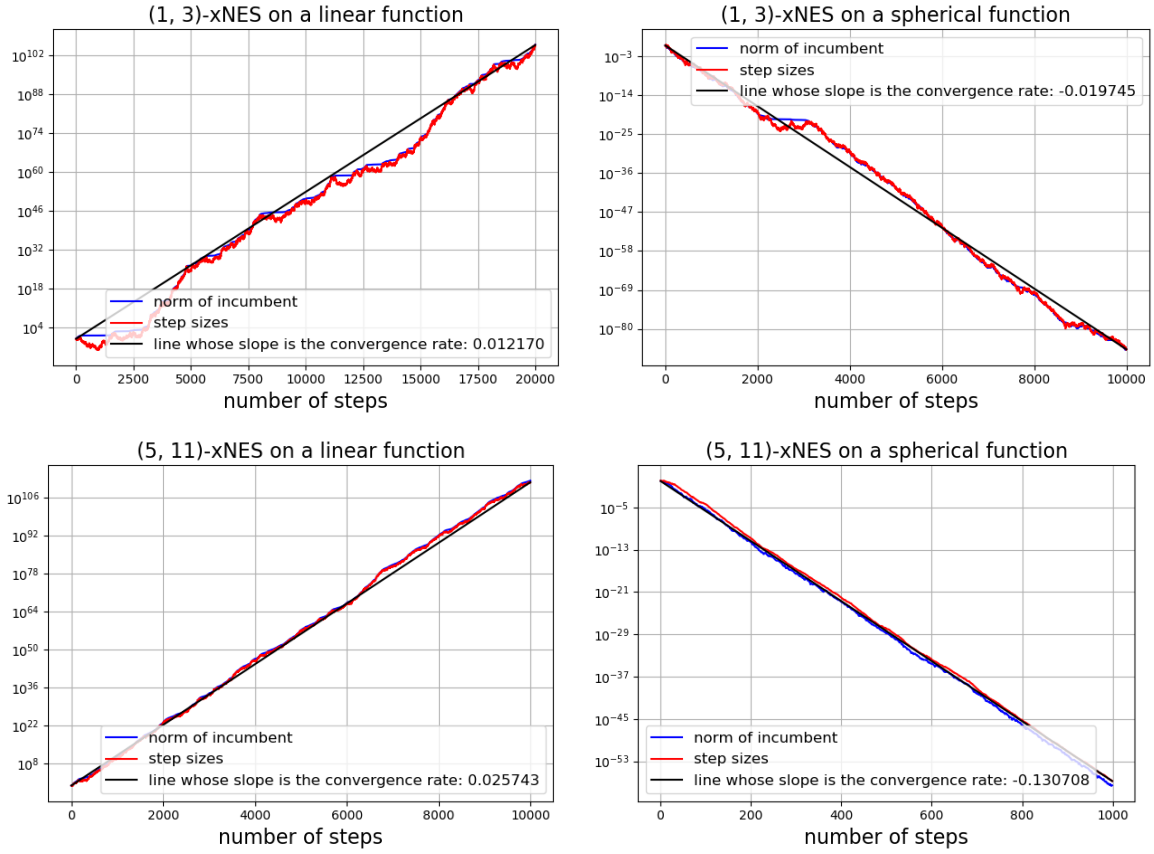


Figure A.4: Illustration of Theorem 5 for xNES algorithms. Left: f is the linear function $x \to x_1$. Right: f is the spherical function: $x \to \|x\|^2$. Top: $\mu = 1$, $\lambda = 3$ and the weight is 1. Bottom: $\mu = 5$, $\lambda = 11$ and the weights are all equal to $\frac{1}{5}$.

