Titre: Title:	Contributions to Multiobjective Blackbox Optimization
Auteur: Author:	Ludovic Salomon
Date:	2022
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Salomon, L. (2022). Contributions to Multiobjective Blackbox Optimization [Thèse de doctorat, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/10355/

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Contributions to multiobjective blackbox optimization

LUDOVIC SALOMON
Département de mathématiques et génie industriel

Thèse présentée en vue de l'obtention du diplôme de *Philosophiæ Doctor*
Mathématiques

Mai 2022

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Cette thèse intitulée :

Contributions to multiobjective blackbox optimization

présentée par **Ludovic SALOMON**
en vue de l'obtention du diplôme de *Philosophiæ Doctor*
a été dûment acceptée par le jury d'examen constitué de :

Michel GENDREAU, président
Sébastien LE DIGABEL, membre et directeur de recherche
Jean BIGEON, membre et codirecteur de recherche
Youssef DIOUANE, membre
Ana Luísa CUSTÓDIO, membre externe

DEDICATION

To my four grand-parents.

ACKNOWLEDGEMENTS

Un seul mot, usé, mais qui brille comme une vieille pièce de monnaie: Merci !

Pablo Neruda

Writing these sentences is definitely not an easy task. Of course, as many students before me, I am afraid of forgetting people who have contributed to the finalization of this work (and there are many); the deadline to submit the manuscript is coming and I need to hurry to finish in time. But it is also the realization that typing these lines may close a special chapter of my life. As an informal conclusion, I need to dedicate to it a sufficient effort. Because it gives credit where it is due; because it opens a small window on the personality of the author, his/her (poor in my case) mastery of the language, a slice of his/her daily life. The exercise, less formal than the redaction of the main scientific material, less constrained by the ruthless Antidote software, allows it. In other terms (in my opinion), the Acknowledgements section reminds us that behind any scientific contribution, real people are hidden.

I would firstly like to thank my two advisors Prof. Le Digabel and Prof. Bigeon, without who this journey would have never begun. I am deeply grateful to both of you for giving me the opportunity to embark for an internship master, then a PhD. I want to thank you for your patience (e.g., reading the mails that I regularly sent you each week during four years), discussions, your supportive mentoring and kindness, the freedom to let me experiment or engage in annex research activities (teaching, supervising, My 180s thesis, and so on). I have learnt a lot from you, and proudly repeating myself, I thank you once again !

I would also like to thank my two other co-authors: Prof. Audet, for his help and explanations about nonsmooth calculus; and M.Sc.A. Cartier, who gave me the opportunity to deepen her work in a first article. I would like to thank Prof. Custódio for inviting me to present my research in Lisboa (such an extraordinary city), and for the subsequent discussions about multiobjective derivative-free optimization.

My thanks go to the committee members: Prof. Gendreau, Prof. Diouane and Prof. Custódio. It is an honour to have three highly respected members of the optimization community to evaluate my work. I am very grateful to you. I would also like to thank the representative Prof. Bourgault.

I am indebted to the following people who lent me technical and administrative support: Marie, Karine and Marilyne for their assistance and the organization of the numerous conferences given at Gerad; Pierre, Edoh, and Khalid, I apologize for the technical issues that

I often caused. I thank you for the numerous computation networking techniques that you taught me (unfortunately, I did not manage to transform the network of computers at Gerad into my personal supercomputer). I do not forget the administration from Polytechnique Montréal: especially, a huge thank for Melisa and Soraya for their help and the attentive track of my fellowship.

I would also like to thank Prof. Jomphe and Prof. Adjengue for giving me the opportunity to teach during several semesters; Prof. Batailly for entrusting me with an IICAP tutorial; and Mrs Guo for her assistance, availability, and kindness.

Money is the backbone of academic research. My gratitude goes to the IVADO institute and the CRSNG, which provided me with a substantial income. I also would like to thank the Gerad and the JuMP foundation for the conference fees.

This PhD would not have been the same without all the great people I have met. I hope I did not forget someone. From Montréal, a big thank (between others) to Alexis, Lisa, Adrien, Solène, Eloïse, Pierre-Yves, Dounia, Joseph, Christian, Marie, Florian, Mathieu T, B, and G, Julie, Stéphane, Vinicius, Carmen, Damoon, Emmanuel, Antoine, Maha, Illarya, Marie-Ange and Fanny. I really enjoyed the tips and tricks to get a fellowship or free food/coffee, the week-ends in chalets, parties, discussions, support from graph theory, and so many other moments that I would mention if the number of pages was limitless. Over the Atlantic Ocean, a big thank (between others) to Baptiste, Louis, Juliette, Audrey, Clément, Julian, Julien, Claire, Raphaël, Matthieu, Valentin, Vincent, Alexandre and Pierre who were always available to share and create precious memories each time I was coming back to France, ...

To conclude, I am deeply blessed to my parents for their love, support, education and their advices. You are certainly the first sources of inspiration which encourage me to pursue in this direction. I would like to thank my brothers Doan and Thao, my sister My Lan; the enlarged family, i.e., Laure, Marion, Augustin, Juliette and my two stepparents Olivier and Valérie for their support, love, and the moments spent together; my grand-parents/ông bà Xuan, Marie-Claude, Jean-Jacques and Claire to which this work is dedicated, and the whole family in general. Finally, a special thank to Suzanne for her patience, love and still positive attitude, which always pushes me to surpass myself since these three years (for the better I hope).

RÉSUMÉ

L'optimisation de boîtes noires et l'optimisation sans dérivées sont deux disciplines mathématiques qui traitent des problèmes ne possédant pas de formulation analytique exploitable, rendant caduques les techniques d'optimisation usuelles fondées sur la dérivation. Ces types de problèmes découlent souvent de simulations numériques de systèmes complexes dont la structure ne peut être exploitée, appelées communément des boîtes noires. En outre, la complexification de ces systèmes a mis en évidence les limites d'un modèle d'optimisation unique ne prenant en compte qu'un seul objectif. En effet, de nombreux problèmes d'ingénierie mettent en jeu plusieurs critères, souvent contradictoires, que l'on cherche à optimiser en même temps. L'optimisation multiobjectif intègre ce cadre. La résolution ne donne pas une solution unique, mais un ensemble de points reflétant les différents compromis qui existent entre les objectifs.

Les algorithmes qui s'attaquent à des problèmes de boîtes noires multiobjectif ne sont pas aussi développés que leurs homologues mono-objectif. Des heuristiques stochastiques et coûteuses, pouvant poser des problèmes de robustesse, ont longtemps dominé le champ de recherche. L'extension de solveurs mono-objectif déterministes et efficaces pour des boîtes noires possédant des preuves de convergence à l'optimisation multicritères n'a démarré que depuis une dizaine d'années et en est encore à ses débuts.

Cette thèse propose donc de nouvelles contributions théoriques et pratiques pour l'optimisation multiobjectif déterministe de boîtes noires.

La multiplication de nouveaux algorithmes multiobjectif nécessite d'autant plus le développement d'outils d'analyse afin de pouvoir valider leurs performances et les comparer entre eux, dans le but de guider l'utilisateur vers la ou les méthodes les plus adaptées à son application. La première contribution de cette thèse est une étude sur les indicateurs de performance pour l'optimisation multiobjectif, publiée dans *European Journal of Operational Research*. Ces derniers évaluent la qualité de l'ensemble des solutions retournées par une méthode multiobjectif sur un problème donné. Cette étude synthétise les propriétés de 63 indicateurs issus de la littérature scientifique, les classifie selon leurs propriétés et liste leurs principaux usages.

La deuxième contribution est une nouvelle extension de la méthode de recherche directe par treillis adaptatifs (MADS) à l'optimisation multiobjectif de boîtes noires. Ce nouvel algorithme, DMulti-MADS, s'inspire des algorithmes DMS et BiMADS, créés pour résoudre ce type de problèmes. Nous démontrons que DMulti-MADS garantit la convergence vers un

ensemble de points localement non dominés en se basant sur le calcul non lisse de Clarke sous des hypothèses classiques d'algorithmes de recherche directe. Afin de pouvoir valider sa performance, une extension des profils de données à l'optimisation multiobjectif est introduite, basée sur les recommandations de l'étude précédente. Des tests numériques montrent que Dmulti-MADS est compétitif par rapport à des algorithmes de l'état de l'art. Il est également plus performant pour des budgets d'évaluation de taille faible ou moyenne que BiMADS. Cet algorithme a fait l'objet d'une publication parue dans *Computational Optimization and Applications*.

La dernière contribution propose deux nouvelles stratégies appliquées à DMulti-MADS pour gérer des contraintes d'inégalité de type boîtes noires. Les deux stratégies utilisent une fonction de violation des contraintes, que l'on cherche à minimiser en même temps que les objectifs. Ces deux nouvelles méthodes, nommées DMulti-MADS-PB et DMulti-MADS-TEB, conservent les propriétés de convergence de DMulti-MADS. Elles sont comparées par rapport à une méthode de pénalité ainsi que les algorithmes de l'état de l'art BiMADS, DFMO et NSGA-II pouvant gérer ce type de contraintes. Les tests numériques montrent qu'elles affichent des performances supérieures aux autres algorithmes sur un ensemble de problèmes analytiques issus de la littérature. Elles surpassent également les autres méthodes sur deux des trois problèmes d'ingénierie réels considérés.

ABSTRACT

Blackbox optimization and derivative-free optimization are two subdisciplines of numerical optimization. They address problems which do not have an exploitable analytical formulation of objective and/or constraint functions. This absence of structure makes the deployment of usual derivative-based solvers impossible. Such problems commonly arise from numerical simulations modelling complex systems. Besides, real engineering situations have highlighted the limits of single-objective modelling. Indeed, many engineering problems involve several and often contradictory criteria, which one wants to optimize at the same time. Multiobjective optimization takes this framework into account. The resolution of such problems does not give a single optimal solution, but a set of points which represent the (potential) best trade-offs between the different objectives.

Algorithms which tackle multiobjective blackbox problems are not as developed as their single-objective counterparts. For a very long time, stochastic and costly heuristics, with questionable reliability, have dominated the field of research. The generalization of deterministic and efficient single-objective methods to multiobjective blackbox optimization has only started about ten years ago and is still at its beginnings.

This thesis therefore aims to explore further this research area by bringing new theoretical and practical contributions in deterministic multiobjective blackbox optimization.

The increase in the development of new multiobjective methods requires all the more the conception of pertinent benchmarking tools to assess their performance and compare them. When done well, such analysis can be of great value to guide the user in the selection of the most suited algorithm to solve his/her applications. The first contribution of this thesis is a review on performance indicators for multiobjective optimization, published in *European Journal of Operational Research*. These performance indicators evaluate the quality of all trade-off solutions generated by a multiobjective solver on a problem. This review summarizes the properties of 63 indicators from the scientific literature, classifies them and presents main applications of these quality metrics.

The second contribution is a new extension of the Mesh Adaptive Direct Search (MADS) method to multiobjective blackbox optimization. This new algorithm, denoted as DMulti-MADS, is inspired by the two multiobjective blackbox solvers DMS and BiMADS. We prove that DMulti-MADS is guaranteed to converge towards a set of locally non-dominated points based on the Clarke calculus, under common directional search assumptions. To validate its performance, new data profiles for multiobjective optimization are introduced by following the

recommendations of the previous survey. Numerical experiments show that Dmulti-MADS is competitive with state-of-the-art algorithms. It surpasses BiMADS when considering a small to medium budget of function evaluations. It has been published in *Computational Optimization and Applications*.

The last contribution proposes two new strategies applied to DMulti-MADS to handle black-box inequality constraints. Both strategies use a single constraint violation function which aggregates constraints. The goal is to minimize the constraint violation function and at the same time the objectives. We prove that these two approaches retain the convergence properties of DMulti-MADS. These two methods are compared with a penalty-based approach and the state-of-the-art solvers BiMADS, NSGA-II and DFMO, which can handle these constraints. Numerical experiments show that these two approaches are more efficient than the other algorithms on a set of analytical problems taken from the literature. They also outperform the other solvers on two of the three real engineering problems considered.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

BiMADS	Biobjective Mesh Adaptive Direct Search
BOTR	BiObjective Trust-Region
CS	Coordinate Search
DFMO	Derivative-Free MultiObjective Optimization
DFN	Derivative-Free Nonsmooth
DMS	Direct MultiSearch
DMulti-MADS	DMultiobjective Mesh Adaptive Direct Search
EB	Extreme Barrier
EMT	Expensive Multiobjective Trust-region
IF	Implicit Filtering
GPS	Generalized Pattern Search
GSS	Generating Search Set
KTCQ	Kuhn-Tucker Constraint Qualification
LICQ	Linear Independence Constraint Qualification
MADS	Mesh Adaptive Direct Search
MHT	Multiobjective Heterogeneous Trust-region
MOIF	MultiObjective Implicit Filtering
MOO	MultiObjective Optimization
MultiMADS	Multiobjective Mesh Adaptive Direct Search
NSGA-II	Non-dominated Sorting Genetic Algorithm II
PB	Progressive Barrier
SOO	Single-Objective Optimization

TEB	Two-phase Extreme Barrier
(MO)BBO	(MultiObjective) BlackBox Optimization
(MO)DFO	(MultiObjective) Derivative-Free Optimization
(M)KKT	(Multiobjective) Karush-Kuhn-Tucker

LIST OF SYMBOLS

General notations and symbols

a, α	Scalar values
A	Countable set
$\mathbf{A}$	Matrix
$\mathcal{A}$	Non countable set
$\mathbf{a}$	Vector
$\mathbf{a}^j$	Set element
$\mathbf{a}_i$	Vector element
$[\bullet, \ldots, \bullet]$	Matrix of column vectors
$\{\bullet, \ldots, \bullet\}$	Set
$[\bullet; \bullet]$	Closed interval
$(\bullet; \bullet)$	Open interval
$\mathbb{N}$	Set of natural numbers
$\mathbb{N}^*$	Set of strictly positive natural numbers
$\mathbb{R}$	Set of real numbers
$\mathbb{R}^n$	Set of n-dimensional real numbers
$\mathbb{R}^m_+$	m-dimensional positive orthant
$\mathbb{R}^{n \times m}$	Set of matrices of dimensions $n \times m$
$\mathbb{Z}$	Set of integers
$\mathcal{B}$	Ball
$\mathcal{C}$	Cone
$\mathcal{N}$	Neighbourhood
$\mathbf{I}_n$	Identity matrix of size n
$\nabla(\bullet)$	Gradient
$\nabla^2(\bullet)$	Hessian
$\lvert \bullet \rvert$	Set cardinality or absolute value (of a scalar)
$\lVert \bullet \rVert_p$	p-norm of a vector

Symbols related to mathematical optimization

Ω	Feasible set
$\mathcal{Y}$	Objective space
f	Objective function
c	Constraint functions
$\mathcal{J}$	Number of constraint functions
m	Dimensions of the objective space
n	Dimensions of the decision space
$\mathbf{x}$	Decision vector
$\mathbf{y}$	Objective vector
$\mathbf{r}$	Reference point
$\mathbf{l}$	Lower bound
$\mathbf{u}$	Upper bound
$\mathcal{Y}_P$	Pareto front
$\mathcal{X}_P$	Pareto set
$\mathbf{y}^I$	Ideal objective vector
$\mathbf{y}^N$	Nadir objective vector
$\mathbf{y}^U$	Utopian objective vector
Y_N	Pareto front approximation
X_N	Pareto set approximation
$\mathcal{T}_\Omega^{Co}(\mathbf{x})$	Contingent cone to Ω at $\mathbf{x}$
$\mathcal{T}_\Omega^{Cl}(\mathbf{x})$	Clarke tangent cone to Ω at $\mathbf{x}$
$\mathcal{T}_\Omega^{H}(\mathbf{x})$	Hypertangent cone to Ω at $\mathbf{x}$
ν	Lipschitz constant
$\mathbf{d}$	Direction

Note : The text tries to remain close to the notations presented above. However, it may deviate from these conventions, particularly from an article to another. If it is the case, an explanation of the notations used in the context is always given.

CHAPTER 1 INTRODUCTION

L'organisation de ce travail tient au souci de faire la part égale à l'exploration de trois thèmes.

Claire Salomon-Bayet

1.1 Context

Optimization is a branch of mathematics which aims to specify and compute the maxima/minima of a function, whose solution may be submitted to some constraints. Its applications are numerous and range from the design of energy networks, imaging or finance to numerical analysis and optimal control. The increasing complexity of engineering systems coupled to more powerful computing machines for these last twenty years has resulted in the conception and usage of a class of new nonlinear optimization techniques, designed as *blackbox optimization* (BBO).

A blackbox is any process such that when provided an input, it returns one or several outputs and the inner workings of this process are not analytically available. The user deals with an objective function for which no partial information or structure is available, i.e., derivatives. Thus, no gradient-based method can be considered, as the blackbox function may be nonsmooth, non-differentiable, or non-convex and subject to noise. Sometimes, blackbox evaluations are the outcome of expensive simulations (e.g., plane motor design) running from a few minutes to several hours, limiting the number of calls to the blackbox.

Blackbox optimization (BBO) is then the "study of design and analysis of algorithms that assume the objective and/or constraint functions are given by blackboxes" [20]. Its goal is the conception of numerical methods to find the most appropriate combination of input parameters corresponding to a feasible optimum given by the outputs of the blackbox. BBO is distinct from *derivative-free optimization* (DFO) [67], another subdiscipline, which is often quoted in the BBO literature. As its name indicates, DFO is the study and design of algorithms which do not use derivatives. However, they may exist although they are not available.

Real engineering problems have also highlighted the limits of a universal single-objective model. For example, in some design problems, the modeller wants to maximize the solidity of a structure, and at the same time minimize its weight and the construction cost. In finance, an investor could look for a portfolio with high returns and low volatility. In machine learning,

the design of classification systems which are at the same time precise, robust and energy-efficient has become an important issue. Often, these criteria are in conflict: the improvement of one objective can be detrimental to another. *Multiobjective optimization* (MOO) methods aim to provide the decision maker with a set of feasible solutions representing the best trade-offs between the different objectives. Its knowledge enables him/her to visualize the consequences of his choices, and to make the convenient decision according to him/her.

Historically, one can trace the beginnings of single-objective BBO to the 50s, with the conception of the coordinate search method [113]. The development of a convergence analysis framework [244] in the 90s has relaunched the development of new convergent-based methods for DFO and BBO. Since then, the field has taken more and more importance, as the following textbooks [20, 67] and surveys reflect it [21, 160, 164]. Unfortunately, multiobjective approaches are not generally as established as single objective optimization algorithms, especially for blackbox applications. Indeed, for a long time, the only methods to tackle multiobjective blackbox optimization problems were heuristics, such as evolutionary multiobjective optimization algorithms [88, 89], particule-swarm algorithms [207], and so on [127, 237]. Most of these solvers are stochastic, which can raise potential reliability concerns. They also require an important budget of evaluations (e.g., from 20,000 to 50,000, as in [86, 144]) to work, which can be impracticable when the blackbox function is costly. Note that the substitution of costly functions by cheap interpolation models to guide the search (see for example [196]) has partly removed this last issue.

Since some years, researchers have deployed many efforts to extend convergent-based deterministic single-objective BBO methods to multiobjective BBO [27, 28, 74, 126, 180, 223]. This thesis is in line with these works. More precisely, it explores different ways to extend the single-objective BBO Mesh Adaptive Direct Search (MADS) algorithm [15] to multiobjective BBO. MADS [15] is a direct search method designed to address the resolution of constrained single-objective BBO problems. It provides an extremely strong convergence analysis framework, based on the Clarke calculus [59] but is flexible enough to integrate efficient heuristics (see [29, 55, 66]). The NOMAD software [167] implements a state-of-the-art version of the MADS algorithm.

Between 2008 and 2010, two extensions of the MADS algorithm to multiobjective BBO optimization were developed: BiMADS [27] which takes into account two objectives, and MultiMADS [28] which deals with more than two objectives. Both methods use the scalarization approach. It transforms the initial MOO problem into a succession of single-objective subproblems. Each subproblem can then be tackled by a single-objective optimization method, in this case MADS. Consequently, BiMADS and MultiMADS directly benefit from the con-

vergence analysis of MADS and its new improvements. However, the main drawback is the management of the total budget of evaluations, which has to be split between all subproblems. Too much effort can be allocated to the resolution of a particular subproblem at the expense of a more thorough exploration of the objective space for potential new interesting solutions. In 2011, Custódio et al. [74] proposed the Direct MultiSearch (DMS) framework, which extends all direct search methods to MOO and does not aggregate objective functions. Possessing a good performance, DMS has inspired the development of new deterministic multiobjective BBO algorithms since its publication.