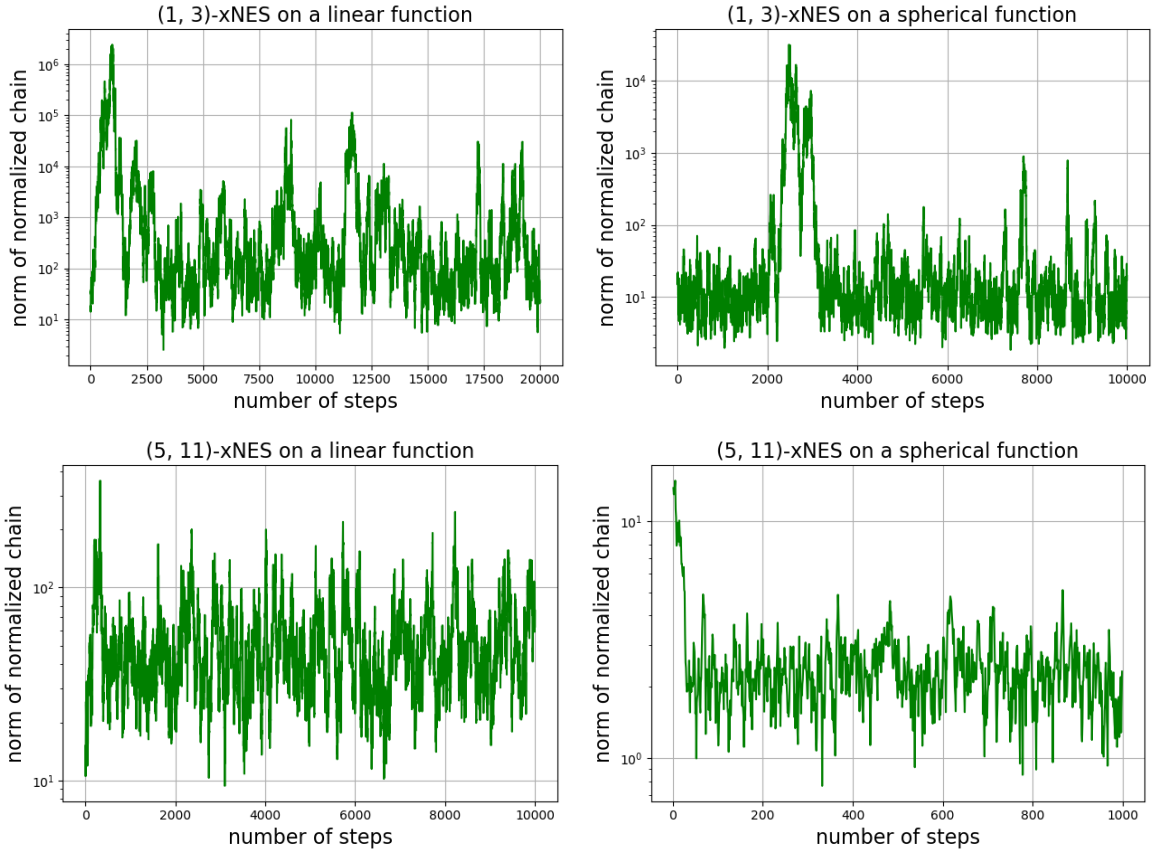


Figure A.5: Illustration of $\{\|Z_k\|; k \in \mathbb{N}\}$ from the main results in Section 4.3 on xNES. Left: f is the linear function $x \to x_1$. Right: f is the spherical function: $x \to \|x\|^2$. Top: $\mu = 1$, $\lambda = 3$ and the weight is 1. Bottom: $\mu = 5$, $\lambda = 11$ and the weights are all equal to $\frac{1}{5}$.

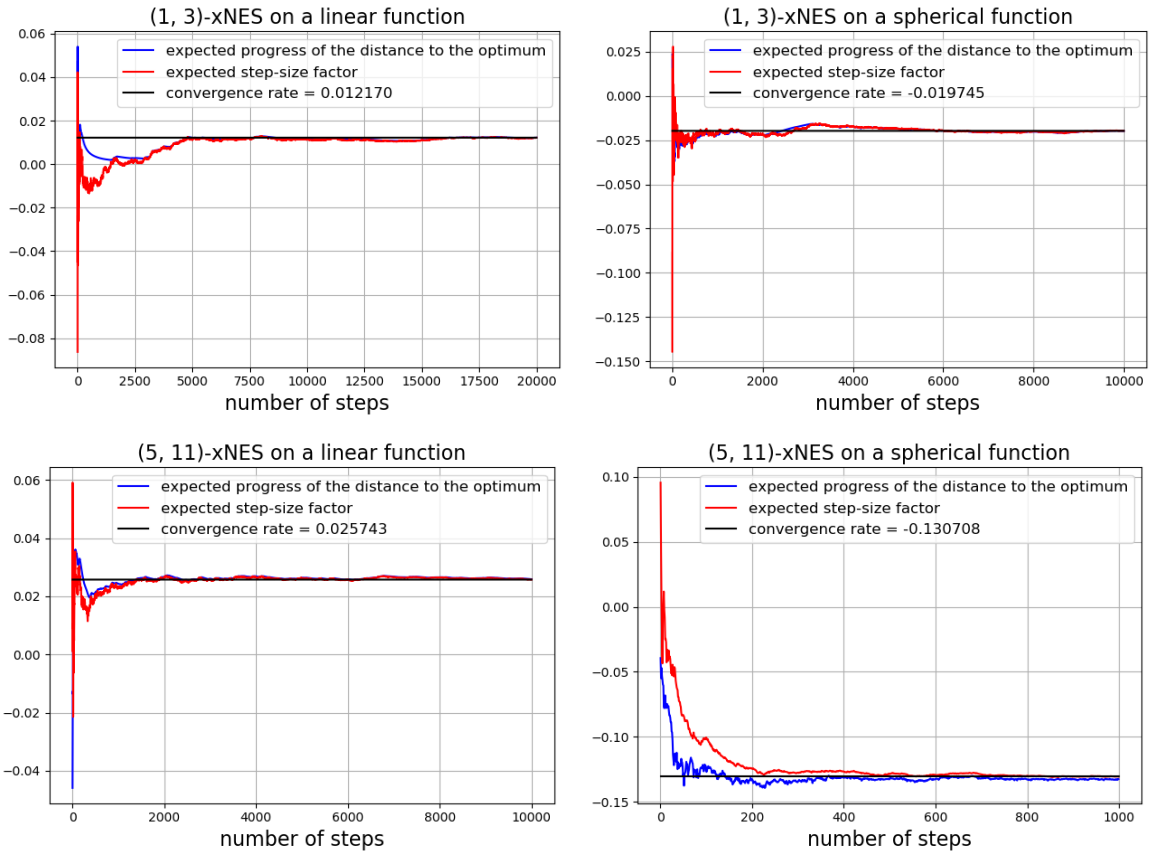


Figure A.6: Illustration of Theorem 5 for xNES algorithms. Left: f is the linear function $x \to x_1$. Right: f is the spherical function: $x \to \|x\|^2$. Top: $\mu = 1$, $\lambda = 3$ and the weight is 1. Bottom: $\mu = 5$, $\lambda = 11$ and the weights are set as $\mathrm{weights} = \mathrm{cma.recombination\_weights.RecombinationWeights}(5)$.

Appendix B

The Matlab code of COMO-CMA-ES

```
function [paretoFront, ...   % objectives
    paretoSet, ...           % parameters
    out, opts] = COMOCMAES(...                % struct with
       information
    problem, ...             % problem-string
    nObj, ...                % number of objectives
    xstart, ...              % initial sample point(s) (if
       only one point is given, the initial population will
       contain copies; len(xstart,1)=nVar (number of
       variables))
    insigma, ...             % initial step size(s)
    inopts)             % struct with options (optional)

% OPTS = COMOCMAES returns default options.
% OPTS = COMOCMAES('defaults') returns default options
   quietly.
% OPTS = COMOCMAES('displayoptions') displays options.
% OPTS = COMOCMAES('defaults', OPTS) supplements options OPTS
   with default
% options.
%
% function call:
% [PARETOFRONT, PARETOSET[, OUT]] = COMOCMAES(PROBLEM, NOBJ,
   XSTART[, OPTS])
%
```

```
19 % Input arguments:
20 %  PROBLEM is a string function name like 'DTLZ1'. Each
   problem
21 %     takes as argument a column vector of variables together
   with the
22 %     number of objectives and a penalty factor and returns
   [objectives,
23 %     variables] (both column vectors). The feedback of
   variables can be
24 %     used to repair illegal values. The parameter
   'penaltyfactor' can be
25 %     used to handle constraint violations.
26 %     Calling a problem with a matrix as first parameter
   interprets the
27 %     columns as individual solutions and computes the
   objective vectors
28 %     and the repaired variables for all solutions in
   parallel.
29 %  NOBJ gives the number of objectives of the multiobjective
   problem
30 %  XSTART indicates the initial sample points that will be
   used to
31 %     initialize the individual means of MO-CMA-ES's sample
   distributions.
32 %     The number of rows thereby gives the number of
   variables and the
33 %     number of columns the population size. If the number of
   columns does
34 %     not match the population size given in opts.nPop, the
   first column of
35 %     XSTART is used to initialize the mean vectors of all
   sample
36 %     distributions. Note that also a string can be given
   that will be
37 %     evaluated as MATLAB code such as 'rand(10, nPop)'.
38 %  OPTS (an optional argument) is a struct holding additional
   input
39 %     options. Valid field names and a short documentation
   can be
40 %     discovered by looking at the default options (type
   'mocmaes'
41 %     without arguments, see above). Empty or missing fields
```

```
   in OPTS
%     invoke the default value, i.e. OPTS needs not to have
   all valid
%     field names.  Capitalization does not matter and
   unambiguous
%     abbreviations can be used for the field names. If a
   string is
%     given where a numerical value is needed, the string is
   evaluated
%     by eval, where
%     'nVar' expands to the problem dimension
%     'nObj' expands to the objectives dimension
%     'nPop' expands to the population size
%     'countEval' expands to the number of the recent
   evaluation
%     'nPV' expands to the number paretofronts
%
% Output:
%  PARETOFRONT is a matrix holding the objectives in rows.
   Each column
%     holds the objective vector of one solution.
%  PARETOSET is a matrix holding the parameters in rows. Each
   column holds
%     one solution.
%  OUT is a struct with additional information about the run.
%



% ---------- Set Defaults for Options
   -------------------------------
% options: general - these are evaluated once
defopts.nPop              = '10          % size of the
   population, which means here the number of kernels';
defopts.nOffspring        = '(4+floor( 3*log(nVar) )) * 10
            % total number of offspring';
defopts.popsize           = '4+floor( 3*log(nVar) )
  % number of offspring per kernel';
defopts.maxEval           = 'inf     % maximum number of
   evaluations';
defopts.refpoint          = '(1 + nVar).*ones(1, nObj) %
   reference point of the hypervolume';
```

```
70 defopts.bounds            = '[-inf, inf] % bounds manage the
       boundary constraints';
71 % bounds can be of size (1,2), or (2,1) or of type [lb; rb]
       where lb and rb
72 % respectfully represent the lower and upper bounds of size
       (1, nVar), or
73 % of type [lb, rb] where lb and rb respectfully represent the
       lower and
74 % upper bounds of size (nVar, 1).
75 defopts.okresume          = 'False % resume former run';
76 defopts.isarchive          = 'False % whether or not we keep
       track of the non-dominated points';
77 defopts.resumefile        = '';
78 defopts.maxiter           = 'inf % maximum number of
       iteration during a run';
79 defopts.number_asks       = 'py.str("all") % the number of
       kernels from which we generate offspring simultaneously';
80 % In the algorithm, when we do 'moes.ask(number_asks)', then
       'number_asks' kernels generate
81 % offspring, that they will pass to the 'tell' method, after
       all offspring
82 % are evaluated (in parallel). If opts.number_asks = 'all',
       then all
83 % kernels are asked during a call of the 'ask' method.
84 defopts.tolx              = '1e-6 % tolerence in x for
       stopping criterion';
85 defopts.frac_inactive         = '1 % fraction of inactive
       kernel for stopping';
86 defopts.logger            = '1 % if 1, we log data to file.
       And if 0, we do not';
87 defopts.elitist           = '0 % 3 possibilities: 0 , 1 , 2';
88 % if 0: non-elitist mode, if 1: elitist mode, and 2  for
       'init': only
89 % the start is elitist.
90 defopts.verb_disp         = '100 % display results each
       verb_disp iterations';
91 defopts.display           = 'on            % display some
       things during the run and graphics';
92 defopts.showWaitbar       = 'on  % FHU display waitbar if
       closed during process stop optimization';
93 defopts.abscissa          = '0 % if 1, the x_axis represents
       count evals, and if 0, it shows the iterations ';