1.2 Research objectives

This thesis aims to bring new methodological and theoretical contributions to the field of multiobjective BBO. Precisely, the objectives of this work are the following:

- The development of new algorithms possessing strong convergence properties (which is a guarantee of reliability) and performance (relative to the number of function evaluations required to obtain a good solution set). This work then prioritizes the development of new deterministic-based convergent methods over heuristic approaches.
- The description of a rigorous benchmark framework to assess the performance of multiobjective BBO algorithms. Indeed, the existing tools possess some drawbacks (e.g., noncompliance with the dominance relation, impossibility to rank more than two solvers), which can mislead their interpretation.
- As a byproduct, the thorough comparison of several state-of-the-art multiobjective DFO algorithms with this new framework. One could claim it has already been done when these solvers were introduced. However, in research, additional experimental validation or refutation is always pertinent.
- Reproducibility. An implementation of the proposed algorithms should be, of course, available to facilitate the reproduction of experiments, as a baseline for future benchmarking, or to test them on new engineering applications. Ideally, the methods should also be easy to understand and to reimplement. The number of algorithm parameters should be kept at a minimum, which limits the intervention of an expert.

1.3 Research contributions

This thesis proposes the three following contributions, each of them corresponding to one chapter/article.

- The first contribution is a general survey on performance indicators in multiobjective optimization. Performance indicators enable to evaluate the quality of a solution set produced by a MOO algorithm, and consequently, its performance, hence their name. This survey provides a classification of an important number of them, a list of their properties and limitations, and some of their use.

- Inspired by the deterministic algorithms DMS [74], BiMADS [27] and DFMO [180], the second contribution is a new extension of MADS to MOO called DMulti-MADS. Although specialized on bound-constrained problems, this method can be easily adapted to deal with general constraints (via an extreme barrier approach). It is proved this algorithm generates sequences of points which converge to a set of locally non-dominated solutions (which is similar to DMS, which is guaranteed to generate at least one local Pareto point). To validate its performance over state-of-the-art algorithms, data profiles for MOO are defined and numerical experiments are conducted on a standard set of academic problems.

- In the last contribution, we propose the integration of two new inequality constraint-handling approaches into the DMulti-MADS algorithm. Both use a constraint violation function. The first is a two-phase approach, where the search for the feasible point is prioritized over the improvement of objectives. The second is an extension of the progressive barrier for single-objective optimization [16] to multiple objectives. Numerical experiments are conducted on analytical benchmarks and three "real" engineering applications, validating these two approaches over existing state-of-the-art algorithms.

1.4 Plan

Chapter 2 presents a critical review of the literature. It provides a summary of fundamental concepts of MOO, a statement of central optimality conditions in MOO, and a description of the main algorithms mentioned in the three contributions. Chapter 3 is dedicated to the organization of this work. It also details the expected research results of this thesis. Chapters 4, 5 and 6 represent the contributions of this thesis. They are followed by a general discussion and conclusion.

CHAPTER 2 LITERATURE REVIEW

Il avait vu au cours d'un demi-siècle plusieurs générations d'idées tomber en poussière.

Marguerite Yourcenar, *L'oeuvre au noir*

This chapter is divided into three parts. Section 2.1 introduces the main concepts, notations and optimality conditions of multiobjective optimization required to understand the research. Section 2.2 is dedicated to the description of some of the most important single-objective DFO algorithms. Indeed, although this thesis focuses on the conception of new deterministic multiobjective BBO methods, their design incorporates many mechanisms taken from single-objective DFO/BBO algorithms. It is then relevant to detail them. Section 2.3 offers a summary of the state-of-the-art convergent-based multiobjective DFO algorithms, mentioned in the next chapters.

2.1 Multiobjective optimization: concepts and notations

2.1.1 Definitions

Optimization, as a mathematical discipline, aims to obtain the *maximum* or *minimum* of one or several *objective functions*, by finding adequate *decision variables* in the domain of the objective functions. These decision variables can be subjected to equality or inequality constraints.

Generically, a multiobjective optimization problem is of the form

$$MOP \ : \ \min_{\mathbf{x}\in\Omega} f(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x}))^\top$$

where $\Omega \subseteq \mathbb{R}^n$ is the *feasible decision space*, i.e., the decision vector $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)^\top$ must belong to Ω when presented to the decision maker. $\mathbb{R}^n$ is the *decision space*. The image of the feasible decision space by the objective function f is the *feasible objective space*, denoted as $\mathcal{Y} = f(\Omega)$. $\mathcal{Y}$ is a subset (not always strict) of $\mathbb{R}^m$, named the *objective space*, whose elements are called *objective vectors*. Note that objective vectors may not always belong to $\mathcal{Y}$.

Contrary to single-objective optimization (SOO), the objective function is not a real scalar-valued function but a vector-valued function, composed of $m \geq 2$ scalar-valued objective

functions $f_i : \mathbb{R}^n \to \mathbb{R}$ for $i = 1, 2, \ldots, m$. By convention, all objectives f_i for $i = 1, 2, \ldots, m$ are to be minimized. The maximization of a criterion f_i is equivalent to the minimization of $-f_i$. The conversion to the standard form is then straightforward.

In SOO, the ranking of different feasible solutions according to their objective function is trivial, as the set $\mathbb{R}$ is totally ordered. In MOO, minimizing simultaneously all objective functions is not always possible, due to the potential conflicting nature of the different criteria. One then needs to introduce a partial relation order on $\mathbb{R}^m$ to compare objective vectors. The following cone order relation is adopted [98].

Definition 1 (Comparison between objective vectors [79])**.** Given two objective vectors $\mathbf{y}^1$ and $\mathbf{y}^2$ in the objective space $\mathbb{R}^m$,

- $\mathbf{y}^1 \leqq \mathbf{y}^2$ ($\mathbf{y}^1$ *weakly dominates* $\mathbf{y}^2$) if and only if $\mathbf{y}^1_i \leq \mathbf{y}^2_i$ for $i = 1, 2, \ldots, m$.
- $\mathbf{y}^1 \leq \mathbf{y}^2$ ($\mathbf{y}^1$ *dominates* $\mathbf{y}^2$) if and only if $\mathbf{y}^1 \leqq \mathbf{y}^2$ and $\mathbf{y}^1 \neq \mathbf{y}^2$.
- $\mathbf{y}^1 < \mathbf{y}^2$ ($\mathbf{y}^1$ *strictly dominates* $\mathbf{y}^2$) if and only if $\mathbf{y}^1_i < \mathbf{y}^2_i$ for $i = 1, 2, \ldots, m$.
- In the case when neither $\mathbf{y}^1 \nleqq \mathbf{y}^2$ nor $\mathbf{y}^1 \ngeqq \mathbf{y}^2$, $\mathbf{y}^1$ and $\mathbf{y}^2$ are said to be *incomparable* or *non dominated*.

With this definition, it is then possible to compare decision vectors, using the Pareto dominance relation order.

Definition 2 (Comparison between decision vectors [27])**.** Given two decision vectors $\mathbf{x}^1$ and $\mathbf{x}^2$ in the feasible decision space Ω,

- $\mathbf{x}^1 \preceq \mathbf{x}^2$ ($\mathbf{x}^1$ *weakly dominates* $\mathbf{x}^2$) if and only if $f(\mathbf{x}^1) \leqq f(\mathbf{x}^2)$.
- $\mathbf{x}^1 \prec \mathbf{x}^2$ ($\mathbf{x}^1$ *dominates* $\mathbf{x}^2$) if and only if $f(\mathbf{x}^1) \leq f(\mathbf{x}^2)$.
- $\mathbf{x}^1 \prec\prec \mathbf{x}^2$ ($\mathbf{x}^1$ *strictly dominates* $\mathbf{x}^2$) if and only if $f(\mathbf{x}^1) < f(\mathbf{x}^2)$.
- $\mathbf{x}^1 \sim \mathbf{x}^2$ ($\mathbf{x}^1$ and $\mathbf{x}^2$ are *incomparable* or *non dominated*) if $f(\mathbf{x}^1)$ and $f(\mathbf{x}^2)$ are incomparable.

This definition is illustrated on Figure 2.1 for a biobjective minimization problem, where $\Omega \subset \mathbb{R}^2$. The objective function f maps the feasible decision space Ω to the feasible objective space $\mathcal{Y} \subset \mathbb{R}^2$, delimited by the curve on the right part of the figure. Relatively to the decision vector $\mathbf{x}^1 \in \Omega$, three zones are defined. The *dominance zone* is the set of feasible decision

vectors which dominate $\mathbf{x}^1$. The *dominated zone* is the set of feasible decisions vectors which are dominated by $\mathbf{x}^1$. The *indifference zone* is the set of feasible decision vectors which are incomparable to $\mathbf{x}^1$. The definition of these three zones can also be applied to objective vectors. In Figure 2.1, $\mathbf{x}^1$ dominates $\mathbf{x}^3$ and is dominated by $\mathbf{x}^2$; the pairs of decision vectors $(\mathbf{x}^2, \mathbf{x}^4)$ and $(\mathbf{x}^3, \mathbf{x}^4)$ are indifferent.

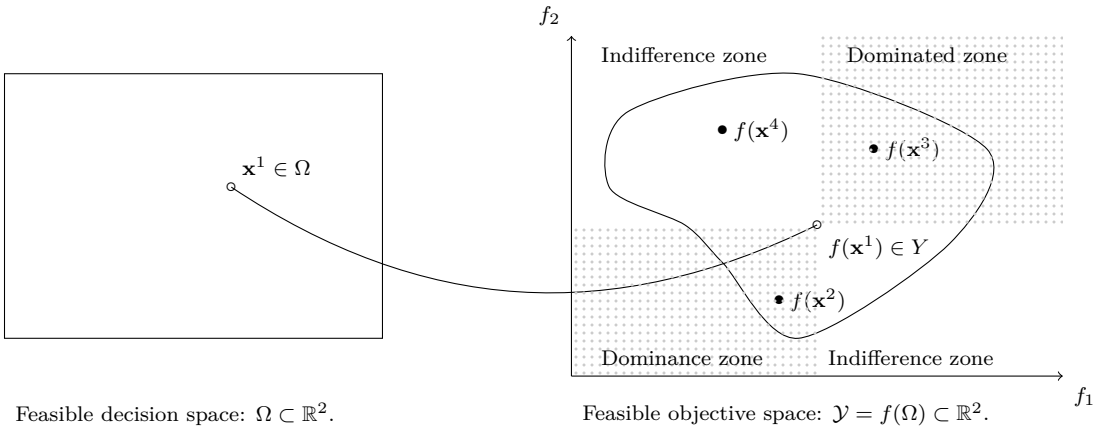


Figure 2.1 An illustration of Pareto dominance for a minimization biobjective problem with 2 variables (inspired by [27]): $\mathbf{x}^2 \prec \mathbf{x}^1 \prec \mathbf{x}^3$, $\mathbf{x}^2 \sim \mathbf{x}^4$ and $\mathbf{x}^3 \sim \mathbf{x}^4$.

The definition above enables to introduce the notion of Pareto optimality for MOO.

Definition 3 (Pareto optimality)**.** A decision vector $\mathbf{x}^\star \in \Omega$ is said to be (globally) *Pareto optimal* if there does not exist any other decision vector $\mathbf{x} \in \Omega$ such that $\mathbf{x} \prec \mathbf{x}^\star$.

As in SOO, it may be difficult to obtain global solutions. In this case, one should design methods which guarantee at least local optimality conditions.

Definition 4 (Locally Pareto optimality)**.** A decision vector $\mathbf{x}^\star \in \Omega$ is said to be *locally Pareto optimal* if there does not exist any other decision vector $\mathbf{x} \in \Omega \cap \mathcal{N}(\mathbf{x}^\star)$ such that $\mathbf{x} \prec \mathbf{x}^\star$, where $\mathcal{N}(\mathbf{x}^\star)$ is a neighbourhood of $\mathbf{x}^\star$.

A Pareto optimal decision vector is always locally Pareto optimal. The converse is not always true unless some specific assumptions are given on the problem.

The set of all Pareto optimal solutions (potentially infinite [98]) is called the *Pareto set* denoted by $\mathcal{X}_P$ and its image by the objective function f is the *Pareto front* denoted by $\mathcal{Y}_P$. Similarly, the image of a set of locally Pareto optimal decision vectors by the objective function f is a *local Pareto front*. The Pareto front for a problem with m objectives is at most of dimension $m-1$, and may be non smooth or discontinuous. For example, for a biobjective

problem, the Pareto front can be a curve or a point or a combination of curves/points or the empty set. The *ideal objective vector* and *nadir objective vector*, as defined below, respectively provide the Pareto front with a lower and an upper bound.

Definition 5 (Ideal objective vector [205])**.** The *ideal objective vector* $\mathbf{y}^I \in \mathbb{R}^m$ is defined as $\mathbf{y}^I = (\mathbf{y}_1^I, \mathbf{y}_2^I, \ldots, \mathbf{y}_m^I)^\top$ whose elements are:

$$\mathbf{y}_i^I = \min_{\mathbf{x}\in\Omega} f_i(\mathbf{x}), \text{ for } i = 1, 2, \ldots, m.$$

Definition 6 (Nadir objective vector [205])**.** The *nadir objective vector* $\mathbf{y}^N \in \mathbb{R}^m$ is defined as $\mathbf{y}^N = (\mathbf{y}_1^N, \mathbf{y}_2^N, \ldots, \mathbf{y}_m^N)^\top$ whose elements are:

$$\mathbf{y}_i^N = \max_{\mathbf{x}\in\mathcal{X}_P} f_i(\mathbf{x}), \text{ for } i = 1, 2, \ldots, m.$$

Definition 7 (Utopian objective vector [205])**.** Given a scalar $\epsilon > 0$, the *utopian objective vector* $\mathbf{y}^U \in \mathbb{R}^m$ is defined as $\mathbf{y}^U = (\mathbf{y}_1^U, \mathbf{y}_2^U, \ldots, \mathbf{y}_m^U)^\top$ whose elements are:

$$\mathbf{y}_i^U = \mathbf{y}_i^I - \epsilon, \text{ for } i = 1, 2, \ldots, m.$$

If the ideal objective vector belongs to the Pareto front, the Pareto front is reduced to this objective vector, and there exists a feasible decision vector whose image by f is the ideal objective vector. Practically, as the different objectives are often contradictory, this situation does not generally occur. The ideal objective vector is often used as a reference point in the objective space to reach for. The utopian objective vector indicates a goal which dominates all Pareto optimal objective vectors.

In Figure 2.2, a piecewise-continuous biobjective Pareto front for a minimization optimization problem and its ideal and nadir objective vectors are illustrated. Note that the point $\hat{\mathbf{x}} \in \Omega$ is not Pareto optimal, as one can find some elements of $\mathcal{Y}_P$ which dominate $f(\hat{\mathbf{x}})$.

Remark. The field of optimization which uses a comparison based on a partial ordering space structure provided as a fixed, pointed, convex and closed cone, and not only $\mathbb{R}_+^m$, is designed as *vector optimization*. Vector optimization is thus a generalization of MOO. This thesis does not focus on vector optimization problems: an interested reader can refer to [98, 148] for more details on this topic.

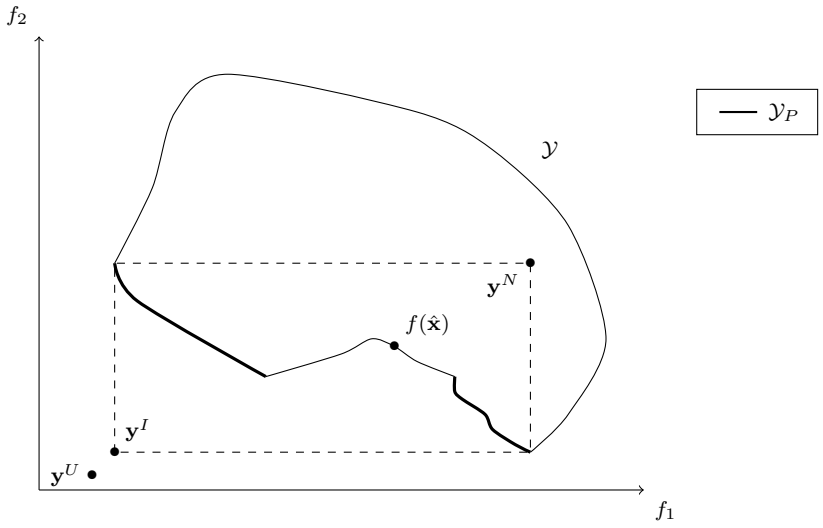


Figure 2.2 Pareto front $\mathcal{Y}_P$ and ideal, utopian and nadir objective vectors $\mathbf{y}^I$, $\mathbf{y}^U$ and $\mathbf{y}^N$ in the feasible objective space $\mathcal{Y}$ for a biobjective minimization problem.

2.1.2 Optimality conditions

Optimization algorithms are iterative. Starting from a first "guess", they generate a sequence of improved estimates, until they reach a stopping criterion. It is common for researchers to prove that the limit of the sequence generated by their methods satisfies necessary optimal conditions, under diverse assumptions (e.g. convexity or differentiability of the objectives, presence of constraints). This "certificate" brings to the user a certain reliability. Convergence analysis also provides a framework in which a given algorithm should simply "work".

This subsection gives a set of optimality conditions for MOO (the main references are [35, 59, 190, 201]) used along this thesis. As this work is about DFO and BBO, this subsection will only focus on zero order and first-order optimality conditions. The reader is invited to take a look at [190, 201] for more information about second-order optimality conditions results. This subsection equally omits proofs, which can be found in [35, 59, 190, 201].

Preliminary notations

For the rest of this thesis, the following notations will be used.

The notation $f \in \mathscr{C}^0$ signifies that the function f is *continuous* on its definition domain and the notation $f \in \mathscr{C}^k$ for $k \in \mathbb{N}^*$ signifies that the function f is k times *differentiable* on its definition domain and all its partial derivatives at order k are continuous. For example, $f \in \mathscr{C}^1$ on $\mathbb{R}^n$ means that f is differentiable on $\mathbb{R}^n$ and its gradient ∇f is continuous on $\mathbb{R}^n$.

Furthermore, convergence analysis of many DFO algorithms typically relies on the assumptions that involved functions or gradients are (locally) Lipschitz continuous on their definition domains.

Definition 8 (Lipschitz Continuous)**.** A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is *Lipschitz continuous* on a set $\Omega \subseteq \mathbb{R}^n$ if and only if there exists a scalar $\nu > 0$ such that $\forall(\mathbf{x}^1, \mathbf{x}^2) \in \Omega^2$,

$$\|f(\mathbf{x}^1) - f(\mathbf{x}^2)\| \leq \nu \|\mathbf{x}^1 - \mathbf{x}^2\|.$$

ν is denoted as the *Lipschitz constant* of f relatively to the set Ω.

A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is *locally Lipschitz* at $\mathbf{x}^* \in \Omega \subseteq \mathbb{R}^n$ (or Lipschitz near $\mathbf{x}$) if there exists a neighbourhood $\mathcal{N}(\mathbf{x}^\star)$ of $\mathbf{x}^*$ such that f is Lipschitz continuous on $\mathcal{N}(\mathbf{x}^*) \cap \Omega$.

Following [20], we use the following notation $f \in \mathscr{C}^{0+}$ to signify that f is locally Lipschitz continuous at all $\mathbf{x}$ on its definition domain; $f \in \mathscr{C}^{0+}$ *with constant* $\nu > 0$ means that f is Lipschitz continuous (of Lipschitz constant ν) on its definition domain. The notation $f \in \mathscr{C}^{1+}$ (with constant $\nu > 0$) signifies that $f \in \mathscr{C}^1$ and $\nabla f \in C^{0+}$ (with constant $\nu > 0$). Similarly, $f \in \mathscr{C}^{2+}$ (with constant $K > 0$) signifies that $f \in \mathscr{C}^2$ and $\nabla^2 f \in \mathscr{C}^{0+}$ (with constant $\nu > 0$).

Unconstrained smooth optimization

In this paragraph, the following unconstrained optimization multiobjective problem is considered:

$$\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x}))^\top,$$

where the objectives f_i for $i = 1, 2, \ldots, m$ are supposed to be continuously differentiable on $\mathbb{R}^n$. This assumption is reasonable in a DFO context where derivatives may exist even if they are not available.