```

```
94
95 % ---------------------- Handling Input Parameters
   ----------------------
96
97 if nargin < 1 || isequal(problem, 'defaults') % pass default
   options
98     if nargin < 5
99         disp('Default options returned (type "help mocmaes"
              for help).');
100    end
101    paretoFront = defopts;
102    if nargin > 5 % supplement second argument with default
          options
103        paretoFront = getoptions(inopts, defopts);
104    end
105    return;
106 end
107
108 if isequal(problem, 'displayoptions')
109    names = fieldnames(defopts);
110    for name = names'
111        disp([name{:} repmat(' ', 1, 20-length(name{:})) ':
              ''' defopts.(name{:}) '''']);
112    end
113    return;
114 end
115
116 if isempty(problem)
117    error('Objective function not determined');
118 end
119 % if ~(ischar(problem)
120 %     error('first argument ''problem'' must be a string or
   an handle function');
121 % end
122
123 % Compose options opts
124 if nargin < 5 || isempty(inopts) % no input options available
125    opts = defopts;
126 else
127    opts = getoptions(inopts, defopts);
128 end
129
```

```
130 % ----------- Importing python modules that we need
    -------------------
131
132 py.importlib.import_module('comocma');
133 py.importlib.import_module('cma');
134
135 % ----------------------- Initialization
    ------------------------------
136 %clc;
137
138 % get parameters for initialization
139
140 % initial means
141 if nargin < 3
142     xstart = [];
143 end
144 if isempty(xstart)
145     error('Initial search points and problem dimension not
        determined');
146 end
147 xstart = myeval(xstart);
148 maxiter = myeval(opts.maxiter);
149 maxEval = myeval(opts.maxEval);
150 resume = myeval(opts.okresume);
151 isarchive = myeval(opts.isarchive);
152 resumefile =opts.resumefile;
153 %resumefile = py.str(resumefile); % in case it's a matlab char
154 logger = myeval(opts.logger);
155 display = myeval(opts.display);
156 abscissa = myeval(opts.abscissa);
157 nVar = size(xstart,2);
158 nPop = size(xstart,1);
159 tolx = myeval(opts.tolx);
160 frac_inactive=myeval(opts.frac_inactive);
161 verb_disp = myeval(opts.verb_disp);
162 displayWaitbar     = myeval(opts.showWaitbar); % FHU display
    waitbar
163 elitist = myeval(opts.elitist);
164 if elitist == 0
165     elitist = py.False;
166 elseif elitist == 2
167     elitist = py.str('init');
```

```
168 else
169     elitist = py.True;
170 end
171 nPop_opts = myeval(opts.nPop); % in order to test for
    consistency
172 if nPop ~= nPop_opts
173     xstart = repmat(xstart(1,:), nPop_opts, 1);
174     nPop = nPop_opts;
175 end
176 number_asks = myeval(opts.number_asks);
177 if isa(number_asks, 'double') || isa(number_asks, 'int64') ||
    isa(number_asks, 'int32')
178     number_asks = py.int(number_asks);
179 end
180
181 % initial step sizes
182 if nargin < 4
183     insigma = [];
184 end
185
186
187 if any(insigma) <= 0
188     error('Initial step sizes cannot be <= 0.');
189 end
190 if size(insigma, 1) ~= nPop
191     insigma = repmat(insigma(1, :), nPop, 1);
192 end
193
194
195 refpoint = myeval(opts.refpoint);
196 if size(refpoint, 1) ~= 1
197     refpoint = refpoint';
198 end
199
200 reference_point = py.numpy.array(refpoint);
201 x_starts = py.list({});
202 sigma0 = py.list({});
203
204 for i = 1:nPop
205     x_starts.append(py.numpy.array(xstart(i,:)));
206     sigma0.append(py.float(insigma(i)));
207 end
```

```
208
209 bounds = myeval(opts.bounds);
210 if size(bounds, 1) ~= 2
211     bounds = bounds';
212 end
213 lb = bounds(1, :);
214 rb = bounds(2, :);
215
216 newbounds = py.list({lb, rb});
217 popsize = py.int(myeval(opts.popsize));
218 %TODO: authorize the possibility for different options to
      different cma-es
219 cmaes_opts = py.dict(struct('popsize', popsize, 'bounds',
      newbounds, 'tolx', tolx, 'CMA_elitist', elitist));
220 list_of_solvers = py.comocma.get_cmas(x_starts, sigma0,
      pyargs('inopts', cmaes_opts));
221 sofomore_opts = py.dict(struct('archive', isarchive));
222
223 if display && logger
224
225     figure(44444);
226     h(1)=subplot(2,2,1);%#TODO: FHU subplot initialization to
        avoid trouble with waitbar
227     h(2)=subplot(2,2,2);
228     h(3)=subplot(2,2,3);
229     h(4)=subplot(2,2,4);
230     %         h(5)=subplot(2,3,5);
231     %         h(6)=subplot(2,3,6);
232     hold off;
233
234
235 end
236
237 % ----------------------- end Initialization
      ------------------------------
238
239 if ~resume
240
241     moes = py.comocma.Sofomore(list_of_solvers,
        reference_point, sofomore_opts);
242
243 else
```

```
244     try
245         moes = py.pickle.load(py.open(py.str(resumefile) +
                py.str('.pkl'), 'rb'));
246         %TODO: a way to modify the options of moes here.
247     catch % when the file doesn't exist
248         %(for example 0 iteration has been made).
249         moes = py.comocma.Sofomore(list_of_solvers,
                reference_point, sofomore_opts);
250     end
251
252
253 end
254
255 stopflag = {};
256 stopByUser=false;
257 if displayWaitbar
258     try
259         hw=
                waitbar(int64(moes.countiter)/maxiter,WaitbarString,
                'Name', 'COMOCMAES');
260     catch
261
262         hw= waitbar(int64(moes.countiter)/maxiter,'from
                resume','Name','COMOCMAES');
263     end
264 end
265 %% boucle de calage
266 inactive=0;
267 while moes.stop() == 0 && moes.countiter < maxiter &&
    moes.countevals < maxEval && ~stopByUser &&
    inactive<frac_inactive
268
269     X = moes.ask(number_asks); %
270     X_matlab = zeros(int64(py.len(X)), nVar);
271     for i=1:size(X_matlab,1)
272         X_matlab(i,:) = double(py.array.array('d',X{i}));
273     end
274     f_values_and_constraints = feval(problem, X_matlab);
275     F_matlab = f_values_and_constraints(:, 1:nObj);
276     C_matlab = f_values_and_constraints(:,nObj+1:end);
277     C = py.list({});
278     if size(C_matlab, 2) ~= 0 % we have constraints
```

```
279         for i=1:size(C_matlab,2)
280             C.append(py.list(C_matlab(:,i)'));
281             %         C.append(py.list(C_matlab(i,:)))
282         end
283     end
284     F = py.list({});
285     for i=1:size(X_matlab,1)
286         F.append(py.list(F_matlab(i,:)));
287         %         C.append(py.list(C_matlab(i,:)))
288     end
289
290 %    moes.tell(X, F, py.list(C));
291     moes.tell(X, F);
292
293     moes.disp(verb_disp);
294     drawnow; % to display immediately what is in disp()
295
296
297     if logger
298         moes.logger.add()
299     end
300
301     if strcmp(resumefile, '') == 0
302         py.pickle.dump(moes, py.open(py.str(resumefile) +
                py.str('.pkl'), 'wb'))
303         % TODO: expose the name of the saved file in the
                options
304
305     end
306
307
308     if display && logger && mod(int64(moes.countiter),
            verb_disp) == 0
309         inactive=myplot(moes, abscissa, nObj, nVar, opts,
                hw);%, refpoint);
310
311     elseif logger && mod(int64(moes.countiter), verb_disp) ==
            0
312         filenames_ratio = py.list({moes.logger.name_prefix +
                py.str('ratio_inactive_kernels.dat'),...
313             moes.logger.name_prefix +
                    py.str('ratio_nondom_incumb.dat'),
```

```
                moes.logger.name_prefix +
                py.str('ratio_nondom_offsp_incumb.dat')});
314         tuple_ratio = moes.logger.load(filenames_ratio);
315         res_ratio = tuple_ratio{3};
316         inactive = max(double(py.array.array('d',
               res_ratio{1})));
317
318     end
319     displayWaitbar = myeval(opts.showWaitbar);
320     if displayWaitbar && moes.countiter > 2
321         max_max_stds = double(moes.max_max_stds);
322         WaitbarString=['It ', num2str(int64(moes.countiter)),
               ' HVmax=', num2str(...
323         double(py.float(moes.best_hypervolume_pareto_front)),
               '%.3e'),' max(max stds)=', num2str(max_max_stds,
               '%.3e')];
324
325         maxiter = myeval(opts.maxiter);
326         if ishandle(hw)
327             waitbar(double(int64(moes.countiter))/maxiter,
                   hw, WaitbarString, 'Name', 'COMOCMAES');
328         else % waitbar closed-> stop
329             stopByUser=true;
330             stopflag={stopflag{:},'Stop by user'};
331         end
332     end
333
334
335 end
336
337 % last display (end of while loop):
338 if display
339     myplot(moes, abscissa, nObj, nVar, opts, hw);
340 end
341
342
343 paretoFront = zeros(int64(py.len(moes.pareto_front_cut)),
       nObj);
344 for i = 1:size(paretoFront,1)
345     infront = moes.pareto_front_cut{i};
346     paretoFront(i,:) = double(py.array.array('d', infront));
347 end
```

```
348
349 paretoSet = zeros(int64(py.len(moes.pareto_set_cut)), nVar);
350 for i = 1:size(paretoSet,1)
351     infront = moes.pareto_set_cut{i};
352     paretoSet(i,:) = double(py.array.array('d', infront));
353 end
354 out = struct();
355
356 if isarchive
357     archive = zeros(int64(py.len(moes.archive)), nObj);
358     for i = 1:size(archive,1)
359         infront = moes.archive{i};
360         archive(i,:) = double(py.array.array('d', infront));
361     end
362     out.archive = archive;
363 end
364 out.termination_status = moes.termination_status;
365 out.num_kernels = int64(py.len(moes));
366 out.stop = moes.stop();
367 out.countiter = int64(moes.countiter);
368 out.countevals = double(moes.countevals);
369 out.best_hypervolume_pareto_front =
    double(py.float(moes.best_hypervolume_pareto_front));
370 out.hypervolume_pareto_front =
    double(py.float(moes.pareto_front_cut.hypervolume));
371 if isarchive
372     out.hypervolume_archive =
        double(py.float(moes.archive.hypervolume));
373 end
374 out.moes = moes;
375 if ishandle(hw)
376     close(hw);
377 end
378 end
379 %%-----------------------------------------------
380 function opts=getoptions(inopts, defopts)
381 % OPTS = GETOPTIONS(INOPTS, DEFOPTS) handles an arbitrary
    number of
382 % optional arguments to a function. The given arguments are
    collected
383 % in the struct INOPTS.  GETOPTIONS matches INOPTS with a
    default
```

```
384 % options struct DEFOPTS and returns the merge OPTS.  Empty
    or missing
385 % fields in INOPTS invoke the default value.  Fieldnames in
    INOPTS can
386 % be abbreviated.
387 if nargin < 2 || isempty(defopts) % no default options
    available
388     opts=inopts;
389     return;
390 elseif isempty(inopts) % empty inopts invoke default options
391     opts = defopts;
392     return;
393 elseif ~isstruct(defopts) % handle a single option value
394     if isempty(inopts)
395         opts = defopts;
396     elseif ~isstruct(inopts)
397         opts = inopts;
398     else
399         error('Input options are a struct, while default
            options are not');
400     end
401     return;
402 elseif ~isstruct(inopts) % no valid input options
403     error('The options need to be a struct or empty');
404 end
405
406 opts = defopts; % start from defopts
407 % if necessary overwrite opts fields by inopts values
408 defnames = fieldnames(defopts);
409 idxmatched = []; % indices of defopts that already matched
410 for name = fieldnames(inopts)'
411     name = name{1}; % name of i-th inopts-field
412     idx = strncmpi(defnames, name, length(name));
413     if sum(idx) > 1
414         error(['option "' name '" is not an unambigous
            abbreviation. ' ...
415             'Use opts=RMFIELD(opts, ''' name, ...
416             ''') to remove the field from the struct.']);
417     end
418     if sum(idx) == 1
419         defname  = defnames{find(idx)};
420         if ismember(find(idx), idxmatched)
```

```
421             error(['input options match more than ones with
                    "' ...
422                 defname '". ' ...
423                 'Use opts=RMFIELD(opts, ''' name, ...
424                 ''') to remove the field from the struct.']);
425         end
426         idxmatched = [idxmatched find(idx)];
427         val = getfield(inopts, name);