The following necessary first order condition of Fritz-John type, adapted from [190], is then given.

Theorem 1 (Necessary first order condition for unconstrained multiobjective optimization)**.** *Let $\mathbf{x}^\star \in \mathbb{R}^n$ be locally Pareto optimal. If $f : \mathbb{R}^n \to \mathbb{R}^m$ is $\mathscr{C}^1$, then there exists $\lambda \in \mathbb{R}^m_+$, $\lambda \neq 0_{\mathbb{R}^m}$ such that*

$$\sum_{i=1}^{m} \lambda_i \nabla f_i(\mathbf{x}^\star) = 0.$$

When $m = 1$, one falls back on the classical necessary first order condition for unconstrained SOO, i.e. $\nabla f(x^\star) = 0$, where $f : \mathbb{R}^n \to \mathbb{R}$.

Several authors (e.g. [117]) use this alternative condition. Let $\mathbf{x}^\star \in \mathbb{R}^n$ be locally Pareto optimal. If $f : \mathbb{R}^n \to \mathbb{R}^m$ is $\mathscr{C}^1$, then for all vectors $\mathbf{d} \in \mathbb{R}^m$, there exists at least one index $i_0 \in \{1, 2, \ldots, m\}$ such that

$$\nabla f_{i_0}(\mathbf{x}^\star)^\top \mathbf{d} \geq 0.$$

In other words, there does not exist any vector $\mathbf{d} \in \mathbb{R}^n$ which is a descent direction for all objective functions f_i for $i = 1, 2, \ldots, m$. Can be demonstrated by absurd.

Constrained smooth optimization

In this paragraph, the following inequality constrained MOO problem is considered, as in [190]:

$$\min_{\mathbf{x}\in\Omega} f(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x}))^\top,$$

where $\Omega = \{\mathbf{x} \in \mathbb{R}^n : c_j(\mathbf{x}) \leq 0, j \in \mathcal{J}\}$, with $\mathcal{J}$ the set of inequality constraints. As in the unconstrained case, all objective functions f_i for $i = 1, 2, \ldots, m$ are assumed to be continuously differentiable on $\mathbb{R}^n$, as for the c_j for $j \in \mathcal{J}$ which describe the set of constraints.

The presence of constraints requires additional assumptions designed as *constraint qualifications*. Roughly speaking, constraint qualifications offer guarantees that the linearized feasible set (in terms of constraint gradients) is similar to the constraint set in the vicinity of an optimal feasible solution $x^\star$ [201]. Following [190], the *Kuhn-Tucker constraint qualification* is then presented.

Definition 9. Assume the constraints $c_j(x)$ for $j \in \mathcal{J}$ are $\mathscr{C}^1$ at $\mathbf{x}^\star \in \Omega$. The constrained multiobjective problem satisfies the *Kuhn-Tucker constraint qualification* at $\mathbf{x}^\star \in \Omega$ if for any $\mathbf{d} \in \mathbb{R}^n$ such that $\nabla c_j(\mathbf{x}^\star)^\top \mathbf{d} \leq 0$ for all $j \in \mathcal{J}(\mathbf{x}^\star)$, where $\mathcal{J}(\mathbf{x}^\star)$ is the set of active constraints at $\mathbf{x}^\star$, i.e.

$$\mathcal{J}(\mathbf{x}^\star) = \{j \in \mathcal{J} : c_j(\mathbf{x}^\star) = 0\},$$

there exists a function $a : [0, 1] \to \mathbb{R}^n$ continuously differentiable at 0 and some scalar $\alpha > 0$ such that $a(0) = \mathbf{x}^\star$, $c(a(t)) \leq 0$ for all $t \in [0, 1]$ and $a'(0) = \alpha\mathbf{d}$.

One can then give the *Karush-Kuhn-Tucker conditions* for multiobjective optimization, denoted as MKKT conditions, to make a distinction with the single-objective case.

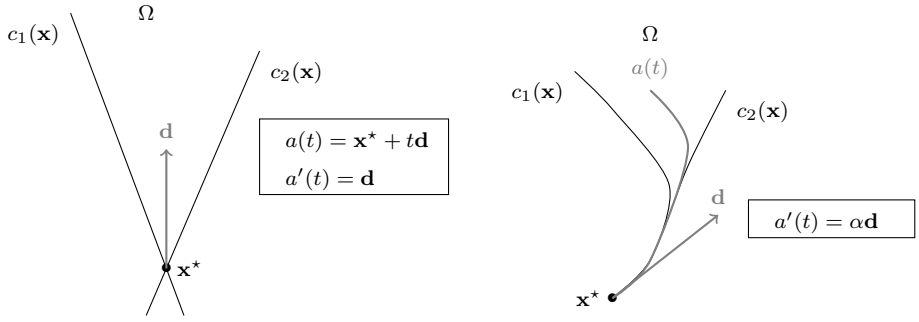
Theorem 2 (MKKT conditions)**.** *Let $\mathbf{x}^\star$ be a local Pareto optimal solution of (MOP). Suppose that the objective functions f_i for $i = 1, 2, \ldots, m$ and the inequality constraint functions c_j for $j \in \mathcal{J}$ are $\mathscr{C}^1$ at $\mathbf{x}^\star \in \Omega$, and that the Kuhn-Tucker constraint qualification holds at*

$\mathbf{x}^\star \in \Omega$. *Then there exists* $\lambda \geq 0$, $\lambda \in \mathbb{R}^m$ *and* $\mu \geq 0$, $\mu \in \mathbb{R}^{|\mathcal{J}|}$ *such that*

$$\begin{aligned}
&\sum_{i=1}^{m} \lambda_i \nabla f_i(\mathbf{x}^\star) + \sum_{j \in \mathcal{J}} \mu_j \nabla c_j(\mathbf{x}^\star) = 0, \\
&\mu_j c_j(\mathbf{x}^\star) = 0 \text{ for } j \in \mathcal{J}, \\
&(\lambda, \mu) \neq (0_{\mathbb{R}^m}, 0_{\mathbb{R}^{|\mathcal{J}|}}), \\
&\lambda \neq 0_{\mathbb{R}^m}.
\end{aligned}$$

When $m = 1$, the MKKT conditions reduce to the classical KKT conditions for constrained single-objective optimization (see [201, Theorem 12.1]).

Remark. Many general references on optimization (for example [135, 201]) rather use the *linear independence constraint qualification* (LICQ), which imposes that the set of gradients of active constraints at an optimal solution are linearly independent, to derive Karush-Kuhn-Tucker optimality necessary conditions. Note that Theorem 2 still holds if one substitutes the Kuhn-Tucker constraint qualification by the LICQ, as the LICQ implies that the Kuhn-Tucker constraint qualification holds (but it is not reciprocal). Figure 2.3 illustrates their relations.



(a) LICQ ⇒ KTCQ. (b) KTCQ is satisfied but not LICQ.

Figure 2.3 Two examples illustrating the relation between the LICQ and the Kuhn-Tucker constraint qualification (KTCQ)

If one removes the constraint qualification assumption, the following Fritz-John necessary conditions [190] still apply.

Theorem 3 (Fritz John necessary condition for Pareto optimality)**.** *Let* $\mathbf{x}^\star$ *be a locally Pareto optimal solution of (MOP) and suppose that the objective functions* f_i *for* $i = 1, 2, \ldots, m$ *and the inequality constraint functions* c_j *for* $j \in \mathcal{J}$ *are* $\mathscr{C}^1$ *at* $\mathbf{x}^\star \in \Omega$. *Then there exists* $\lambda \geq 0$,

$\lambda \in \mathbb{R}^m$ *and* $\mu \geq 0$, $\mu \in \mathbb{R}^{|\mathcal{J}|}$, *such that*

$$\begin{aligned}
&\sum_{i=1}^{m} \lambda_i \nabla f_i(\mathbf{x}^\star) + \sum_{j \in \mathcal{J}} \mu_j \nabla c_j(\mathbf{x}^\star) = 0, \\
&\mu_j c_j(\mathbf{x}^\star) = 0 \text{ for } j \in \mathcal{J}, \\
&(\lambda, \mu) \neq (0_{\mathbb{R}^m}, 0_{\mathbb{R}^{|\mathcal{J}|}}).
\end{aligned}$$

However, the guarantee that the multiplier λ associated to the objective function is strictly positive is lost, which may imply some degeneracy.

In DFO, the algebraic formulations of the objective functions are not explicitly available. In this context, the MKKT or KKT conditions cannot be deployed to design practical algorithms contrary to classical numerical optimization (for example, see [201] for a list of available constrained methods based on the exploitation of KKT equations). However, it is still possible to construct convergent-based DFO methods which rely on a more geometrical view of constraint conditions. Cones are useful tools in this context.

Definition 10 (Cone)**.** A set $\mathcal{C} \subseteq \mathbb{R}^n$ is a *cone* if and only for all $\mathbf{d} \in \mathcal{C}$ and every positive scalar $\alpha > 0$, $\alpha \mathbf{d} \in \mathbb{R}^n$.

For example, $\mathbb{R}^n$ and the singleton set $\{0_{\mathbb{R}^n}\}$ are cones. Cones can be convex, closed or open. The description below follows [201]. The notion of *(Bouligand) tangent cone* is important.

Definition 11 (Contingent cone or (Bouligand) tangent cone)**.** A vector $\mathbf{v} \in \mathbb{R}^n$ is said to be a tangent vector to Ω at the point $\mathbf{x} \in \mathbb{R}^n$ in the closure of Ω (not necessary, as the border is comprised in this set, but taken from [15]) if there exists a feasible sequence $\{\mathbf{v}^k\}$ that converges to $\mathbf{x}$ and a sequence of positive scalar $\{t^k\}$ such that $\mathbf{v} = \lim_{k \to +\infty} t^k(\mathbf{v}^k - \mathbf{x})$.

The set $\mathcal{T}_\Omega^{Co}(\mathbf{x})$ of all tangent vectors to Ω at $\mathbf{x}$ is called the *contingent cone* or the *(Bouligand) tangent cone* to Ω at $\mathbf{x}$.

An illustration of a tangent cone is given in Figure 2.4.

The (Bouligand) tangent cone represents all the directions in which the decision vector $\mathbf{x}$ can be perturbed without violating the constraints. The following theorem for first-order constrained conditions can then be given [201].

Theorem 4. *Let* $\mathbf{x}^\star \in \Omega$ *be locally Pareto optimal, and* $\mathcal{T}_\Omega^{Co}(\mathbf{x}^\star)$ *the tangent cone to* Ω *at* $\mathbf{x}^\star$*. Suppose that* f_i *for* $i = 1, 2, \ldots, m$ *and* c_j *for* $j \in \mathcal{J}$ *are* $\mathscr{C}^1$ *at* $\mathbf{x}^\star$*. Then for all* $\mathbf{d} \in \mathcal{T}_\Omega^{Co}(\mathbf{x}^\star)$*, there exists at least an objective index* $i(\mathbf{d}) \in \{1, 2, \ldots, m\}$ *such that* $\nabla f_{i(\mathbf{d})}(\mathbf{x}^\star) \geq 0$*.*

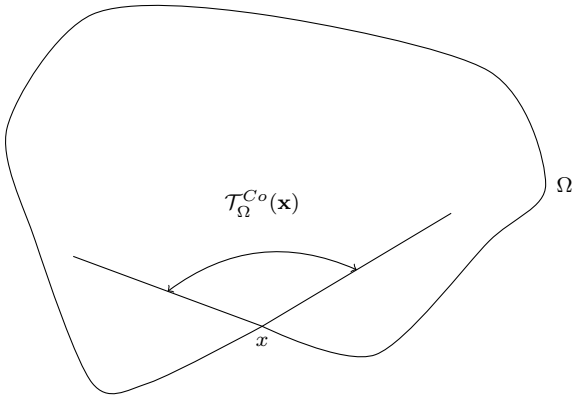


Figure 2.4 Tangent cone of $\mathbf{x} \in \Omega$ of a nonconvex set Ω.

Nonsmooth optimization

In general, nonsmooth optimization focuses on the analysis of optimization problems where objective and constraint functions are at least locally Lipschitz continuous at optimality. Many applications cannot be classified into the category of continuously differentiable problems but belong to the larger set of locally Lipschitz continuous problems. It is then reasonable to consider them in the context of BBO and DFO.

This subsection provides a summary of notions from nonsmooth optimization used along this thesis. For more details, the reader is invited to read the works of Clarke [59] or the most recent reference [35].

In nonsmooth optimization, one cannot use classical first order derivatives to assess optimality conditions. The *Clarke generalized directional derivative* [59] generalizes continuous directional derivatives to locally Lispchitz continuous functions.

Definition 12. Let $f : \mathbb{R}^n \to \mathbb{R}$ be a locally Lipschitz continuous function near $\mathbf{x} \in \mathbb{R}^n$. Then the *generalized Clarke directional derivative* of f at $\mathbf{x}$ in the direction $\mathbf{d} \in \mathbb{R}^n$ exists and is defined as

$$f^o(\mathbf{x}; \mathbf{d}) = \limsup_{\substack{\mathbf{y} \to \mathbf{x} \\ t \searrow 0}} \frac{f(y + t\mathbf{d}) - f(\mathbf{y})}{t}.$$

To illustrate this definition, the classical following example is considered [20, 59]. $f : x \to |x|$ is not differentiable at $x = 0$. On the contrary, $\forall d \in \mathbb{R}$,

$$|d| = \lim_{t \searrow 0} \frac{|td| - 0}{t} \leq \limsup_{\substack{y \to 0 \\ t \searrow 0}} \frac{|y + td| - |y|}{t} = f^o(0; d) \leq \limsup_{\substack{y \to 0 \\ t \searrow 0}} \frac{|y| + |td| - |y|}{t} = |d|,$$

hence $f^o(0; d) = |d|$ for all $d \in \mathbb{R}$.

In unconstrained nonsmooth optimization, first-order conditions can then be given using generalized Clarke derivatives.

Theorem 5 (First order conditions in unconstrained nonsmooth multiobjective optimization (adapted from [74]))**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be locally Lipschitz continuous at $\mathbf{x}^\star$. If $\mathbf{x}^\star$ is a local Pareto optimal solution of $\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x})$, then for all $\mathbf{d} \in \mathbb{R}^n$, there exists an objective index $i(\mathbf{d}) \in \{1, 2, \ldots, m\}$ such that $f^o_{i(\mathbf{d})}(\mathbf{x}^\star; \mathbf{d}) \geq 0$.*

If $m = 1$, one obtains the following first order conditions in unconstrained nonsmooth optimization:

$$f^o(\mathbf{x}^\star; \mathbf{d}) \geq 0 \text{ for all } \mathbf{d} \in \mathbb{R}^n.$$

In presence of some constraints, Theorem 5 needs some adjustments. Unfortunately, in a blackbox context, one cannot derive some results using the tangent cone from nonsmooth assumptions, as the construction of the tangent cone requires the constraint functions to be smooth. The notion of tangency needs to be generalized by introducing the following definition, taken from [59].

Definition 13 (Hypertangent cone)**.** A vector $\mathbf{d} \in \mathbb{R}^n$ is said to be a *hypertangent vector* to the set $\Omega \subset \mathbb{R}^n \neq \emptyset$ at $\mathbf{x} \in \Omega$ if there exists a scalar $\epsilon > 0$ such that

$$\mathbf{y} + t\mathbf{w} \in \Omega \text{ for all } y \in \Omega \cap \mathcal{B}_\epsilon(\mathbf{x}),\ \mathbf{w} \in \mathcal{B}_\epsilon(\mathbf{d}) \text{ and } 0 < t < \epsilon$$

where $\mathcal{B}_\epsilon(\mathbf{x})$ is the open ball of radius $\epsilon > 0$ centred in $\mathbf{x}$.

The set of all hypertangent vectors to Ω at $\mathbf{x}$ is called the *hypertangent cone* to Ω at $\mathbf{x}$ and is denoted by $\mathcal{T}^H_\Omega(\mathbf{x})$.

The hypertangent cone is the interior of the Clarke tangent cone [59] defined below. Reciprocally, the Clarke tangent cone can be considered as the closure of the hypertangent cone.

Definition 14 (Clarke tangent cone)**.** A vector $\mathbf{v} \in \mathbb{R}^n$ is said to be a *Clarke tangent vector* to the set $\Omega \subseteq \mathbb{R}^n \neq \emptyset$ at the point $\mathbf{x}$ in the closure of Ω if for every sequence $\{\mathbf{x}^k\}$ of elements of Ω that converges to $\mathbf{x}$ and for every sequence of positive real numbers $\{t^k\}$ which converge to 0, there exists a sequence of vectors $\{\mathbf{v}^k\}$ converging to $\mathbf{v}$ such that $\mathbf{x}^k + t^k\mathbf{v}^k \in \Omega$.

The set of all Clarke tangent vectors to Ω at $\mathbf{x}$ is called the *Clarke tangent cone* to Ω at $\mathbf{x}$ denoted as $\mathcal{T}^{Cl}_\Omega(\mathbf{x})$.

Contrary to the unconstrained nonsmooth case, one needs to slightly adapt the definition of the Clarke generalized directional derivative to the constrained case (to be sure the evaluation is restricted to points in the feasible domain), denoted as the Clarke-Jahn generalized directional derivative [149].

Definition 15. Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be locally Lipschitz continuous near $\mathbf{x} \in \Omega$, where $\Omega \subseteq \mathbb{R}^n \neq \emptyset$. Then for $\mathbf{d} \in \mathcal{T}_\Omega^H(\mathbf{x}) \neq \emptyset$, the *Clarke-Jahn generalized directional derivative* at $\mathbf{x} \in \Omega$ in the direction $\mathbf{d} \in \mathcal{T}_\Omega^H(\mathbf{x})$ exists and is defined by

$$f^o(\mathbf{x};\mathbf{d}) = \limsup_{\substack{\mathbf{y}\to\mathbf{x},\ \mathbf{y}\in\Omega \\ t\searrow 0,\ \mathbf{y}+t\mathbf{v}\in\Omega}} \frac{f(\mathbf{y}+t\mathbf{d}) - f(\mathbf{y})}{t}.$$

One can compute the Clarke-Jahn generalized directional derivatives at x in the direction $\mathbf{v} \in \mathcal{T}_\Omega^{Cl}(\mathbf{v})$ by passing at the limit, i.e. $f^o(\mathbf{x};\mathbf{v}) = \lim_{\mathbf{d}\to\mathbf{v},\mathbf{d}\in\mathcal{T}_\Omega^H(\mathbf{x})} f^o(\mathbf{x};\mathbf{d})$ (see [15, Proposition 3.9] for a proof). Finally, the first-order constrained condition for nonsmooth MOO theorem can be given.

Theorem 6 (First order conditions in constrained nonsmooth multiobjective optimization (taken from [74]))**.** *Let $f : \mathbb{R}^n \to (\mathbb{R} \cup \{+\infty\})^m$ be locally Lipschitz continuous near a point $\mathbf{x}^\star \in \Omega \subseteq \mathbb{R}^n \neq \emptyset$. $\mathbf{x}^\star$ is said to be Pareto-Clarke optimal if for all directions $\mathbf{d} \in \mathcal{T}_\Omega^{Cl}(\mathbf{x})$, there exists an index $i(\mathbf{d}) \in \{1, 2, \ldots, m\}$ such that $f^o_{i(\mathbf{d})}(\mathbf{x};\mathbf{d}) \geq 0$.*

Descent directions

The concept of a *descent direction* is central in numerical optimization.

Definition 16 (Descent direction for single-objective optimization [20])**.** A direction $\mathbf{d} \in \mathbb{R}^n$ is said to be a *descent direction* of the function $f : \mathbb{R}^n \to \mathbb{R}$ at $\mathbf{x} \in \mathbb{R}^n$ if and only if there exists a scalar $\bar{t} > 0$ such that $f(\mathbf{x} + t\mathbf{d}) < f(\mathbf{x}), \forall t \in (0, \bar{t}\,]$.