428         % next line can replace previous line from MATLAB
                version 6.5.0 on and in octave
429         % val = inopts.(name);
430         if isstruct(val) % valid syntax only from version
                6.5.0
431             opts = setfield(opts, defname, ...
432                 getoptions(val, getfield(defopts, defname)));
433         elseif isstruct(getfield(defopts, defname))
434             % next three lines can replace previous three
                    lines from MATLAB
435             % version 6.5.0 on
436             %   opts.(defname) = ...
437             %       getoptions(val, defopts.(defname));
438             % elseif isstruct(defopts.(defname))
439             warning(['option "' name '" disregarded (must be
                    struct)']);
440         elseif ~isempty(val) % empty value: do nothing, i.e.
                stick to default
441             opts = setfield(opts, defnames{find(idx)}, val);
442             % next line can replace previous line from MATLAB
                    version 6.5.0 on
443             % opts.(defname) = inopts.(name);
444         end
445     else
446         warning(['option "' name '" disregarded (unknown
                field name)']);
447     end
448 end
449 end
450 %%-----------------------------------------
451 %%-----------------------------------------
452 function res=myeval(s)
453 if ischar(s)
454     if strncmpi(s, 'yes', 3) || strncmpi(s, 'on', 2) ...
```

```
455             || strncmpi(s, 'true', 4) || strncmp(s, '1 ', 2)
456         res = 1;
457     elseif strncmpi(s, 'no', 2) || strncmpi(s, 'off', 3) ...
458             || strncmpi(s, 'false', 5) || strncmp(s, '0 ', 2)
459         res = 0;
460     else
461         try res = evalin('caller', s); catch
462             error(['String value "' s '" cannot be
                    evaluated']);
463         end
464         try res ~= 0; catch
465             error(['String value "' s '" cannot be evaluated
                    reasonably']);
466         end
467     end
468 else
469     res = s;
470 end
471 end
472
473
474 function max_inactive=myplot(moes, abscissa, nObj, nVar,
    opts, hw, refpoint)
475 filenames_ratio = py.list({moes.logger.name_prefix +
    py.str('ratio_inactive_kernels.dat') ,...
476     moes.logger.name_prefix +
          py.str('ratio_nondom_incumb.dat'),
          moes.logger.name_prefix +
          py.str('ratio_nondom_offsp_incumb.dat')});
477 tuple_ratio = moes.logger.load(filenames_ratio);
478 iter_ratio = double(py.array.array('d', tuple_ratio{1}));
479 countevals_ratio = double(py.array.array('d',
    tuple_ratio{2}));
480 res_ratio = tuple_ratio{3};
481
482 filenames_hypervolume = py.list({moes.logger.name_prefix +
    py.str('hypervolume.dat') ,...
483     moes.logger.name_prefix +
          py.str('hypervolume_archive.dat'),
          moes.logger.name_prefix + py.str('len_archive.dat')});
484 tuple_hypervolume = moes.logger.load(filenames_hypervolume);
485 iter_hypervolume = double(py.array.array('d',
```

```
    tuple_hypervolume{1}));
countevals_hypervolume = double(py.array.array('d',
    tuple_hypervolume{2}));
res_hypervolume = tuple_hypervolume{3};

filenames_median = py.list({moes.logger.name_prefix +
    py.str('median_sigmas.dat'),...
     moes.logger.name_prefix + py.str('median_min_stds.dat'),
        moes.logger.name_prefix +
        py.str('median_max_stds.dat')});
tuple_median = moes.logger.load(filenames_median);
iter_median = double(py.array.array('d', tuple_median{1}));
countevals_median= double(py.array.array('d',
    tuple_median{2}));
res_median = tuple_median{3};

filenames_median_stds = py.list({moes.logger.name_prefix +
    py.str('median_stds.dat')});
tuple_median_stds = moes.logger.load(filenames_median_stds);
iter_median_stds = double(py.array.array('d',
    tuple_median_stds{1}));
countevals_median_stds= double(py.array.array('d',
    tuple_median_stds{2}));
res_median_stds_python = py.list(tuple_median_stds{3}{1});
res_median_stds = zeros(size(iter_median_stds, 2), nVar);
for i=1:size(res_median_stds, 1)
    res_median_stds(i,:) =
        double(py.array.array('d',res_median_stds_python{i}));
end

if abscissa
    x_axis_hypervolume = countevals_hypervolume;
    x_axis_median = countevals_median;
    x_axis_ratio = countevals_ratio;
    x_axis_median_stds = countevals_median_stds;
else
    x_axis_hypervolume = iter_hypervolume;
    x_axis_median = iter_median;
    x_axis_ratio = iter_ratio;

    x_axis_median_stds = iter_median_stds;
    %         x_axis_median_stds = countevals_median_stds;
```

```
518 end
519
520 figure(44444);
521 h(1)=subplot(2,2,1);%#TODO: FHU subplot initialization to
      avoid trouble with waitbar
522 h(2)=subplot(2,2,2);
523 h(3)=subplot(2,2,3);
524 h(4)=subplot(2,2,4);
525 %fixedpop = population; % testing
526 linestyle = '--';
527 set(0, 'CurrentFigure', 44444);
528
529 %%%%%%%%%%%%%%%%%
530 %             subplot(2,3,1);
531 axes(h(1)); %TODO FHU: replace subplot by axes
532 %%%%%%%%%%%%%%%%%
533 % evolution of HV:
534 cla;
535 hold off;
536 HV = double(py.array.array('d', res_hypervolume{1}));
537 HV_max = double(py.float(moes.best_hypervolume_pareto_front));
538 sigmas = double(py.array.array('d', res_median{1}));
539 min_stds = double(py.array.array('d', res_median{2}));
540 max_stds = double(py.array.array('d', res_median{3}));
541
542
543 semilogy(x_axis_hypervolume, HV_max - HV, 'LineStyle',
      linestyle, ...
544     'Color', 'red');
545 hold on;
546 if moes.isarchive ~= 0
547     HV_archive = double(py.array.array('d',
          res_hypervolume{2}));
548     HV_archive_max = HV_archive(end);
549     length_archive = double(py.array.array('d',
          res_hypervolume{3}));
550     inverse_length_archive = 1./length_archive;
551     semilogy(x_axis_hypervolume, HV_archive_max - HV_archive,
          'LineStyle', linestyle, ...
552         'Color', 'blue');
553     hold on
554     semilogy(x_axis_hypervolume, inverse_length_archive,
```

```
        'LineStyle', linestyle, ...
555          'Color', 'cyan');