When $\mathbf{d} \in \mathbb{R}^n$ is a descent direction of a scalar-valued function f at $\mathbf{x} \in \mathbb{R}^n$, evaluating $f(\mathbf{x} + t\mathbf{d})$ for small values of t can lead to a better solution. The choice of relevant descent directions in SOO depends on the properties of the objective function and the constraints. Some well-known descent directions (a more detailed review can be found for example in [201, Chapter 2]) are reminded:

- The *steepest descent direction* for SOO : $\mathbf{d} = -\nabla f(\mathbf{x})$, defined when f is $\mathscr{C}^1$ at $\mathbf{x} \in \mathbb{R}^n$.
- The *Newton direction* for SOO: $\mathbf{d} = -(\nabla^2 f(\mathbf{x}))^{-1}\nabla f(\mathbf{x})$, assuming that f is $\mathscr{C}^2$ and $\nabla^2 f(\mathbf{x})$ is positive definite at $\mathbf{x} \in \mathbb{R}^n$.

- The *Quasi-Newton* direction for SOO: $\mathbf{d} = -\mathbf{H}^{-1}\nabla f(\mathbf{x})$, where $\mathbf{H}$ is a symmetric positive definite approximation of the Hessian of f at $\mathbf{x} \in \mathbb{R}^n$, assuming f is $\mathscr{C}^2$.

Many single-objective DFO methods use approximations of these descent directions.

The concept of descent direction for MOO is more recent. To the best of our knowledge, the first extensions of classical descent directions from SOO to MOO date from 2000 and are described in [117]. Since then, an important number of "descent-based" algorithms have been proposed [9,116,117,193,194,209–211,236]. Although DFO methods do not have access to derivatives, rendering the use of well-known descent directions impracticable, some of them rely on approximations (e.g. [53, 126]), which are used to guide the search. It is then interesting to mention them.

Definition 17. (Descent direction for multi-objective optimization (adapted from [116])). A direction $\mathbf{d} \in \mathbb{R}^n$ is said to be a *descent direction* of the function $f : \mathbb{R}^n \to \mathbb{R}^m$ at $\mathbf{x} \in \mathbb{R}^n$ if and only if there exists a scalar $\bar{t} > 0$ such that $f_i(\mathbf{x} + t\mathbf{d}) < f_i(\mathbf{x}), \forall t \in (0, \bar{t}]$ for all $i = 1, 2, \ldots, m$.

Fliege and Svaiter [117] define the *steepest descent direction for multiobjective optimization* as follows. Given $f : \mathbb{R}^n \to \mathbb{R}^m$ continuously differentiable at $\mathbf{x} \in \mathbb{R}^n$, the steepest descent direction $\mathbf{d}$ of f at $\mathbf{x}$ is defined as the solution of the following optimization problem

$$\min_{\mathbf{d}\in\mathbb{R}^n} \max_{i\in\{1,2,\ldots,m\}} \nabla f_i(\mathbf{x})^\top \mathbf{d} + \frac{1}{2}\|\mathbf{d}\|^2.$$

Note that this problem can be reformulated as a quadratic convex program with linear constraints:

$$\begin{aligned} \min_{(\alpha,\mathbf{d})\in\mathbb{R}\times\mathbb{R}^n} \quad & \alpha + \frac{1}{2}\|\mathbf{d}\|^2 \\ \text{s.t} \quad & \nabla f_i(\mathbf{x})^\top \mathbf{d} \leq \alpha, \quad i = 1, 2, \ldots, m \end{aligned}$$

which has a unique solution. The optimal value of this problem can also be used as a stopping criterion. When $m = 1$, the solution reduces to $\mathbf{d} = -\nabla f(\mathbf{x})$.

Similarly, Fliege et al [116] define the *Newton descent direction for multiobjective optimization* as follows. Given $f : \mathbb{R}^n \to \mathbb{R}^m$ twice continuously differentiable at $\mathbf{x} \in \mathbb{R}^n$, the Newton direction of f at $\mathbf{x}$ is defined as the solution of the following subproblem:

$$\min_{\mathbf{d}\in\mathbb{R}^n} \max_{i\in\{1,2,\ldots,m\}} \nabla f_i(\mathbf{x})^\top \mathbf{d} + \frac{1}{2}\mathbf{d}^\top \nabla^2 f_i(\mathbf{x})\mathbf{d},$$

assuming $\nabla^2 f_i(\mathbf{x})$ is symmetric positive definite for all $i = 1, 2, \ldots, m$.

A Quasi-Newton descent direction for MOO is similarly defined [193,210] by replacing $\nabla^2 f_i(\mathbf{x})$ in the previous problem by $\mathbf{B}^i$, where $\mathbf{B}^i$ is a symmetric positive definite approximation of the i-th objective function Hessian.

2.2 Derivative-free optimization methods for single-objective optimization

Many blackbox and derivative-free MOO methods are extensions of single-objective DFO algorithms. To understand their design, it is thus important to know their relative single-objective version. This section aims to provide a short summary of the principal ones used in this work. In this part, f is a scalar-valued function. The reader is invited to refer to the following books [20, 67] and the extensive recent surveys [21, 160, 164] for more information.

Deterministic DFO methods can be broadly classified into two categories [20, 67]. The first are *direct search methods*, which attempt to find solutions by only evaluating and comparing points, without any other information. The second are *Model-based methods*. They encompass, among others, linesearch and trust-region procedures, build an approximation of the objective and constraints functions, the so-called model. They use the model to guide future iterations. Model-based algorithms are very effective when the functions are smooth (even if gradients are not analytically available), whereas the choice of direct search methods is more pertinent when dealing with nonsmooth functions. This classification is nevertheless more and more approximative. Indeed, new hybrid methods (see [66, 70, 75, 182] for example) have emerged for these last years, which combine properties of these two groups to overcome their limitations and improve their efficiency. Nonetheless, for simplicity, this work will conserve the classification into the two categories.

2.2.1 Direct search methods

The use of direct search methods in engineering applications can be traced to the works of Fermi and Metropolis [113] on the Coordinate Search (CS) algorithm. But it is Hooke and Jeeves [143] who use the term of "direct search" methods for the first time. According to Kolda, Lewis and Torczon in their monumental review on this topic [160], direct search algorithms possess the two following features:

- They do not make any use of gradient-based information, contrary to model-based DFO methods.
- They are iterative procedures. They work from a point, the so-called best solution found until the previous iteration, and examine a given and finite set of decision vectors

in its "neighbourhood" (this notion is not mathematical here, and the definition of a neighbourhood in a direct search context depends on the choice of the method). At the end of the iteration, if a better candidate is found, it is then designated as the best new solution point. Otherwise, a parameter linked to the size of the neighbourhood is decreased, resulting in a more refined search for the next iteration.

The Mesh Adaptive Direct Search algorithm for constrained optimization

The Mesh Adaptive Direct Search (MADS) algorithm [15] is a generalization of Generalized Pattern Search (GPS) methods [13, 244], which themselves generalize pattern search algorithms (including the CS algorithm) to nonsmooth constrained optimization. The treatment of these methods in this section is different from the traditional approach adopted in the main references on DFO [20, 67]. Usually, an introduction to these methods follows a historical perspective: it begins with the CS algorithm, then GPS, to conclude with MADS. Here, the description focuses more on MADS, as it is one of the most elaborate and efficient direct search methods. Particular cases such as CS and GPS are derived from it. The presentation is inspired from [20]. Before starting, one needs the following definitions [81].

Definition 18 (Positive spanning set)**.** A set of finite vectors $\mathbb{D}$ in $\mathbb{R}^n$ is said to be a *positive spanning set* for $\mathbb{R}^n$ if the set of all positive combinations of $\mathbb{D}$ spans $\mathbb{R}^n$, i.e.,

$$\text{pspan}(\mathbb{D}) = \left\{ \sum_{j=1}^{|\mathbb{D}|} \lambda_j \mathbf{d}^j, \lambda_j \geq 0, \mathbf{d}^j \in \mathbb{D} \right\} = \mathbb{R}^n.$$

An important property of positive spanning sets is that every open half-space of $\mathbb{R}^n$ contains at least one of their elements [67, Theorem 2.3]. In the unconstrained differentiable case, it implies that for a given point such that $\nabla f(\mathbf{x}) \neq 0$, there exists an element of the positive spanning set $\mathbf{d} \in \mathbb{D}$ such that $\mathbf{d}$ is a descent direction of f at $\mathbf{x} \in \mathbb{R}^n$ [67, 160]. Intuitively, exploring around directions of a positive spanning set can lead to a decrease of the value of f. It is then interesting to look for positive spanning sets of minimum size that span $\mathbb{R}^n$.

Definition 19 (Positive basis)**.** A *positive basis* is a minimum positive spanning set in the sense of inclusion, i.e., all strictly subsets of this set do not positively span $\mathbb{R}^n$.

MADS [15] is a direct search algorithm designed to solve single-objective BBO constrained problems, i.e.,

$$\min_{\mathbf{x} \in \Omega} f(\mathbf{x})$$

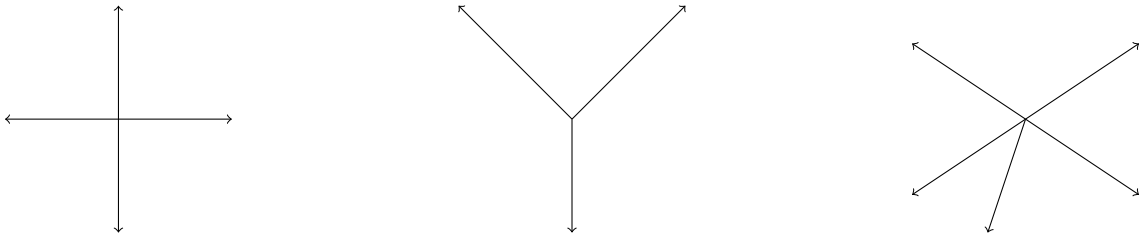


Figure 2.5 Examples of positive spanning sets in $\mathbb{R}^2$ (inspired by [20, 67]): on the left, a positive basis with $2n = 4$ directions; in the center, a minimal positive basis with $n + 1 = 3$ directions; on the right, a positive spanning set which is not a positive basis with 5 directions.

where $\Omega \subseteq \mathbb{R}^n \neq \emptyset$, $int(\Omega) \neq \emptyset$ and $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$. Allowing the objective function f to take infinity values means the evaluations of f at a given decision vector can fail.

At each iteration k, MADS attempts to find better points in the decision space, belonging to the current mesh M^k.

Definition 20 (Mesh [15])**.** Let $\mathbf{G} \in \mathbb{R}^{n \times n}$ be a non-singular matrix and $\mathbf{Z} \in \mathbb{Z}^{n \times n_D}$ be such that the columns of $\mathbf{Z}$ form a positive spanning set for $\mathbb{R}^n$. Define $\mathbf{D} = \mathbf{GZ}$. At iteration k, the current *mesh of coarseness* $\delta^k > 0$, generated by $\mathbf{D}$ is defined by

$$M^k = \bigcup_{\mathbf{x} \in V^k} \left\{ \mathbf{x} + \delta^k \mathbf{D}\mathbf{z} : \mathbf{z} \in \mathbb{N}^{n_D} \right\} \subset \mathbb{R}^n$$

where V^k is the set of points already evaluated by the start of iteration k.

The mesh is roughly speaking a discretization of the decision space parametrized by its *mesh size parameter* $\delta^k > 0$ and generated by a positive spanning matrix $\mathbf{D}$ (as $\mathbf{G}$ is non singular). Classical choices for $\mathbf{G}$ and $\mathbf{Z}$ are $\mathbf{Z} = [-\mathbf{I}_n \ \mathbf{I}_n]$ and $\mathbf{G} = \mathbf{I}_n$, where $\mathbf{I}_n$ is the identity matrix of size n, so $\mathbf{D} = [-\mathbf{I}_n \ \mathbf{I}_n]$. With this configuration, projecting points on the mesh is easier, but other choices are possible [67]. When the evaluations are expensive, it is common to save evaluated points in a cache V^k to avoid unnecessary evaluations.

Each iteration is organized around two steps: a *search* and a *poll*. The search is an optional step, as it does not play any role in the convergence analysis. During the search, the user can use any technique to generate trial points, as long as they lie on the mesh M^k and their number is finite. The poll is less flexible, as the convergence analysis depends on it. The poll consists in a local exploration around the best feasible decision vector found until iteration k denoted as $\mathbf{x}_F^k \in \Omega$. More precisely, the poll candidates must belong to a subset of the mesh called a (MADS) frame [15].

Definition 21 (Frame)**.** At iteration k, the frame is defined to be the set

$$P^k = \left\{ \mathbf{x}_F^k + \delta^k \mathbf{d} : \mathbf{d} \in \mathbb{D}_\Delta^k \right\}$$

where $\mathbb{D}_\Delta^k$ is a positive spanning set such that $0 \notin \mathbb{D}_\Delta^k$ and for each $d \in \mathbb{D}_\Delta^k$,

- $\mathbf{d}$ can be written as a nonnegative integer combination of the directions in $\mathbb{D}$, and $\mathbf{d} = \mathbf{D}\mathbf{u}$ for some vector $\mathbf{u} \in \mathbb{N}^{n_D}$ that may depend on the iteration number k;
- the distance from the frame center $\mathbf{x}_F^k$ to a frame point $\mathbf{x}_F^k + \delta^k \mathbf{d} \in P^k$ is bounded above by a constant times the frame size parameter Δ^k: $\delta^k \|\mathbf{d}\| \leq \Delta^k \max\{\mathbf{d}' : \mathbf{d}' \in \mathbf{D}\}$, where $0 < \delta^k \leq \Delta^k$ and

$$\lim_{k \in K} \delta^k = 0 \text{ if and only if } \lim_{k \in K} \Delta^k = 0 \text{ for every infinite subset of indices } K.$$

To satisfy the last point of the definition above, implementations of MADS set the following relation between the mesh size and the frame size parameter: $\delta^k = \min\{\Delta^k, (\Delta^k)^2\} \Leftrightarrow \Delta^k = \max\{\sqrt{\delta^k}, \delta^k\}$ [20]. As the frame size parameter decreases, the mesh size parameter decreases faster. The mesh refines, and the number of potential candidates increases. Examples of frames can be found on Figure 2.6. To take into account the scale of variables, it is common to associate to each variable its own frame and mesh size parameter [18, 26, 167]. MADS can use different strategies to generate polling directions [2, 15, 18, 22, 26]. One of the most efficient is the OrthoMADS strategy, proposed in [2].

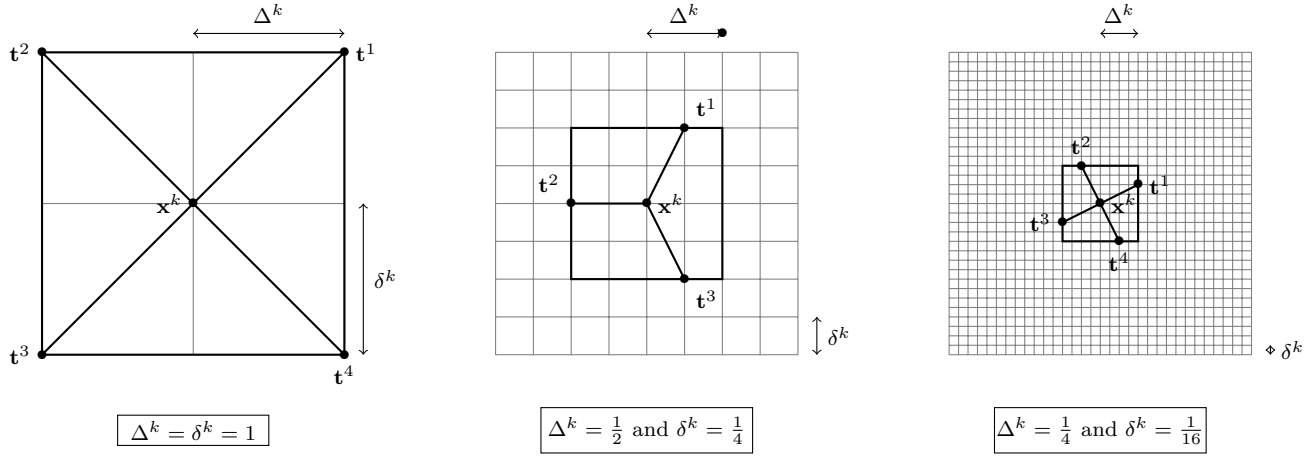


Figure 2.6 Examples of frames and meshes in $\mathbb{R}^2$ (inspired by [20]).

A poll can be opportunistic (as soon as the algorithm finds a better candidate, the poll is

interrupted) or complete (all candidates are evaluated), as it does not affect the convergence analysis.

Once the poll and the search are complete, the mesh is updated. If the algorithm does not find a better incumbent during iteration k, the mesh size and frame size parameters are reduced, increasing the mesh resolution to evaluate at the next iteration trial points closer to the best incumbent $\mathbf{x}_F^k$. Otherwise, the current incumbent is updated, and the mesh and frame size parameters are increased. Care must be taken on the strategies of updating the frame and the mesh for the convergence (for example, by using the same mesh size adjustment parameter τ for the increasing or coarsening of the mesh).

Algorithm 5 summarizes the main steps of MADS.

Algorithm 1 The mesh adaptive direct search algorithm (MADS) (inspired by [20])

Given $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$, $\mathbf{x}_F^0 \in \Omega$ a starting point, choose $\Delta^0 > 0$ the initial frame size parameter, $\tau \in \mathbb{Q} \cap (0; 1)$ the mesh size adjustment parameter, and $\mathbf{D} = \mathbf{GZ}$ a positive spanning set matrix.
for $k = 0, 1, 2, \ldots$ **do**
Set the mesh size parameter at $\delta^k = \min\left\{\Delta^k, \left(\Delta^k\right)^2\right\}$.
1. **Search step** (optional): Evaluate f at a finite set of points $S^k \subset M^k$. If successful, go to 3.
2. **Poll step**: Select a positive spanning set $\mathbb{D}_\Delta^k \subset \mathbf{D}$. Evaluate f at the set of poll points $P^k = \{\mathbf{x}_F^k + \delta^k \mathbf{d} : \mathbf{d} \in \mathbb{D}_\Delta^k\} \subset F^k$ where F^k is the frame of extent Δ^k.
3. **Parameter update**:
if the iteration is successful (there exists $\mathbf{t}^s \in S^k$ or $\mathbf{t}^p \in P^k$ such that $f(\mathbf{t}^s) < f(\mathbf{x}_F^k)$ or $f(\mathbf{t}^p) < f(\mathbf{x}_F^k)$) **then**
set $\mathbf{x}_F^{k+1} := \mathbf{t}^s$ or $\mathbf{t}^p$ and $\Delta^{k+1} := \tau^{-1}\Delta^k$.
else
set $\mathbf{x}_F^{k+1} := \mathbf{x}_F^k$ and $\Delta^{k+1} := \tau\Delta^k$.
end if
end for

MADS stops when one of the following stopping criteria is satisfied: either a maximal number of blackbox evaluations is reached or the frame/mesh size parameter is below a threshold value provided by the user.

Under mild assumptions [1], the MADS convergence analysis [15] guarantees the existence of an accumulation point $\hat{\mathbf{x}}_F \in \Omega$ such that its Clarke generalized derivative satisfies $f^o(\hat{\mathbf{x}}_F; \mathbf{d}) \geq 0$ for all directions $\mathbf{d} \in \mathbb{R}^n$ belonging to the Clarke tangent cone $\mathcal{T}_\Omega^{Cl}(\hat{\mathbf{x}}_F)$[2]. This strong

[1] There exists a feasible starting point, its objective value is finite and all iterates x_F^k lie in a compact set; a work that relaxes the first assumption can be found at [16].

[2] If $int(\Omega) \neq \emptyset$ and $\hat{\mathbf{x}}_F \in int(\Omega)$, then $\mathcal{T}_\Omega^H(\hat{\mathbf{x}}_F) \neq \emptyset$, and consequently, $\mathcal{T}_\Omega^{Cl}(\hat{\mathbf{x}}_F) \neq \emptyset$.

convergence result holds only if MADS generates poll directions which are asymptotically dense in the unit sphere. Some polling strategies which satisfy this criterion can be found at [2, 15,18,22,26]. When the objective function is strictly differentiable at the limit point $\hat{\mathbf{x}}_F \in \Omega$, this solution satisfies first-order stationarity conditions [15]. Second-order convergence results can be found in [1]. Convergence analysis in the discontinuous case is also derived in [253].

As it is mentioned at the beginning of this subsection, the MADS framework generalizes many other direct search methods. To obtain the Generalized Pattern Search (GPS) algorithm [13, 244], one needs to remove the asymptotically dense direction property. More precisely, during iteration k, all candidates must still be evaluated on a mesh M^k. Poll candidates must belong to the set $P^k = \{\mathbf{x}_F^k + \delta^k \mathbf{d} : \mathbf{d} \in \mathbb{D}_\Delta^k\} \subset M^k$, where $\mathbb{D}_\Delta^k$ is a positive spanning set. However, the set of directions $\bigcup_{k\in\mathbb{N}} D^k$ along all iterations is finite, contrary to MADS. The frame size parameter is no more required. Under the same assumptions as for MADS, one can show GPS generates an accumulation point $\hat{\mathbf{x}}_F \in \Omega$ such that its Clarke generalized derivative satisfies $f^o(\hat{\mathbf{x}}_F; \mathbf{d}) \geq 0$ for all $\mathbf{d} \in \mathbb{D}$, where $\mathbb{D}$ is a positive basis (one of the positive basis selected for an infinity of iterations when building the poll set).

The Coordinate Search (CS) algorithm for unconstrained optimization uses the positive basis $[\mathbf{I}_n \ \ -\mathbf{I}_n]$ to build its poll directions. It does not possess a search step. The step size parameter δ^k is kept constant when the iteration is successful and divided by 2 when the iteration is a failure (i.e., $\tau = \frac{1}{2}$). Assuming f possesses bounded level sets and is continuously differentiable, one can show that CS generates a sequence of iterates which converges to a stationary point $\hat{\mathbf{x}} \in \mathbb{R}^n$ for unconstrained optimization, i.e., $\nabla f(\hat{\mathbf{x}}) = 0$ (see for example [20, Chapter 3]). These results are similar as the ones of GPS.

Since its conception in 2006 [15], MADS has benefited from important theoretical and practical improvements. A filter approach to take into account blackbox constraints based on a progressive barrier is detailed in [16]. Other poll strategies can be found in [22, 248]. Implementations of efficient search steps are also explored in [29, 66]. Theoretical results of MADS with stochastic/noisy blackboxes are found in [19].

Remark. The direct search Generating Search Set (GSS) algorithm framework [160] is a direct-search method with the same global convergence properties than the MADS algorithm, i.e., it guarantees the existence of a Clarke-optimal accumulation point $\hat{\mathbf{x}}$ under mild assumptions. This algorithm removes the restriction that all iterates should lie on the mesh M^k (designed too in the literature as integer lattices [67]). Nonetheless, it imposes a sufficient decrease condition on the choice of a new incumbent point, i.e., $\mathbf{x}^{k+1} \neq \mathbf{x}^k$ is accepted as a new incumbent if

$$f(\mathbf{x}^{k+1}) < f(\mathbf{x}^k) - \rho(\delta^k)$$

where the forcing function $\rho : \mathbb{R}_+ \to \mathbb{R}_+$ is continuous, positive and satisfies

$$\lim_{t \searrow 0} \frac{\rho(t)}{t} = 0 \text{ and } \rho(t_1) \leq \rho(t_2) \text{ if } t_1 < t_2.$$

Common examples of forcing functions are $\rho : t \in \mathbb{R}_+ \mapsto t^{1+a}$ for some scalar $a > 0$.

From a practical point of view, the MADS framework exploits all evaluations. Indeed, some points for the GSS framework can be discarded, as they do not satisfy the sufficient decrease condition. For an engineer, if a blackbox is costly, it can be difficult to accept that a new candidate is not used even if it possesses a better objective value. Adjusting the mesh to consider integer and binary variables is also straightforward [18]. On the other hand, many derivative-free algorithms with equality constraints use GSS, as it is easier to use projections, or adapt the generation of directions without the mesh (see for example [54,160,162,170,171]) to deal with these constraints. To the best of our knowledge, complexity results have also been explored only within the GSS framework [124, 131, 132, 252].

The Implicit Filtering algorithm

The Implicit Filtering (IF) algorithm, proposed by Gilmore and Kelley [129, 154], can be seen as a hybrid between a pattern search algorithm and a Newton-based linesearch method. IF targets derivative-free bound-constrained problems with continuously differentiable functions. From the rest of this subsection, the feasible decision set will be given as $\Omega = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}\}$ with $\mathbf{l}, \mathbf{u} \in \mathbb{R}^n$ such that $\mathbf{l}_i < \mathbf{u}_i$ for $i = 1, 2, \ldots, n$. IF is initially given a starting point $\mathbf{x}^0 \in \Omega$, a stepsize δ^0 and a fixed decrease parameter $\tau \in (0; 1)$.

At each iteration k, IF starts from the current solution $\mathbf{x}^k$, with its associated step size $\delta^k > 0$. IF first builds a *stencil* by evaluating the following points of this set:

$$\{\mathbf{x}^k + \delta^k \mathbf{d} : \mathbf{d} \in \mathbf{D}\}$$

where $\mathbf{D} = [\mathbf{I}_n \ \ -\mathbf{I}_n]$ is the set of stencil directions, $\mathbf{I}_n$ the identity matrix of size $n \times n$. As for GSS, the stepsize is decreased when the stencil step fails (i.e., no better current solution is found). Note that contrary to GSS or MADS methods, the stencil is not evaluated opportunistically.

When there is no stencil failure, i.e., there exists a descent direction, a linesearch is then conducted. IF constructs a central difference-based approximated gradient

$$\nabla_\delta f(\mathbf{x}) = \left(\frac{f(\mathbf{x} + \delta \mathbf{e}_i) - f(\mathbf{x} - \delta \mathbf{e}_i)}{2\delta} \right)_{1 \leq i \leq n}$$

(it can be adapted to take into account the bounds, see [126] for an illustration) where $\mathbf{e}_i$ is the i-th coordinate vector. An approximated Hessian (or identity matrix) $\mathbf{H}^k$ is provided to compute a descent direction, given by:

$$\mathbf{d}^k = -\left(\mathbf{H}^k\right)^{-1} \nabla_{\delta^k} f(\mathbf{x}^k)$$

if the norm of the approximated gradient is not below δ^k. Then IF performs a linesearch along $\mathbf{d}^k$ using the following Armijo backtracking procedure: find the least integer $l \in \{0, 1, \ldots, l_{\max}\}$ such that

$$f(\mathbf{x}^k + \alpha^l \mathbf{d}^k) - f(\mathbf{x}^k) \leq \eta\ \alpha^l\ \nabla_{\delta^k} f(\mathbf{x}^k)^\top \mathbf{d}^k$$

where $l_{\max} \in \mathbb{N}$, $\eta, \alpha \in (0; 1)$ are user-specified parameters.

At the end of the iteration, IF updates the current solution (i.e. $\mathbf{x}^{k+1} := \mathbf{x}^k + \alpha^l \mathbf{d}^k$ if the linesearch is successful; $\mathbf{x}^{k+1} := \mathbf{x}^k$ otherwise), and if there is failure in both steps, decreases δ^k, i.e., $\delta^{k+1} := \tau \delta^k$.

The algorithm stops when δ^k is below a user tolerance (not too low due to numerical issues) or the evaluation budget is exhausted.

Assuming f is $\mathscr{C}^1$, IF converges to a first-order stationary solution. This algorithm may not be a good choice when solving blackbox problems, where functions are not differentiable or evaluations may fail.

2.2.2 Linesearch-based methods

Linesearch-based DFO techniques, as their name implies, are inspired by gradient-based algorithms, which search for a better incumbent along a promising direction by solving a unidimensional minimization subproblem. One can trace them back to the following works [82, 134]. This subsection only discusses the Derivative-Free Nonsmooth (DFN) algorithm [109] which belongs to this group of methods. On the one hand, DFN addresses the resolution (without derivatives) of general nonsmooth (but Lipschitz continuous) constrained optimization problems. On the other hand, DFN is one of the principal inspirations for the conception of the Derivative-Free Multiobjective Optimization (DFMO) algorithm, described in Section 2.3.2. DFN may be thought as a variant of a GSS direct search method, which would exploit only one direction at each iteration.

DFN considers the following SOO problem:

$$\min_{\mathbf{x}\in\Omega\subset\mathbb{R}^n} f(\mathbf{x})$$

where $\Omega = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{l} \leq \mathbf{x} \leq \mathbf{u} \text{ and } c_j(\mathbf{x}) \leq 0 \text{ for } j \in \mathcal{J}\}$, with $\mathbf{l} < \mathbf{u}$. It also assumes that the objective function f and the inequality functions $c_j : \mathbb{R}^n \to \mathbb{R}$ for $j \in \mathcal{J}$ are Lipschitz continuous and the bound variables $\mathbf{l}$ and $\mathbf{u}$ are finite. The following penalized problem is then defined:

$$\min_{\mathbf{l}\leq\mathbf{x}\leq\mathbf{u}} Z_\epsilon(\mathbf{x}) = f(\mathbf{x}) + \frac{1}{\epsilon}\sum_{j\in\mathcal{J}} \max\{0, c_j(\mathbf{x})\}$$

where $\epsilon > 0$ is a penalty parameter fixed by the user.

At iteration k, DFN selects a direction $\mathbf{d}^k \in \mathbb{R}^n$ such that $\|\mathbf{d}^k\| = 1$, where $\{\mathbf{d}^k\}_{k\in\mathbb{N}}$ is a sequence predefined by the user. The choice of such directions can involve various strategies: coordinate search [109], OrthoMADS strategy [2, 109], and so on. DFN then performs a search along the two directions $\pm\mathbf{d}^k$. If a sufficient reduction of the penalized objective function is obtained, DFN tries to find a suitable steplength parameter for the next iteration via an extrapolation along the direction $\mathbf{d}^k$. Otherwise, if the two initial evaluations fail, the tentative steplength is decreased, which enables a finest exploration around the current best solution. Note that during the whole iteration, the algorithm systematically executes a projection such that all iterates remain in the bound set $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}\}$.

It can be shown that the algorithm generates an accumulation point which is stationary for the initial constrained problem assuming the penalty parameter is sufficiently small. Thus, the numerical performance of the algorithm may be linked to the choice of this external parameter. DFN also assumes that all involved functions are Lipschitz continuous. This assumption may be too restrictive in a BBO context. Extensions of this algorithm to mixed integer derivative-free problems can be found in [178,179,181]. Other variants [177,183] have been developed for the linear inequality case, with stronger assumptions (i.e. differentiability).

2.2.3 Model-based approaches

Even though the blackbox does not return derivatives and it is impracticable to compute them by finite differences, the underlying functions can still be smooth. Model-based approaches are mainly designed to address these problems. The principal idea behind these methods is to approximate the objective (and the constraints) function by a "good" model, which is used to guide the optimization process. This subsection summarizes the main points of these approaches. For more details, the reader is referred to [20, 21, 67].

The theoretical convergence and practical performance of model-based methods is strongly related to the way the model "is close" to the true blackbox. Formally, the characterization of a "good" model is expressed by its error bounds. The most common classes of "good" surrogates are *fully linear* and *fully quadratic* models [67]. Roughly speaking, a model is said to be fully linear on an open ball if the error on the function value scales at most quadratically, and if the error on the function gradient scales at most linearly with the radius of the open ball. Similarly, a model is said to be fully quadratic on an open ball if the error on the function value, gradient and hessian scale at most respectively cubically, quadratically and linearly with the radius of the open ball. Fully quadratic models are more precise if stronger assumptions on the blackbox are satisfied. *Order N function accuracy* and *order N subgradient accuracy* generalize fully linear and quadratic models to nonsmooth optimization [21]. They respectively impose an error bound on the function and its subgradient which scales at most N times with the radius of the open ball.

As highlighted in [21], this terminology enables to decouple the construction of models from the design of model-based algorithms. It is common for researchers to start from a derivative-based "basic" framework. The use of an accurate model enables to guarantee error bounds on the value, gradient or Hessian of the true function. The model can then be directly incorporated into the derivative-based framework, without changing the convergence analysis (multiplied by the complexity cost required to build the model). One can find an example of such an algorithm in [20, Chapter 11].

Thus, the conception of many model-based DFO algorithms has started from derivative-based methods and exploits their properties. For example, the integration of radial basis functions models into convergent-based algorithms has been proposed in [216, 261, 262]; Sampaio and Toint [224] developed a funnel DFO trust-region algorithm that takes into account smooth equality constraints; and Augustin and Marzouk [33] adapted a trust-region algorithm to treat DFO constrained problems. Note that many of these methods assume that the problem is smooth even if derivatives are not available. To the best of our knowledge, Liuzzi et al [182] are the first to propose a model-based approach for nonsmooth problems that does not exploit its composite structure. Their quadratic model possesses two characteristics which enables them to prove their algorithm convergence to a Clarke stationary point. It exploits dense directions in the unit sphere to approximate its linear part. The maximal eigenvalue of its Hessian is bounded by a negative power of the trust-region radius (which itself converges to 0). We invite the reader to refer to [21, 164] for a more thorough review.

All procedures mentioned above are local in the sense that they approximate the function in a local region of the decision space. Global surrogate approaches, which have gained a lot

of popularity for these last years, remove this property. The idea is to substitute the true function by a global surrogate, which is used to guide the search for better points. There exists various surrogates: Kriging (in the context of Bayesian optimization [153,191,230,268]), radial basis functions [196,197,212,264], Delaunay-based approaches [7] or other [156]. Many of these methods are heuristics or converge in a probabilistic sense (even though the true function is not noisy) [203, 214], which can lead to reliability issues, e.g., convergence to a non-stationary point.

Finally, model-based approaches can be combined with direct search methods to enhance practical performance. In [76], the authors propose to use simplex gradients to order polling directions. In [55,66,75], the authors incorporate an efficient search step based on linear and quadratic models. More precisely, Custódio et al [75] target GSS, whereas Burmen et al [55] and Conn and Le Digabel [66] tackle MADS. Burmen et al [55] exploit the generation of dense directions in the unit sphere to build an approximation of the Hessian. This technique requires only 2 evaluations, outperforming (at the time of the publication) the search step based on quadratic models proposed by [66]. The approximation of constraint functions by linear models [55] instead of quadratic surrogates as it is done in [66] can also explain this performance gap. Indeed, a quadratic problem with linear constraints is generally simpler to solve than a quadratically constrained quadratic problem. Integration of global surrogates in MADS is described in [238].

2.2.4 Metaheuristics

Metaheuristics are popular approaches to tackle BBO. Their functioning is often based on physical or biological analogies. In this category, one can find evolutionary algorithms [99], particle-swarm methods [207], or simulating annealing [157]. The reader is referred to the following references [127, 237] for a general overview. Although they are versatile and parallelizable, these methods often require an important number of functions evaluations to work, possess arbitrary stopping criteria and many algorithmic parameters to tune to make them efficient. However, some hybrid works which combine direct search methods (using sufficient decrease) and evolutionary algorithms have been proposed with convergence guarantees [95, 96].

2.3 Derivative-free optimization methods for multiobjective optimization

The methods presented in this section aim at generating a discrete representation of the Pareto front for a given MOO problem. Indeed, they are practically limited by the total

budget of evaluations imposed by the engineering context. In general, generating the whole Pareto front is then out of reach, since it is often composed of an infinite number of elements, or the cardinality of the Pareto front is too large to be captured with the provided budget of evaluations. These algorithms are classified into two families: *scalarization-based* approaches (or with *a prior articulation of preferences*), and methods *with a posteriori articulation of preferences* [74].

2.3.1 Scalarization-based approaches

The principle behind scalarization-based methods is to reformulate the initial MOO problem into a series of parameterized single-objective formulations. The solution of a subproblem by an adequate SOO solver is typically a local Pareto optimal decision vector. Different local Pareto solutions are obtained by changing the parameters of the formulation and resolving the new subproblem. Classical formulations are detailed below.

Weighted sum method. The *weighted sum method* [190] consists of converting the MOO problem into a SOO problem which minimizes a convex sum of objective functions:

$$\min_{\mathbf{x}\in\Omega} \sum_{i=1}^{m} w_i f_i(\mathbf{x})$$

where $w_i \geq 0$ for $i = 1, 2, \ldots, m$ are weights such that $\sum_{i=1}^{m} w_i = 1$.
Simple to implement and intuitive to understand, the weighted sum method cannot generate any point on the non-convex part of the Pareto front [80]. Furthermore, this formulation can be redundant, i.e., the same solutions can be reached for different combinations of weights.

Augmented weighted Tchebysheff method. The *augmented weighted Tchebysheff method* [260] aggregates objective functions into a SOO problem defined by

$$\min_{\mathbf{x}\in\Omega} \max_{1\leq i\leq m} \{w_i \left(f_i(\mathbf{x}) - \mathbf{r}_i\right)\} + \rho \sum_{i=1}^{m} (f_i(\mathbf{x}) - \mathbf{r}_i)$$

where $\mathbf{r} \in \mathbb{R}^m$ is a fixed reference point, $w_i \geq 0$ for $i = 1, 2, \ldots, m$ are weights such that $\sum_{i=1}^{m} w_i = 1$, and $\rho > 0$ is an external parameter sufficiently small.
Contrary to the weighted sum method, the augmented weighted Tchebyscheff method can generate points on the non-convex part of the Pareto front. The surrogate-based evolutionary algorithm ParEGO [158] for multiobjective BBO is based on this approach.

When $\rho = 0$, the resulting formulation is known as the *weighted Tchebysheff method*. The Multi-Objective Nonlinear Simplex Search (MONSS) algorithm [267] extends the Nelder-Mead algorithm [198] to MOO via a scalarization approach based on this formulation.

ε-constraint method. The *ε-constraint method* [138] selects one criterion as the objective function of the single-objective formulation and adds to the feasible set the other objectives as inequality constraints. The ε-constraint formulation is given as follows:

$$\begin{aligned} &\min f_l(\mathbf{x}) \\ &\text{subject to} \quad f_j(\mathbf{x}) \leq \boldsymbol{\varepsilon}_j, \quad j = 1, 2, \ldots, m : j \neq l \\ &\qquad\qquad \mathbf{x} \in \Omega \end{aligned}$$

where $\boldsymbol{\varepsilon} \in \mathbb{R}^{m-1}$ such that the problem is still feasible, and $l \in \{1, 2, \ldots, m\}$ the index of one objective function.

By varying the $\boldsymbol{\varepsilon}$ parameter, this method can generate points on the Pareto front. This method possesses several drawbacks. Firstly, it increases the size of the set of constraints, as it adds several of them to the original problem. The second difficulty is the choice of the $\boldsymbol{\varepsilon}$ parameter, which has to be carefully tuned so that the formulation remains feasible or numerically solvable once new constraints are added.

The constrained MOO surrogate-based blackbox p-ARGONAUT algorithm [39] uses the $\boldsymbol{\varepsilon}$-constraint method to reformulate the MOO problem into a single-objective optimization formulation. This problem is then solved by the ARGONAUT algorithm [43] for several values of $\boldsymbol{\varepsilon}$ to get different non-dominated points.

The reader is invited to consult the following survey about scalarization formulations [259]. For the rest of this section, the algorithms presented here will be treated with more detail, as they are sometimes used in the rest of this manuscript.

Direct search methods: BiMADS and MultiMADS

The BiMADS algorithm is a scalarization-based algorithm for constrained biobjective BBO proposed in [27]. It generates a Pareto front approximation by solving a succession of single-objective optimization problems using MADS [15]. More precisely, each formulation combines a reference point $\mathbf{r} \in \mathbb{R}^m$ and all criteria functions f_i for $i = 1, 2, \ldots, m$ into a scalar-valued function $\psi_r : \mathbf{x} \in \Omega \to \mathbb{R} \cup \{\infty\}$ which must satisfy the requirements presented in the following definition [27].

Definition 22. Consider the single-objective optimization problem:

$$R_r : \min_{\mathbf{x}\in\Omega} \psi_r(\mathbf{x}) \text{ with } \psi_r(\mathbf{x}) = \phi_r\left(f(\mathbf{x})\right),$$

where $\phi_r : \mathbb{R}^m \to \mathbb{R}$ is parameterized with respect to some reference point $\mathbf{r} \in \mathbb{R}^m$. Then R_r is said to be a *single-objective formulation at* $\mathbf{r}$ *of MOP* if the following conditions hold:

- If f is Lipschitz near some $\tilde{\mathbf{x}} \in \Omega$, then ψ_r is also Lipschitz near $\tilde{\mathbf{x}} \in \Omega$.
- If f is Lipschitz near some $\tilde{\mathbf{x}} \in \Omega$ with $f(\tilde{\mathbf{x}}) < \mathbf{r}$ component-wise, and if $\mathbf{d} \in \mathcal{T}_\Omega^{Cl}(\tilde{\mathbf{x}})$ is such that $f_i^o(\tilde{\mathbf{x}}; \mathbf{d}) < 0$ for $i = 1, 2, \ldots, m$ then $\psi_r^o(\tilde{\mathbf{x}}; \mathbf{d}) < 0$.