556 end
557
558 %         hold on
559 %         semilogy(x_axis_median, sigmas, 'LineStyle',
    linestyle, ...
560 %             'Color', 'k');
561 %         hold on
562 %         semilogy(x_axis_median, min_stds, 'LineStyle',
    linestyle, ...
563 %             'Color', 'green');
564 %         hold on
565 %         semilogy(x_axis_median, max_stds, 'LineStyle',
    linestyle, ...
566 %             'Color', 'green');
567 %                 hold on
568 %             legend('HV_{max} - HV', 'HV_{archive_{max}} -
    HV_{archive}', 'inverse length archive', 'median sigmas',
    'median min stds', 'median max stds');
569 title('HV_{max}-HV (red,blue), inverse-length-archive(cyan)
    ');
570 if abscissa
571     xlabel('Count evals');
572 else
573     xlabel('Niter');
574 end
575 %text(0,100,sprintf('HV = %e', max(HVtotal)));
576 ax = axis;
577 text(ax(1),
    10^(log10(ax(3))+0.05*(log10(ax(4))-log10(ax(3)))), ...
578     [ ' HV=' num2str(double(py.float(...
579     moes.best_hypervolume_pareto_front)), '%.15g')]);
580
581
582 grid on;
583
584 %%%%%%%%%%%%%%%%%%
585 %             subplot(2,3,2);
586 axes(h(2)); %TODO FHU: replace subplot by axes
587 %%%%%%%%%%%%%%%%%%
588 % population in objective space over time:
```

```
589 cla;
590 hold off;
591
592 currentFront = zeros(int64(py.len(moes.pareto_front_cut)),
      nObj);
593 for i = 1:size(currentFront,1)
594     infront = moes.pareto_front_cut{i};
595     currentFront(i,:) = double(py.array.array('d', infront));
596 end
597 % need to plot all kernels in objective space
598 % All kernels in objective space
599 All_kernels = zeros(int64(py.len(moes.kernels)), nObj);
600 for i = 1:size(All_kernels,1)
601     kernel = moes.kernels{i};
602     infront = kernel.objective_values;
603     All_kernels(i,:) = double(py.array.array('d', infront));
604 end
605
606
607 if moes.isarchive ~= 0
608     currentArchive = zeros(int64(py.len(moes.archive)), nObj);
609     for i = 1:size(currentArchive,1)
610         infront = moes.archive{i};
611         currentArchive(i,:) = double(py.array.array('d',
              infront));
612     end
613 end
614
615 if nObj == 3 % 3 objectives
616
617     if moes.isarchive ~= 0
618         plot3(currentArchive(:,1), currentArchive(:,2),
              currentArchive(:,3),'.', 'Color', 'blue');
619         xlabel('f_1');
620         ylabel('f_2');
621         zlabel('f_3');
622         MinAbciss=min(currentArchive);
623         hold on;
624     else
625         MinAbciss=min(currentFront);
626     end
627     plot3(All_kernels(:,1), All_kernels(:,2),
```

```
        All_kernels(:,3),'o', 'Color', 'green');hold on
628     plot3(currentFront(:,1), currentFront(:,2),
        currentFront(:,3),'o', 'Color', 'red');
629     xlabel('f_1');
630     ylabel('f_2');
631     zlabel('f_3');
632     if nargin>6
633         xlim([MinAbciss(1),refpoint(1)]);
634         ylim([MinAbciss(2),refpoint(2)]);
635         zlim([MinAbciss(3),refpoint(3)]);
636     end
637 else  % 2 objectives
638
639
640     if moes.isarchive ~= 0
641         plot(currentArchive(:,1), currentArchive(:,2),'.',
            'Color', 'blue');
642         xlabel('f_1');
643         ylabel('f_2');
644         hold on;
645         MinAbciss=min(currentArchive);
646     else
647         MinAbciss=min(currentFront);
648     end
649     plot(All_kernels(:,1), All_kernels(:,2),'o', 'Color',
        'green');hold on
650     plot(currentFront(:,1), currentFront(:,2),'o', 'Color',
        'red');
651     xlabel('f_1');
652     ylabel('f_2');
653     if nargin>6
654         xlim([MinAbciss(1),refpoint(1)]);
655         ylim([MinAbciss(2),refpoint(2)]);
656     end
657 end
658
659 title('objective space: archive (blue), kernels(red)');
660
661 %         legend('estimated front','archive');
662 grid on;
663
664 %%%%%%%%%%%%%%%%%%
```

```
665 %             subplot(2,3,3);
666 axes(h(3)); %TODO FHU: replace subplot by axes
667 cla;
668 % axis([-0.01, 1.01]);
669 %%%%%%%%%%%%%%%%%%
670 % population in decision space (projection) over time:
671 hold off;
672
673 inactive = double(py.array.array('d', res_ratio{1}));
674 max_inactive=max(inactive);
675 nondom_incumbent = double(py.array.array('d', res_ratio{2}));
676 %         first_quartile_nondom_offspring_incumbent =
      double(py.array.array('d', res_ratio{3}));
677 median_nondom_offspring_incumbent =
      double(py.array.array('d', res_ratio{4}));
678 %         last_quartile_nondom_offspring_incumbent =
      double(py.array.array('d', res_ratio{5}));
679
680 plot(x_axis_ratio, inactive, '-', 'Color', 'red');
681 hold on;
682 plot(x_axis_ratio, nondom_incumbent, '-', 'Color', 'blue');
683 hold on;
684 %             plot(x_axis_ratio,
      first_quartile_nondom_offspring_incumbent, '-', 'Color',
      'green');
685 %         hold on;
686 plot(x_axis_ratio, median_nondom_offspring_incumbent, '-',
      'Color', 'k');
687 %             hold on;
688 %             plot(x_axis_ratio,
      last_quartile_nondom_offspring_incumbent, '-', 'Color',
      'green');
689
690 if abscissa
691     xlabel('Count evals');
692 else
693     xlabel('Niter');
694 end
695 ylabel('Ratios');
696 title('Fracpareto(Kernels:blue,offspring:black) and inactive
      kernels(red)');
697 %         legend('inactive kernels','non-dominated
```

```
   incumbents','1st quartile non-dom incumbent +
   offspring',...
%              'median non-dom incumbent + offspring', 'last
   quartile non-dom incumbent + offspring');
grid on;

%%%%%%%%%%%%%%%%%%
%              subplot(2,3,5);
axes(h(4)); %TODO FHU: replace subplot by axes
%%%%%%%%%%%%%%%%%%
% axis ratios of all covariances:
cla;
hold off;
semilogy(x_axis_median_stds, res_median_stds);
hold on;
tol = myeval(opts.tolx);
tol = tol / double(moes.kernels{1}.sigma0);
semilogy(x_axis_median_stds, tol *
   ones(size(x_axis_median_stds)));

if abscissa
    xlabel('Count evals');
else
    xlabel('Niter');
end
title('(sorted) median standard deviations');
grid on

hold off;
drawnow;



end % of plotting
%%-------------------------------------------------------------------
```