The first condition ensures the preservation of local Lipschitz continuity of the objective function f when reformulated into a SOO problem. The second condition involves Clarke descent directions for the f_i criteria for $i = 1, 2, \ldots, m$, and ψ_r.

BiMADS uses the *single-objective product formulation*. Let $\mathbf{r} \in \mathbb{R}^m$ be a reference point in the objective space. The single-objective product formulation is defined as

$$R_r^{prod} : \min_{\mathbf{x}\in\Omega} \psi_r^{prod}(\mathbf{x}) \text{ with } \psi_r^{prod}(\mathbf{x}) = \phi_r^{prod}\left(f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x})\right) = -\prod_{i=1}^{m} \left((\mathbf{r}_i - f_i(\mathbf{x}))_+\right)^2$$

where $(\mathbf{r}_i - f_i(\mathbf{x}))_+ = \max\{\mathbf{r}_i - f_i(\mathbf{x}), 0\}$ for $i = 1, 2, \ldots, m$. Audet et al [27] show that the single-objective product formulation is a single-objective formulation at $\mathbf{r}$ of MOP in the sense of Definition 22.

At the initialization step, BiMADS solves the two SOO problems

$$\min_{\mathbf{x}\in\Omega} f_1(\mathbf{x}) \text{ and } \min_{\mathbf{x}\in\Omega} f_2(\mathbf{x})$$

using MADS to get approximations of the extreme elements of the Pareto set. BiMADS stores all the non-dominated points found during the two resolutions in a list L^0.

BiMADS uses the fact that non-dominated points can be (lexicographically) ordered in the biobjective space. At each iteration, the reference point $\mathbf{r}^k \in \mathbb{R}^m$ is chosen such that it emphasizes the search towards zones in the objective space where three consecutive points are the most spaced. Due to the nature of the single-objective formulation problem, trial points are most likely to be in the dominance zone with respect to $\mathbf{r}^k$.

The main steps of BiMADS are summarized in Algorithm 2. The reader is invited to consult [27] for more details.

Algorithm 2 The BiMADS algorithm (adapted from [27])

Given $f : \mathbb{R}^n \to (\mathbb{R} \cup \{+\infty\})^2$, and $V^0 \subset \Omega$ a finite set of initial points,
Initialization: Solve the single-objective problems $\min_{\mathbf{x}\in\Omega} f_i(\mathbf{x})$ for $i = 1, 2$ using the MADS algorithm starting from V^0.
Let $L^0 := \{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^{|L^0|}\} \subset \Omega$ be the set of non-dominated points obtained from the two previous subproblem resolutions such that $f_1(\mathbf{x}^1) < f_1(\mathbf{x}^2) < \ldots < f_1(\mathbf{x}^{|L^0|})$ and $f_2(\mathbf{x}^1) > f_2(x^2) > \ldots > f_2(\mathbf{x}^{|L^0|})$. Initialize the weights $w(\mathbf{x}^j) = 0$ for all $x^j \in L^0$.
for $k = 0, 1, 2, \ldots$ **do**
Choice of the reference point:

1. If $|L^k| = 1$, set $\chi^k := \dfrac{\chi}{w(\mathbf{x}^1) + 1}$ where χ is some positive constant. Then solve again the single-objective problems $\min_{\mathbf{x}\in\Omega} f_i(\mathbf{x})$ for $i = 1, 2$ using the MADS algorithm until the mesh size parameter is below $O(\chi^k)$ and continue to the step: update L^k.

2. If $|L^k| = 2$, set $\chi^k := \dfrac{\|f(\mathbf{x}^2) - f(\mathbf{x}^1)\|^2}{w(\mathbf{x}^2) + 1}$ and $\mathbf{r}^k = (f_1(\mathbf{x}^2), f_2(\mathbf{x}^1))^\top$.

3. If $|L^k| > 2$, for each $j = 2, 3, \ldots, |L^k| - 1$, compute

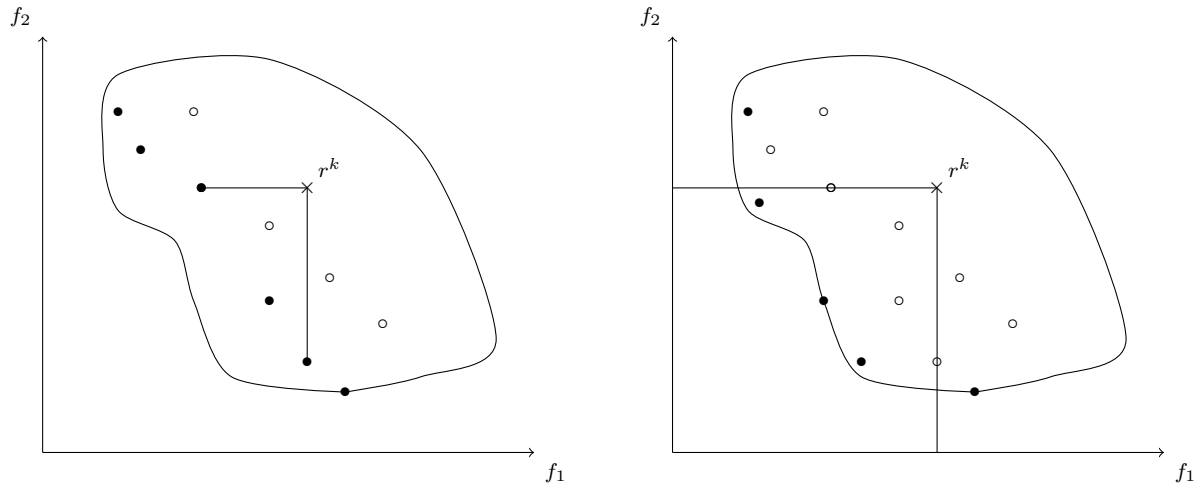
$$\chi^{k,j} := \frac{\|f(\mathbf{x}^{j+1}) - f(\mathbf{x}^j)\|^2 + \|f(\mathbf{x}^j) - f(\mathbf{x}^{j-1})\|^2}{w(\mathbf{x}^j) + 1}.$$

Let $j_0 \in \arg\max_{j\in\{2,3,\ldots,|L^k|-1\}} \chi^{k,j}$. Set $\mathbf{r}^k = (f_1(\mathbf{x}^{j_0+1}), f_2(\mathbf{x}^{j_0-1}))^\top$ and $\chi^k := \max_{j\in\{2,3,\ldots,|L^k|-1\}} \chi^{k,j}$.

Resolution of a single-objective problem using the reference point $\mathbf{r}^k$ and the objective function by MADS. Terminate MADS when the mesh size parameter is below $\mathcal{O}(\chi^k)$ or when the maximum budget of evaluations is reached.
Update L^k. Remove the dominated points of L^k, set $L^{k+1} := L^k$. For each $\mathbf{x}^j \in L^k$, update their weight: $w(\mathbf{x}^j) := w(\mathbf{x}^j) + 1$.
end for

The weights provide a flexible way to avoid staying in the same zone for too long during successive iterations of BiMADS. An illustration is given in Figure 2.7.

Under mild assumptions, the MADS convergence analysis guarantees that for each single-objective reformulation R_r of MOP at some reference point $\mathbf{r} \in \mathbb{R}^m$, there exists a limit point $\hat{\mathbf{x}} \in \Omega$ generated by MADS such that for all direction $\mathbf{d} \in \mathbb{R}^n$ belonging to the Clarke tangent cone $\mathcal{T}_\Omega^{Cl}(\hat{\mathbf{x}})$ (with $\mathcal{T}_\Omega^H(\hat{x}) \neq \emptyset$), there exists an index $i(\mathbf{d})$ such that the corresponding Clarke generalized derivative $f^o_{i(\mathbf{d})}(\hat{\mathbf{x}}; \mathbf{d}) \geq 0$. Essentially, it means there does not exist any direction in the hypertangent cone which is a descent direction for all objective functions.



(a) Determination of the reference point $\mathbf{r}^k$.

(b) Single objective-minimization.

Figure 2.7 An iteration of BiMADS (inspired by [27]).

Audet et al [28] propose an extension of BiMADS to tackle multiobjective blackbox optimization problems with more than two objectives, named MultiMADS. As BiMADS, MultiMADS solves several multiobjective single-objective reformulations R_r of MOP at some reference point $\mathbf{r} \in \mathbb{R}^m$. But its functioning is similar to the Normal Boundary Intersection (NBI) method [80]. MultiMADS starts by computing the extreme elements of the Pareto front by minimizing each criterion f_i separately. These objective vectors are used to compute the *convex hull of individual minima* [28] (CHIM). At each iteration, MultiMADS selects an element of the CHIM as the reference vector $\mathbf{r}^k \in \mathbb{R}^m$ and solves a single-objective formulation using it. All non-dominated points are collected and added to a list as for BiMADS. Unfortunately, no implementation is available.

Several issues are raised with such scalarization-based methods. The first is the number of evaluations to allocate to each single-objective problem: too few and no promising points can be reached; too many and the algorithm can lack budget to explore potential promising zones in the objective space. Futhermore, at the end of each resolution, the frame and mesh size parameters are reset. These parameters provide valuable information on the zone of research in the decision space to find better solutions, but they are not exploited by both algorithms. BiMADS and MultiMADS use an important number of evaluations to refine the mesh, when this budget could rather be dedicated to enrich the Pareto front approximation.

Trust-region based approaches

The extension of single-objective trust-region DFO algorithms to MOO has been very limited. As far as we know, three convergent-based methods, all based on scalarization reformulations, have been investigated [208, 223, 242].

The BOTR algorithm. The BiObjective Trust-Region (BOTR) algorithm [223] is a DFO unconstrained method for biobjective optimization strongly inspired by BiMADS. Similarly to BiMADS, BOTR keeps a list of non-dominated points which gets closer to the Pareto front as long as the algorithm unfolds. Each element of the list possesses its own trust-region radius.

At iteration k, BOTR selects one element $\mathbf{x}^k$ of the list with its associated trust-region radius Δ^k as the current incumbent. It then constructs the fully linear models (quadratic minimum Frobenius norm or regression) in a trust-region of radius $\tilde{\Delta}^k \leq \Delta^k$:

- $\widetilde{f}_1^k$ and $\widetilde{f}_2^k$ are the quadratic models of objective functions f_1 and f_2.
- $\widetilde{\psi}_r^k$ is the quadratic model of the single-objective formulation ψ_r^{prod} of MOP at $\mathbf{r} = f(\mathbf{x}^k)$, i.e.
$$\forall \mathbf{x} \in \mathbb{R}^n, \psi_r^{prod}(\mathbf{x}) = -\prod_{i=1}^{m} \left((\mathbf{r}_i - f_i(\mathbf{x}))_+ \right)^2$$
with $(\mathbf{r}_i - f_i(\mathbf{x}))_+ = \max\{\mathbf{r}_i - f_i(\mathbf{x}), 0\}$ for $i = 1, 2, \ldots, m$.

As in the single-objective case, a test is performed to estimate the quality of these models according to f_1, f_2 and ψ_r^{prod}. If the test fails, the trust-region radius $\tilde{\Delta}^k$ is reduced, and the models are reconstructed using sample points belonging to this new zone. When the three models are accurate enough, the trust-region radius Δ^k is adjusted using the gradient of all models evaluated at $\mathbf{x}^k$ and the radius $\tilde{\Delta}^k$.

BOTR then solves the three SOO trust-region subproblems in the zone of radius Δ^k. BOTR computes the reduction ratios of the generated solutions. BOTR also computes the reduction ratios of the points involved in the construction of the models using ψ_r^{prod}. Sample points with insufficient reduction ratio value are discarded, since their ψ_r^{prod} value is not significantly better or worse than the current incumbent $\mathbf{x}^k$.

BOTR affects a trust-region radius superior or equal to Δ^k to all sample points satisfying the reduction ratio condition. The three approximated or exact solutions of trust-region subproblems are given a trust-region radius whose value depends on their reduction ratio.

All these points are added to the list of non-dominated decision vectors, and the set of non-dominated points is then updated.

At iteration k, the choice of $\mathbf{x}^k$ depends on two factors:

- Either all generated points at iteration $k-1$ have an insufficient reduction ratio value. In this case, $\mathbf{x}^k := \mathbf{x}^{k-1}$ and $\Delta^k := \tau\Delta^{k-1}$ with $\tau \in (0;1)$.
- Or $\mathbf{x}^k$ is selected in the list of non-dominated points as the most isolated in the objective space according to the Euclidean distance between three consecutive points if its trust-region radius is above a certain tolerance Δ_{tol} (see [223] for more details).

When all elements of the list have their trust-region radius below Δ_{tol}, BOTR stops.

The authors of [223] show that under strict assumptions (notably that f_1, f_2 and ψ_r^{prod} are $\mathscr{C}^1$ and their gradients are Lipschitz continuous on some specific open set), BOTR generates a subsequence of iterates $\mathbf{x}^k$ converging to at least one first-order critical point for one of the three functions f_1, f_2 or ψ_r^{prod}. Numerical experiments show that BOTR, on the analytical benchmarks presented in this article, captures some parts of the Pareto front, even though the problem is not differentiable everywhere. However, to the best of our knowledge, no implementation is available. This algorithm may not be practical to deal with general blackbox problems as the number of assumptions is too restrictive.

The (Expensive) Multiobjective (heterogeneous) Trust-region algorithm. The Multiobjective Heterogeneous Trust-region (MHT) [242] and the Expensive Multiobjective Trust-region (EMT) [208] algorithms are two variants of the same method specifically designed to solve expensive unconstrained MOO problems. Both guarantee (under some strict assumptions implying the differentiability of the function) to generate only one locally Pareto optimal point. MHT [242] considers heterogeneous MOO problems, mixing differentiable and expensive blackbox objectives. EMT [208] targets multiobjective BBO problems, where all objective functions are expensive simulations. A short description is done ; more details of these two works can be found in [241].

At iteration k, EMT constructs quadratic interpolation models (or linear models when the number of variables is large) $\widetilde{f}_i^k$ for $i = 1, 2, \ldots, m$ in a ball $\mathcal{B}_{\Delta^k}(\mathbf{x}^k)$ of radius Δ^k and centre $\mathbf{x}^k$. Then the ideal objective vector of model $\widetilde{f}^k$ in the ball $\mathcal{B}_{\Delta^k}(\mathbf{x}^k)$ is computed, i.e.

$$\tilde{\mathbf{y}}^{I,k} = \left(\min_{\mathbf{x}\in\mathcal{B}_{\Delta^k}(\mathbf{x}^k)} \widetilde{f}_1^k(\mathbf{x}), \min_{\mathbf{x}\in\mathcal{B}_{\Delta^k}(\mathbf{x}^k)} \widetilde{f}_2^k(\mathbf{x}), \ldots, \min_{\mathbf{x}\in\mathcal{B}_{\Delta^k}(\mathbf{x}^k)} \widetilde{f}_m^k(\mathbf{x}) \right)^\top.$$

EMT and MHT look for a sufficient decrease in the function values. The authors propose to move as long as the trust region allows in the direction of $\tilde{\mathbf{y}}^{I,k}$, using the Pascoletti-Serafini scalarization [206], given by

$$\begin{aligned} \min \quad & t \\ \text{s.t.} \quad & f_i(\mathbf{x}^k) + t\ \mathbf{v}_i^k - \widetilde{f}_i^k(\mathbf{x}) \geq 0, i = 1, 2, \ldots, m \\ & t \in \mathbb{R}, \mathbf{x} \in \mathcal{B}_{\Delta^k}(\mathbf{x}^k) \end{aligned}$$

where $\mathbf{v}^k = f(\mathbf{x}^k) - \tilde{\mathbf{y}}^{I,k}$.

For each objective $i = 1, 2, \ldots, m$, EMT computes a reduction ratio ρ_i^k. The ratio used for the trial acceptance test is defined as $\tilde{\rho}^k = \min_{i=1,2,\ldots,m} \rho_i^k$. Depending on this ratio, the trust-region radius is updated.

MHT constructs second order approximations of the analytical objective if Hessian and derivatives are available. The acceptance test for MHT uses the ratio ρ_ϕ^k defined by:

$$\rho_\phi^k = \frac{\phi(\mathbf{x}^k) - \phi(\mathbf{x}^{trial})}{\widetilde{\phi}^k(\mathbf{x}^k) - \widetilde{\phi}^k(\mathbf{x}^{trial})}$$

where $\phi = \max\limits_{1 \leq i \leq m} f_i$ and $\widetilde{\phi}^k = \max\limits_{1 \leq i \leq m} \widetilde{f}_i^k$. This ratio guarantees only the existence of a descent direction for at least one objective, whereas EMT forces a descent direction for all objectives.

Assuming that the objective functions are twice differentiable, both algorithms generate a subsequence of iterates which converges to a local Pareto critical point. However, they remain restricted as they are only designed to find one local Pareto optimal solution.

Others

In the context of scalarization-based methods, many researchers have investigated the use of surrogate-based algorithms. Among the works not yet mentioned in the previous subsections [43, 158], one can mention the extension of single-objective Bayesian optimization via the use of the hypervolume indicator [279]. Given a set of non-dominated points in the objective space, the hypervolume indicator represents the volume of the space dominated by these points and bounded above by a reference objective vector (see Chapter 4 of this document for a more thorough description). This indicator is to be maximized. As far as we know, this research starts by Emmerich [101] which proposes the Expected Hypervolume Improvement Indicator. This formulation transforms a multiobjective optimization problem into a single-objective one which can be passed to efficient existing solvers. However, this indicator is computationally expensive, as its complexity increases exponentially with the number of

objectives [57]. To handle this issue, researchers have proposed stochastic approximations of the hypervolume [44, 111, 112, 269], faster to compute at the detriment of accuracy.

2.3.2 Methods with a posteriori articulation of preferences

Methods with a posteriori articulation of preferences do not aggregate any of the objective functions. They directly work with the original MOO problem. All derivative-free convergent-based approaches presented in this subsection follow the same mechanism. They maintain a current solution set, composed of all non-dominated points found. The algorithm improves this solution set along the iterations, using some iteration rules. From a theoretical point of view, these methods guarantee the convergence of this solution set towards a local Pareto optimal point, under reasonable assumptions (e.g., existence of local minima or (locally) Lipschitz objectives). Practically, it can be observed on analytical benchmarks [74] that the list of non-dominated points gets closer to the Pareto front, as the number of iterations increases.

Direct search methods

The Direct MultiSearch algorithm The Direct Multisearch (DMS) algorithm [74] is a framework which extends classical SOO direct search algorithms (e.g., GSS) to MOO. As in SOO, each iteration is built around two steps: an optional search and a poll, on which convergence depends.

DMS adopts an extreme barrier approach to deal with constraints, by setting:

$$f_\Omega(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & \text{if } \mathbf{x} \in \Omega, \\ (+\infty, +\infty, \ldots, +\infty)^\top & \text{otherwise.} \end{cases}$$

At iteration k, DMS starts from a list of previously evaluated non-dominated feasible points, with their own associated step size parameters. This *iterate list* is denoted as L^k. DMS then selects an element $(\mathbf{x}^k, \delta^k)$ of the iterate list L^k as the current incumbent. The search enables the practitioner to evaluate a finite number of candidates according to an user-based strategy. During the poll step, the algorithm generates poll points belonging to the poll set, defined as:

$$P^k = \left\{ \mathbf{x}^k + \delta^k \mathbf{d} : \mathbf{d} \in \mathbb{D}^k \right\}$$

where $\mathbb{D}^k$ is a positive spanning set.

All new evaluated candidates are submitted to a multiobjective sufficient decrease test as

follows. Let $D(L^k)$ be the subset of the objective space non-dominated by the points of L^k and $\rho : \mathbb{R}_+ \to \mathbb{R}_+$ a forcing function. During the search step, $\mathbf{x} \in \mathbb{R}^n$ is said to be non-dominated with respect to the iterate list L^k if the l_∞ distance between $f(\mathbf{x})$ and $D(L^k)$ is larger than $\rho\left(\delta^k\right)$. During the poll step, $\mathbf{x}^k + \delta^k\mathbf{d} \in \mathbb{R}^n$ is said to be non-dominated with respect to the iterate list L^k if the l_∞ distance between $f(\mathbf{x}^k + \delta^k\mathbf{d})$ and $D(L^k)$ is larger than $\rho\left(\delta^k\|\mathbf{d}\|\right)$ where $\mathbf{d}$ is a polling direction.

DMS adds all evaluated sufficiently non dominated points to the list L^k with an associated step size parameter $\delta \geq \delta^k$. L^k is then filtered to remove all new dominated points.

An iteration is said to be successful if the list L^k changes at the end of iteration k. Otherwise, it is unsuccessful and δ^k is decreased. Algorithm 3 summarizes the main steps of DMS.

Algorithm 3 Direct MultiSearch (DMS) (adapted from [74])

Given $f : \mathbb{R}^n \to (\mathbb{R} \cup \{+\infty\})^m$, choose $\mathbf{x}^0 \in \Omega$ the initial starting point, $\delta^0 > 0$ the initial step size parameter, $0 < \beta_1 \leq \beta_2 < 1$ and $\gamma \geq 1$. Let $\mathcal{D}$ be a (possibly infinite) union of positive spanning sets. Initialize the list of non-dominated points $L^0 = \{(\mathbf{x}^0, \delta^0)\}$.

for $k = 0, 1, 2, \ldots$ **do**

Selection of the current incumbent point: Select one element $\left(\mathbf{x}^k, \delta^k\right)$ of L^k as the current incumbent.

Search step (optional): Evaluate f_Ω at a finite set of points S^k. Set $L_{add} := \{(\mathbf{x}, \delta^k) : \mathbf{x} \in S^k\}$.

Call $L_{filtered} := \texttt{filter}(L^k, L_{add})$ to remove all dominated points from $L^k \cup L_{add}$ using sufficient decrease to see if points in L_{add} are nondominated relatively to L^k. If $L^k \neq L_{filtered}$, set $L^{k+1} := L_{filtered}$, declare the iteration as successful and skip the poll step.

Poll step: Select a positive spanning set $\mathbb{D}^k$ from the set $\mathcal{D}$. Evaluate f_Ω at the set of poll points $P^k = \{\mathbf{x}^k + \delta^k\mathbf{d} : \mathbf{d} \in \mathbb{D}^k\}$. Set $L_{add} := \{(\mathbf{x}, \delta^k) : \mathbf{x} \in P^k\}$.

Call $L_{filtered} := \texttt{filter}(L^k, L_{add})$ to remove all dominated points from $L^k \cup L_{add}$ using sufficient decrease to see if points in L_{add} are nondominated relatively to L^k. If $L^k \neq L_{filtered}$, set $L^{k+1} := L_{filtered}$ and declare the iteration as successful. Otherwise, declare the iteration as unsuccessful.

Parameter update: If the iteration was successful, replace all elements $\left(\mathbf{x}, \delta^k\right)$ of $L^{k+1} \setminus L^k$ by $\left(\mathbf{x}, \delta^{k+1}\right)$ with $\delta^{k+1} = \gamma\delta^k$; replace also $(\mathbf{x}^k, \delta^k)$, if in L^{k+1}, by $(\mathbf{x}^k, \delta^{k+1})$. Otherwise replace the poll center $(\mathbf{x}^k, \delta^k)$ by $(\mathbf{x}^k, \delta^{k+1})$ with $\delta^{k+1} \in [\beta_1\delta^k, \beta_2\delta^k]$.

end for

The choice of the current incumbent remains flexible. Custódio et al [74] propose two strategies to choose it. The current incumbent is chosen as the first point of the iterate list. New found non-dominated points are added to the end of the list. The current incumbent is then moved to the end of the list at the end of the iteration. This strategy enables to diversify the search in the feasible space Ω. Another strategy consists in selecting the current incumbent as the one in the least dense zones of the approximated Pareto front using a spread distance.

More details can be found in [74].

Under mild assumptions, the DMS convergence analysis given in [74] guarantees the existence of at least one accumulation point $\hat{\mathbf{x}}$ such that for each direction $\mathbf{d}$ belonging to the Clarke tangent cone $\mathcal{T}_\Omega^{Cl}(\hat{\mathbf{x}})$ [59] (if $\mathcal{T}_\Omega^{H}(\hat{\mathbf{x}}) \neq \emptyset$), there exists an index $i(\mathbf{d}) \in \{1, 2, \ldots, m\}$ with its Clarke generalized derivative $f_{i(\mathbf{d})}^0(\hat{\mathbf{x}}; \mathbf{d}) \geq 0$. Practically, computational experiments have shown that DMS is able to get several elements of the Pareto front.

The DMS framework adapts easily to the MADS algorithm by using integer lattices [74] instead of sufficient decrease defined by a forcing function. Convergence results remain similar.

Practically, DMS uses a complete polling strategy with the set $\mathcal{D} = [\mathbf{I}_n - \mathbf{I}_n]$ where $\mathbf{I}_n$ is the identity matrix of size $n \times n$ and the constant forcing function $\rho : t \in \mathbb{R}^+ \mapsto 0$ (which is the same to use the classical non-dominated relation). Computational experiments show that the spread distance strategy is more efficient than the FIFO (First In First Out) strategy.

A globalization strategy of the DMS algorithm can be found in [73]. A model-based search strategy, inspired by [75], is proposed in [53]. The authors show that the incorporation of quadratic models in DMS provides computational improvements compared to the original DMS algorithm. However, it adds additional complexity (the search strategy requires the resolution of at worst $2^m - 1$ subproblems, which is only acceptable when m is small [53]). Theoretical complexity results for DMS are derived in [77]. New parallel strategies are explored in [240]. A variant of DMS is equally proposed in [90].

Multiobjective Optimization Implicit Filtering The Multiobjective Optimization Implicit Filtering (MOIF) algorithm [126] is an extension of the single-objective Implicit Filtering method [154] to bound-constrained MOO. For the rest of this description, the feasible set is given by $\Omega = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}\}$ with $\mathbf{l}, \mathbf{u} \in \mathbb{R}^n$. Although two versions of MOIF are described, the description focuses on its "front" version, which aims at capturing the Pareto front.

Similar to DMS, MOIF considers a list of non-dominated points

$$L^k = \left\{ \left(\mathbf{x}^j, \delta^j \right), \mathbf{x}^j \in \Omega, j = 1, 2, \ldots |L^k| \right\}$$

with their associated stepsize $\delta^j > 0$ for $j = 1, 2, \ldots, |L^k|$.

At the beginning of iteration k, MOIF selects one element $(\mathbf{x}^k, \delta^k)$ (using the same criteria as DMS) of the list L^k as the current incumbent. With this pair, MOIF evaluates a set of

stencil points [126], defined as

$$\left\{\mathbf{x}^k + \delta^k\mathbf{e}_1, \mathbf{x}^k + \delta^k\mathbf{e}_2, \ldots, \mathbf{x}^k + \delta^k\mathbf{e}_n, \mathbf{x}^k - \delta^k\mathbf{e}_1, \mathbf{x}^k - \delta^k\mathbf{e}_2, \ldots, \mathbf{x}^k - \delta^k\mathbf{e}_n\right\}$$

with $\mathbf{e}_i$ the i-th coordinate vector of dimensions n. In other terms, a stencil centered in $\mathbf{x}^k$ of stepsize δ^k is equivalent to a poll set of center $\mathbf{x}^k$ and stepsize δ^k with polling directions $[\mathbf{I}_n \ -\mathbf{I}_n]$ where $\mathbf{I}_n$ is the identity matrix of dimensions $n \times n$. A temporary copy of L^k denoted as $\tilde{L}^k$ collects all feasible points with an associated stepsize whose value is equal to δ^k. If $\tilde{L}^k$ is different from L^k, the iteration is considered as successful and $\tilde{L}^k$ becomes L^{k+1}.

Otherwise, the algorithm has not found any other new non dominated point. In this case, MOIF attempts a linesearch. Using pre-evaluated candidates on the stencil set, MOIF computes an approximated gradient of each objective function at $\mathbf{x}^k \in \Omega$. Given $\mathbf{x} \in \Omega$ and $\delta > 0$, the approximated gradient $\nabla_\delta f_i(\mathbf{x})$ for $i = 1, 2, \ldots, m$ at $x \in \Omega$ is defined as

$$\nabla_\delta f_i(\mathbf{x}) = \left(\frac{\partial_\delta f_i}{\partial x_1}(\mathbf{x}), \frac{\partial_\delta f_i}{\partial x_2}(\mathbf{x}), \ldots, \frac{\partial_\delta f_i}{\partial x_n}(\mathbf{x})\right)^\top$$

where for all $j = 1, 2, \ldots, m$,

$$\frac{\partial_\delta f_i}{\partial x_j}(\mathbf{x}) = \begin{cases} \dfrac{f_i(\mathbf{x} + \delta\mathbf{e}_j) - f_i(\mathbf{x} - \delta\mathbf{e}_j)}{2\delta} & \text{if } \mathbf{x} + \delta\mathbf{e}_j \in \Omega,\ \mathbf{x} - \delta\mathbf{e}_j \in \Omega; \\ \dfrac{f_i(\mathbf{x} + \delta\mathbf{e}_j) - f_i(\mathbf{x})}{\delta} & \text{if } \mathbf{x} + \delta\mathbf{e}_j \in \Omega,\ \mathbf{x} - \delta\mathbf{e}_j \notin \Omega; \\ \dfrac{f_i(\mathbf{x}) - f_i(\mathbf{x} - \delta\mathbf{e}_j)}{\delta} & \text{if } \mathbf{x} + \delta\mathbf{e}_j \notin \Omega,\ \mathbf{x} - \delta\mathbf{e}_j \in \Omega; \\ +\infty & \text{otherwise.} \end{cases}$$

If all components of $\nabla_{\delta^k} f_i(\mathbf{x}^k)$ are finite, the following approximated steepest descent direction d^k, solution of the following linear problem (inspired by [117])

$$\begin{aligned} \beta^k = \ & \min_{\mathbf{d} \in \mathbb{R}^n} \ \max_{i \in \{1,2,\ldots,m\}} \nabla_{\delta^k} f_i(\mathbf{x}^k)^\top \mathbf{d} \\ & \text{s.t} \quad \mathbf{x}^k + \mathbf{d} \in \Omega \end{aligned}$$

is computed.

If β^k satisfies the following tolerance criterion $\beta^k \geq -\tau\delta^k$, where $\tau \in (0; 1)$ is a fixed parameter, the line search is not executed. Otherwise, MOIF performs a Goldstein linesearch. If it exists, the new point found at the end of this strategy is then added to L^k. L^k is then filtered to remove non dominated points, resulting in the new iterate list L^{k+1}. Otherwise, the iteration is considered as a failure.

When the iteration is marked as unsuccessful, the stepsize δ^k of the current incumbent $\mathbf{x}^k$ is then reduced and the counter iteration is increased.

Under smoothness assumptions, it is proved than MOIF generates a subsequence of iterates converging to a Pareto stationary point satisfying first order conditions [126].

The performance of this algorithm partly depends on the approximation of the gradient used in the linesearch. If the true function is nonsmooth, this descent direction may not be accurate. However, the integration of a stencil phase (as in DMS) could still keep it robust.

Line search approaches : the Derivative Free MultiObjective algorithm

The Derivative Free MultiObjective (DFMO) algorithm [180] is a constrained derivative-free linesearch algorithm for multiobjective optimization inspired by DMS. This algorithm deals with inequality constraints via an exact penalty approach.

Precisely, given the constrained problem (MOP) where $\Omega = \{\mathbf{x} \in \mathcal{X} : c_j(\mathbf{x}) \leq 0, j \in \mathcal{J}\}$ with $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}\}$ defined as bound constraints, a parameter $\varepsilon > 0$, the following penalty functions are defined as

$$Z_i(\mathbf{x}; \varepsilon) = f_i(\mathbf{x}) + \frac{1}{\varepsilon} \sum_{j \in \mathcal{J}} \max\{0, c_j(\mathbf{x})\} \text{ for } i = 1, 2, \ldots, m.$$

The penalized bound constrained problem is then considered:

$$\min_{\mathbf{x} \in \mathcal{X}} Z(\mathbf{x}; \varepsilon) = \left[Z_1(\mathbf{x}; \varepsilon), Z_2(\mathbf{x}; \varepsilon), \ldots, Z_m(\mathbf{x}; \varepsilon)\right]^\top.$$

Similarly to DMS, DFMO updates a list of non-dominated points (according to the Z function) defined at each iteration k by

$$L^k = \{(\mathbf{x}^j, \delta^j),\ \mathbf{x}^j \in \mathcal{X},\ \delta^j > 0,\ j = 1, 2, \ldots, |L^k|\}$$

where δ^j is the trial step associated with point $\mathbf{x}^j$.

At iteration k, DFMO creates a temporary copy $\tilde{L}^k$ of L^k. For each point of the iterate list L^k, the algorithm starts a linesearch along a direction $\mathbf{d}^k$ projected onto the decision set $\mathcal{X}$. If the direction $\mathbf{d}^k$ guarantees a sufficient decrease, i.e. there exists at least one objective function that decreases enough for a new generated point along $\mathbf{d}^k$ for each point in $\tilde{L}^k$, then the algorithm executes a "larger" search along this direction, generating a new set of points. Otherwise, the trial step is reduced. Algorithm 4 summarizes the main points of DFMO.

Algorithm 4 The DFMO algorithm (adapted from [180])

Given $f : \mathbb{R}^n \to \mathbb{R}^m$, $c_j : \mathbb{R}^n \to \mathbb{R}$ for $j \in \mathcal{J}$, choose $\varepsilon > 0$, $\gamma > 0$, $\tau \in (0; 1)$, $L^0 = \{(\mathbf{x}^j, \alpha^j),\ \mathbf{x}^j \in \mathcal{X},\ \alpha^j > 0,\ j = 1, 2, \ldots, |L^0|\}$, a sequence $\{\mathbf{d}^k\} \subset \mathbb{R}^n$ such that $\|\mathbf{d}^k\| = 1$ for all k.
for $k = 0, 1, 2, \ldots$ **do**
  Set $\tilde{L}^k := L^k$.
  for $j = 1, 2, \ldots, |L^k|$ **do**
    Let $(\mathbf{x}^j, \delta^j)$ be the j-th element of L^k.
    if $Z([\mathbf{x}^j + \delta^j \mathbf{d}^k]_{[l;u]}; \varepsilon) \not\succ Z(\mathbf{x}^l; \varepsilon) - \gamma \left(\delta^j\right)^2 \mathbf{e}$ for all $\mathbf{x}^l \in \tilde{L}^k$ **then**
      Apply a linesearch-based method along the $\mathbf{d}^k$ direction; add new non-dominated points and remove new dominated points from $\tilde{L}^k$.
    else
      Failure Step: If $(\mathbf{x}^j, \alpha^j) \in \tilde{L}^k$, set $\tilde{L}^k := \left(\tilde{L}^k \setminus \{(\mathbf{x}^j, \delta^j)\}\right) \cup \{(\mathbf{x}^j, \tau\delta^j)\}$.
    end if
    Set $L^{k+1} := \tilde{L}^k$.
  end for
end for

DFMO possesses an important drawback. At each iteration, it evaluates at least $|L^k|$ points, whereas an iteration of DMS involves at most the evaluation of $2n$ points (during the poll step). If $|L^k|$ is large, it may become a bargain. However, DFMO guarantees the convergence of L^k to a set of locally Pareto-Clarke optimal points, assuming that the objective functions f_i for $i = 1, 2, \ldots, m$ and the c_j functions for $j \in \mathcal{J}$ are Lipschitz continuous on $\mathcal{X}$. On the contrary, DMS only guarantees the convergence to one local Pareto-Clarke stationary point. In [126], the authors also claim that DMS is better than DFMO for bound-constrained problems. However, it remains one of the few algorithms to manage constraints in derivative-free multiobjective optimization.

Last notes

The most common algorithms in blackbox multiobjective optimization are evolutionary algorithms [84, 89], as shown by the important literature on this topic [88, 89]. Nonetheless, they are heuristics and require an important number of evaluations to work (the authors of [266] use a maximal budget of between 2×10^5 and 5×10^5), which is not practical when functions are expensive. When combined with surrogate models, this last drawback can be removed. Surrogate methods combining RBF models and evolutionary approaches can for example be found in [3, 196, 214, 257]. They do not get a convergence certificate, or if it is the case, it is stochastic-based [214].

CHAPTER 3 ORGANIZATION OF THE THESIS

This chapter details the organization of the research done in this thesis. It also exhibits the motivation points and the initially expected results.

The previous chapter has described several state-of-the-art MOO derivative-free and blackbox methods. It is more difficult to assess the performance of new multiobjective BBO solvers empirically than it is for SOO algorithms. Indeed, in a multiobjective context, the solution is often a set (i.e., the Pareto front); the total ordering induced by a scalar-valued objective function is lost, making it difficult to rank approximation sets obtained by different solvers. Researchers have often employed visualization techniques, as in [27, 28, 223]. However, these tools do not generalize when more than three objectives are involved. They fail to quantify the difference in quality between different approximation sets. They can only highlight the efficiency of a method on one problem. Performance indicators, which map solution sets to a real value, capture several properties of a Pareto front approximation. They can be used to offer a ranking of different algorithms.

Chapter 4 corresponds to [11], which has been published in *European Journal of Operational Research*. This survey classifies a set of 63 performance indicators according to their properties (cardinality, convergence, distribution and spread), lists their advantages and potential drawbacks, and describes some applications. The first intent of this work is to identify good quality metrics which can provide a baseline to implement reliable and intuitive benchmarking comparison tools (similar to what is introduced in [74]), or to guide the search in the objective space towards better solutions. Note that when this survey was into peer-reviewed revision, the study [175] was published, whose authors list more performance indicators (exactly 100). But their work is mostly oriented towards the evolutionary community. The reader is invited to consult it as a useful complement to our work.

Chapters 5 and 6 are dedicated to the presentation of a new extension of MADS to multiobjective constrained optimization, designed as DMulti-MADS. Each chapter focuses on a facet of this new algorithm.

As detailed in the previous chapter, there already exist two extensions of MADS to constrained MOO: the BiMADS [27] and MultiMADS [28] algorithms. The main drawback with such scalarization-based methods is the poor management of the total budget of blackbox evaluations, distributed over single-objective subproblems. Indeed, these algorithms can waste some evaluation calls in a part of the objective space where no improving solution lies. This budget could be dedicated to the exploration of more promising zones in the objective

space. Besides, each time these algorithms start the resolution of a new single-objective problem, they lose information given by the frame and mesh size parameters at the end of the previous optimization. On the other hand, DMS introduced by Custódio et al. [74], presented in the previous chapter, is a framework which generalizes all types of direct search methods to MOO. It maintains a list of non-dominated points, and selects at every iteration a center to perform a poll. It then wastes fewer evaluations, and exploits more of the knowledge of the step size parameter associated to the current poll center. However, DMS only guarantees that it generates at least a sequence of points which converge to a local stationary Pareto point. In a multiobjective context, one would want to find methods which ensure convergence to a set of local Pareto stationary points. DFMO [180] paves the way toward such approaches, but it appears to be less efficient than DMS [126] on bound-constrained problems. As indicated before, it is too restrictive in its convergence assumptions within a blackbox context.

The core of the DMulti-MADS algorithm is presented in Chapter 5, which corresponds to [40], published in *Computational Optimization and Applications*. DMulti-MADS is an extension of MADS to many objectives. From BiMADS, it conserves the same success condition criterion (dominance according to a point and not a list). It also possesses a similar center selection strategy, which is not limited to two objectives. As for DMS, it keeps a list of non-dominated points with their mesh and frame size parameters, from which a poll center is selected. It exploits more efficiently its evaluation budget to improve the current solution set. The main difference with DMS is that DMulti-MADS imposes a restriction on the choice of the poll center. This condition, combined with a convergence analysis close to DMS, enables DMulti-MADS to guarantee convergence to a set of locally stationary points, under more general assumptions than DFMO. Numerical experiments are conducted to evaluate the performance of DMulti-MADS over state-of-the-art algorithms on a set of 100 analytical MOO problems. A DMS variant is implemented to see how the new features affect performance. This chapter also defines data profiles involving the hypervolume indicator [279], based on the conclusions of the previous survey [11]. The hypervolume indicator is intuitive and considers the distance to the Pareto front. This is not the case with the purity metric [74]. Expected results are a better efficiency compared to `Nomad` (BiMADS), and performance at least similar to other state-of-the-art algorithms (NSGA-II, MOIF, DMS).