ECOLE
DOCTORALE
DE MATHEMATIQUES
HADAMARD

Titre: Conception d'algorithmes d'optimisation multiobjective et analyse théorique de stratégies d'évolution

Mots clés: CMA-ES, COMO-CMA-ES, Sofomore, UHVI, xNES, CSA-ES

Résumé: Nous avons analysé les comportements linéaires asymptotiques de stratégies d'évolution avec recombinaison, avec un step-size adaptatif, sur une classe de fonctions très large constituée des fonctions scaling-invariant. Notre cadre d'analyse inclut deux stratégies d'évolution connues que sont xNES et CSA-ES. Notre principale condition de convergence est que le logarithme du step-size croît en espérance sur les fonctions linéaires non triviales, ce qui est la condition optimale qu'on puisse obtenir. Dans le cadre multiobjectif, nous avons introduit un indicateur de performance appelé UHVI, qui crée un biais en faveur des zones non explorées du Pareto set. Nous avons developpé un système d'algorithmes appelé Sofomore, instancié avec le CMA-ES pour obtenir le COMO-CMA-ES. Un package Python appelé pycomocma est publié pour ce travail sur ce lien : https://github.com/CMA-ES/pycomocma.

Title: Design of multiobjective optimization algorithms and theoretical analysis of evolution strategies

Keywords: CMA-ES, COMO-CMA-ES, Sofomore, UHVI, xNES, CSA-ES

Abstract: We have analyzed linear behaviors of step-size adaptive evolution strategies with recombination mechanism, on a wide class of functions that are the scaling-invariant functions. Our framework includes two known step-size adaptation mechanisms derived from the Exponential Natural Evolution Strategy (xNES) and the Cumulative Step-size Adaptation without cumulation, where the CSA with cumulation is the default step-size mechanism of CMA-ES. Our main condition for convergence is that the logarithm of the step-size increases in expectation on non-zero linear functions, which is tighter than similar conditions found in the literature, for other evolution strategies. It is in fact the tightest condition we can have, as it is equivalent to the geometric divergence of the step-size on non-zero linear functions.
In the multiobjective optimization side, we introduce a performance indicator called UHVI: Uncrowded Hypervolume Improvement. It creates a search bias towards the uncovered domain of the Pareto set. We then develop a framework of multiobjective optimization algorithms called Sofomore. We instantiate that framework with the single-objective optimizers CMA-ES, to obtain the COMO-CMA-ES multiobjective algorithm. The pycomocma Python package available at https://github.com/CMA-ES/pycomocma is released for this work.

Institut Polytechnique de Paris

91120 Palaiseau, France