DMulti-MADS like DMS imposes the use of a feasible starting point. This condition is not always available in a real engineering context. Similarly to SOO [16], it could be beneficial to conceive multiobjective BBO methods which can violate constraints during the optimization process, to converge faster later toward feasible solutions. The deterministic multiobjective DFO literature is rather scarce on this topic. To the best of our knowledge, only the DFMO

algorithm [180], which relies on a penalty-based strategy, has been proposed. BiMADS, which uses MADS as its constrained SOO solver could also exploit general constraints. However, the conception of BiMADS precedes the development of the progressive barrier approach for SOO [16]. For this reason, the integration of the progressive barrier into BiMADS has never been theoretically investigated.

Chapter 6, which corresponds to the last article [41], aims at closing this gap. It proposes two new approaches to handle blackbox inequality constraints for multiobjective BBO problems. These two extensions of DMulti-MADS aggregate inequality constraints into a single constraint violation function. The first is a two-phase approach. It searches from a feasible solution by minimizing the aggregation function from an infeasible point. It uses this point for the optimization process detailed in Chapter 5. The second is an extension of the progressive barrier approach to MOO denoted DMulti-MADS-PB. At each iteration, DMulti-MADS-PB aims to improve a feasible and infeasible incumbent. The infeasible incumbent has a constraint violation function value below a threshold, which gradually decreases along the iterations. For both variants, convergence results are proved. Numerical experiments compare these two extensions with the penalty-based approach proposed by [180] using analytical benchmarks and three "real" engineering applications with other state-of-the-art solvers (BiMADS, DFMO, NSGA-II). The objective is to show than the two variants outperform the penalty-based approach and are at least as efficient as NSGA-II, DFMO or BiMADS.

The three articles are now detailed in the next three chapters.

CHAPTER 4 ARTICLE 1: PERFORMANCE INDICATORS IN MULTIOBJECTIVE OPTIMIZATION

Twas brillig, and the slithy toves
Did gyre and gimble in the wabe:
All mimsy were the borogoves,
And the mome raths outgrabe.

Lewis Caroll, *Jabberwocky*

Authors Charles Audet, Jean Bigeon, Dominique Cartier, Sébastien Le Digabel and Ludovic Salomon

Journal Published in *European Journal of Operational Research*

Abstract In recent years, the development of new algorithms for multiobjective optimization has considerably grown. A large number of performance indicators has been introduced to measure the quality of Pareto front approximations produced by these algorithms. In this work, we propose a review of a total of 63 performance indicators partitioned into four groups according to their properties: cardinality, convergence, distribution and spread. Applications of these indicators are presented as well.

Keywords multiobjective optimization, quality indicators, performance indicators.

4.1 Introduction

Since the eighties, a large number of methods has been developed to treat multiobjective optimization problems (e.g [45, 65, 71, 83, 235]). Given that conflicting objectives are provided, the set of solutions, the *Pareto set*, is described as the set of best decision vectors corresponding to the best trade-off points in the objective space. Knowledge of the Pareto set enables the decision maker to visualize the consequences of his/her choices in terms of performance for a criterion at the expense of one or other criteria, and to make appropriate decisions.

Formally, a feasible vector $\mathbf{x}^1$ is said to *(Pareto)-dominate* another feasible vector $\mathbf{x}^2$ if $\mathbf{x}^1$ is at least as good as $\mathbf{x}^2$ for all the objectives, and strictly better than $\mathbf{x}^2$ for at least one objective.

The decision vectors in the feasible set that are not dominated by any other feasible vector are called *Pareto optimal*. The set of non-dominated points in the feasible set is the set of *Pareto solutions*, whose images (by the objective functions) constitute the *Pareto front*.

In single-objective minimization problems, the quality of a given solution is trivial to quantify: the smaller the corresponding objective function value, the better. However, evaluating the quality of an approximation of a Pareto set is non trivial. The question is important for the comparison of algorithms, the definition of stopping criteria, or even the design of multiobjective optimization methods. According to [276], a Pareto set approximation should satisfy the following:

- The distance between the Pareto front and its representation in the objective space should be minimized.
- A good (according to some metric) distribution of the points of the corresponding approximated front in the objective space is desirable.
- The extent of the corresponding approximated front should be maximized, i.e., for each objective, a wide range of values should be covered by the non-dominated points.

To answer this question, many metrics called *performance indicators* [204, 277] have been introduced. Performance indicators can be considered as mappings that assign scores to Pareto front approximations.

Surveys of performance indicators already exist but they focus only on some specific properties. In [65, chapter 7], the authors list some performance indicators to measure the quality of a Pareto front approximation. In [204], an exhaustive survey is conducted on a vast number of performance indicators which are grouped according to their properties. Mathematical frameworks to evaluate performance indicators are proposed in [159, 280] and additional measures and algorithms are listed in [110]. In [58], the authors review some performance indicators and analyze their drawbacks. In [151], an empirical study focuses on the correlations between different indicators with their computational complexity on concave and convex Pareto fronts. Finally, the usage of indicators proposed by the multiobjective evolutionary optimization community prior to 2013 is analyzed in [218].

Another survey [175] on the quality evaluation of solution sets was recently published. It complements the present study by categorizing more performance indicators, but presents them at a higher level of description. Most of the indicators in the present work are listed in [175], but not all of them (e.g., the ones proposed in [74]). While [175] is mainly oriented toward the evolutionary algorithms community, the present work addresses the whole

operational research community, and also addresses some issues in more detail, such as complexity computational costs, as well as data and performance profiles. The reader is invited to consult their survey as a useful complement.

The present work is an attempt to propose a survey offering a panorama on all important aspects of performance indicators contrary to the previous surveys, addressed to the whole multiobjective optimization community. This work systematically analyzes 63 performance indicators by partitioning them into these four categories: Cardinality, Convergence, Distribution and spread, Convergence and distribution. The use of performance metrics targets four cases: comparison of algorithms, embedding of performance indicators into multiobjective methods, suggestion of stopping criteria for multiobjective optimization and identification of promising distribution-based performance indicators. Table 4.1 lists these indicators and their category, classifies them based on their properties, and indicates the section in which they are discussed. This work is organized as follows. Section 4.2 introduces the notations and definitions related to multiobjective optimization and performance indicators. Section 4.3 is the core of this work, and is devoted to the classification of the indicators according to their specific properties. Finally, Section 4.4 presents some applications.

Table 4.1 A summary of performance indicators. The 9 rightmost columns indicate references where the indicators are presented.

Category	**Performance indicators**	**Section**	[58]	[65]	[159]	[280]	[204]	[110]	[151]	[218]	[175]
Cardinality 4.3.1	C-metric/Two sets Coverage [279]	4.3.1	✓	✓	✓		✓		✓	✓	✓
	Error ratio [249]	4.3.1		✓	✓	✓	✓		✓		✓
	Generational non dominated vector generation [250]	4.3.1		✓			✓		✓	✓	
	Generational non dominated vector generation ratio [250]	4.3.1					✓		✓	✓	
	Mutual domination rate [187]	4.3.1								✓	
	Nondominated vector additional [250]	4.3.1		✓			✓		✓		
	Overall nondominated vector generation [249]	4.3.1		✓	✓	✓	✓	✓	✓	✓	✓
	Overall nondominated vector generation ratio [249]	4.3.1		✓	✓	✓	✓		✓	✓	✓
	Ratio of non-dominated points by the reference set [139]	4.3.1					✓		✓		✓
	Ratio of the reference points [139]	4.3.1					✓		✓		✓
Convergence 4.3.2	Degree of Approximation [93]	4.3.2									✓
	ϵ-family [280]	4.3.2				✓			✓	✓	✓
	Generational distance [249]	4.3.2		✓	✓	✓	✓	✓	✓	✓	✓
	γ-metric [84]	4.3.2	✓	✓			✓		✓	✓	
	Maximum Pareto front error [249]	4.3.2		✓	✓	✓	✓	✓	✓		✓
	$M_1^\star$-metric [276]	4.3.2		✓		✓	✓		✓	✓	✓
	Progression metric [249]	4.3.2		✓							
	Seven points average distance [226]	4.3.2					✓		✓		✓
	Standard deviation from the Generational distance [249]	4.3.2		✓							
Distribution and spread 4.3.3	Cluster [263]	4.3.3				✓	✓	✓	✓	✓	✓
	Δ-index [84]	4.3.3	✓				✓	✓		✓	✓
	Δ'-index [84]	4.3.3					✓		✓	✓	✓
	$\Delta^\star$ spread metric [272]	4.3.3						✓	✓	✓	✓
	Distribution metric [271]	4.3.3									
	Diversity comparison indicator [172]	4.3.3									✓
	Diversity indicator [56]	4.3.3									✓
	Entropy metric [108]	4.3.3					✓	✓	✓	✓	✓
	Evenness [189]	4.3.3						✓			✓
	Extension [188]	4.3.3						✓			✓
	Γ-metric [74]	4.3.3				✓					

Table 4.1 A summary of performance indicators. The 9 rightmost columns indicate references where the indicators are presented.

Category	Performance indicators	Section	[58]	[65]	[159]	[280]	[204]	[110]	[151]	[218]	[175]
	Hole Relative Size [64]	4.3.3		✓		✓		✓			✓
	Laumanns metric [166]	4.3.3		✓							
	Modified Diversity indicator [8]	4.3.3									✓
	M_2^*-metric [276]	4.3.3		✓		✓	✓	✓	✓		✓
	M_3^*-metric [276]	4.3.3	✓	✓		✓	✓	✓	✓	✓	✓
	Number of distinct choices [263]	4.3.3				✓	✓	✓	✓		✓
	Outer diameter [277]	4.3.3				✓					✓
	Overall Pareto Spread [263]	4.3.3				✓	✓	✓	✓	✓	✓
	Riesz S-Energy [140]	4.3.3									
	Sigma diversity metric [195]	4.3.3							✓		✓
	Spacing [226]	4.3.3		✓	✓	✓	✓	✓	✓	✓	✓
	U-measure [169]	4.3.3						✓			✓
	Uniform assessment metric [176]	4.3.3									✓
	Uniform distribution [239]	4.3.3					✓		✓		✓
	Uniformity [225]	4.3.3						✓			✓
	Uniformity [188]	4.3.3						✓			✓
Convergence and distribution 4.3.4	Averaged Hausdorff distance [227]	4.3.4							✓		✓
	Cone-based hypervolume [100]	4.3.4									
	Dominance move [174]	4.3.4									✓
	D-metric/Difference coverage of two sets [274]	4.3.4	✓			✓	✓		✓	✓	✓
	D_R-metric [78]	4.3.4			✓	✓	✓			✓	✓
	Hyperarea difference [263]	4.3.4					✓	✓	✓	✓	✓
	Hypervolume indicator (or S-metric) [276]	4.3.4	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Hypervolume Sharpe-ratio indicator [265]	4.3.4									
	Inverted generational distance [62]	4.3.4	✓						✓	✓	✓
	Inverted generation distance with non contributed solutions detection [243]	4.3.4									✓
	G-metric [184]	4.3.4									✓
	Logarithmic hypervolume indicator [123]	4.3.4									
	Modified inverted generational distance [147]	4.3.4									✓
	Performance comparison indicator [173]	4.3.4									✓
	p, q-averaged distance [251]	4.3.4									✓
	R-metric [139]	4.3.4				✓	✓	✓	✓	✓	✓

4.2 Notations and definitions

To apprehend performance indicators, the first part of this section describes the main concepts related to multiobjective optimization. The second part focuses on the theory of Pareto set approximations and performance indicators.

4.2.1 Multiobjective optimization and Pareto dominance

We consider the following continuous multiobjective optimization problem:

$$\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x}))^\top$$

where $\mathcal{X} \subseteq \mathbb{R}^n \neq \emptyset$ is called the *feasible set*, and $f_i : \mathbb{R}^n \to \mathbb{R}$ are m objective functions for $i = 1, 2, \ldots, m$, with $m \geq 2$. The image of the feasible set $\mathcal{Y} = \{f(x) \in \mathbb{R}^m : \mathbf{x} \in \mathcal{X}\}$

is called the *(feasible) objective set*. The sets $\mathbb{R}^n$ and $\mathbb{R}^m$ are respectively denoted as the *decision space* and the *objective space*.

To compare functions objective values, the following cone order relation is adopted [98].

Definition 23 (Dominance relations between objective vectors [79])**.** Given two objective vectors $\mathbf{y}^1$ and $\mathbf{y}^2$ in the objective space $\mathbb{R}^m$, we write:

- $\mathbf{y}^1 \leqq \mathbf{y}^2$ ($\mathbf{y}^1$ *weakly dominates* $\mathbf{y}^2$) if and only if $\mathbf{y}_i^1 \leq \mathbf{y}_i^2$ for all $i = 1, 2, \ldots, m$.
- $\mathbf{y}^1 \leq \mathbf{y}^2$ ($\mathbf{y}^1$ *dominates* $\mathbf{y}^2$) if and only if $\mathbf{y}^1 \leqq \mathbf{y}^2$ and $\mathbf{y}^1 \neq \mathbf{y}^2$.
- $\mathbf{y}^1 < \mathbf{y}^2$ ($\mathbf{y}^1$ *strictly dominates* $\mathbf{y}^2$) if and only if $\mathbf{y}_i^1 < \mathbf{y}_i^2$ for all $i = 1, 2, \ldots, m$.

In the case when neither $\mathbf{y}^1 \nleqq \mathbf{y}^2$ nor $\mathbf{y}^2 \nleqq \mathbf{y}^1$, $\mathbf{y}^1$ and $\mathbf{y}^2$ are said to be *incomparable*.

We can now present the concept of dominance relations for the decision vectors.

Definition 24 (Dominance relations for decision vectors [27])**.** Given two decision vectors $\mathbf{x}^1$ and $\mathbf{x}^2$ in the feasible set $\mathcal{X} \subseteq \mathbb{R}^n$, we write:

- $\mathbf{x}^1 \preceq \mathbf{x}^2$ ($\mathbf{x}^1$ weakly dominates x^2) if and only if $f(\mathbf{x}^1) \leqq f(\mathbf{x}^2)$.
- $\mathbf{x}^1 \prec \mathbf{x}^2$ ($\mathbf{x}^1$ dominates $\mathbf{x}^2$) if and only if $f(\mathbf{x}^1) \leq f(\mathbf{x}^2)$.
- $\mathbf{x}^1 \prec\prec \mathbf{x}^2$ ($\mathbf{x}^1$ strictly dominates $\mathbf{x}^2$) if and only if $f(\mathbf{x}^1) < f(\mathbf{x}^2)$.
- $\mathbf{x}^1 \sim \mathbf{x}^2$ ($\mathbf{x}^1$ and $\mathbf{x}^2$ are incomparable) if neither $\mathbf{x}^1$ weakly dominates $\mathbf{x}^2$ nor $\mathbf{x}^2$ weakly dominates $\mathbf{x}^1$.

With these relations, we now precise the concept of solution in the multiobjective optimization framework.

Definition 25 (Pareto optimality and Pareto solutions [98])**.** The vector $\mathbf{x} \in \mathcal{X}$ is a *Pareto-optimal* solution if there is no other vector in $\mathcal{X}$ that dominates it. The set of Pareto-optimal solutions is called the *Pareto set*, denoted $\mathcal{X}_P$, and the image of the Pareto set is called the *Pareto front*, denoted $\mathcal{Y}_P$.

In single-objective optimization, the set of optimal solutions is often composed of a singleton. In the multiobjective case, the Pareto front usually contains many elements (an infinity in continuous optimization and an exponential number in discrete optimization [98]). For a problem with m objectives, the Pareto front $\mathcal{Y}_P$ is at most of dimension $m - 1$. For three

objectives, the Pareto front is a surface, a curve, a point, or combinations of surfaces, curves and/or points, or the empty set. For two objectives, the Pareto front can be a curve or a point or a combination of curves and/or points, or the empty set. Also, it is interesting to define some bounds on this set.

Definition 26 (Ideal and nadir objective vectors)**.** The *ideal objective vector* $\mathbf{y}^I$ [98] is defined as the objective vector whose components are the solutions of each single-objective problem $\min_{\mathbf{x}\in\mathcal{X}} f_i(\mathbf{x}),\ i = 1, 2, \ldots, m$. The *nadir objective vector* $\mathbf{y}^N$ is defined as the objective vector whose components are the solutions of the single-objective problems $\max_{\mathbf{x}\in\mathcal{X}_P} f_i(\mathbf{x}), i = 1, 2, \ldots, m$.

For computational reasons, the nadir objective vector is often approximated by $\widetilde{\mathbf{y}}^N$ for which the coordinates are defined the following way: let $\mathbf{x}^{k,\star}$ be the solution of the single-objective problem $\min_{\mathbf{x}\in\mathcal{X}} f_k(\mathbf{x})$ for $k = 1, 2, \ldots, m$. The ith coordinate of $\widetilde{\mathbf{y}}^N$ is given by:

$$\widetilde{\mathbf{y}}_i^N = \max_{k=1,2,\ldots,m} f_i(\mathbf{x}^{k,\star}).$$

For a biobjective optimization problem, $\mathbf{y}^N$ equals $\widetilde{\mathbf{y}}^N$. It is not always the case when $m \geq 3$.

An illustration is given in Figure 4.1 where the Pareto front is piecewise continuous. To simplify the notation, continuous Pareto and piecewise continuous Pareto fronts will be respectively designed as continuous and discontinuous Pareto fronts.

Remark. In a multiobjective optimization problem, objectives are not necessarily contradictory, and the set of Pareto solutions may be a singleton. In this study, we assume that this is not the case.

4.2.2 Approximation sets and performance indicators

Generally, whether in the context of continuous or discrete optimization, it is not possible to find or enumerate all elements of the Pareto front. Hence to solve a multiobjective problem, one must look for the best discrete representation of the Pareto front. Evaluating the quality of a Pareto front approximation is not trivial. It itself involves several factors such as the closeness to the Pareto front and the coverage in the objective space. Indicators should capture these factors. To compare multiobjective optimization algorithms, the choice of a good performance indicator is crucial [159]. Hansen and Jaszkiewicz [139] are the first to introduce a mathematical framework to evaluate the performance of indicators according to the comparison of methods. In their work, they define what can be considered as a good measure to evaluate the quality of Pareto front. This work has been extended in [159,277,280].

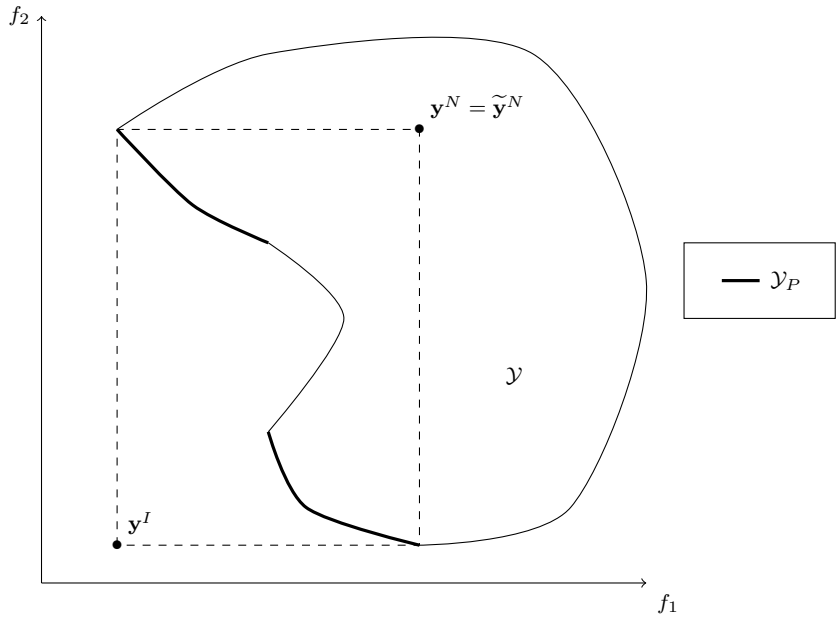


Figure 4.1 Objective space, Pareto front $\mathcal{Y}_P$, ideal objective vector $\mathbf{y}^I$ and nadir objective vector $\mathbf{y}^N$ (inspired by [98]).

We next define the notion of an approximation.

Definition 27 (Pareto set approximation [277])**.** A set of decision vectors X_N in the feasible set is called a *Pareto set approximation* if no element of this set is weakly dominated by any other. The image of such a set in the objective space is called a *Pareto front approximation* denoted $Y_N = f(X_N) \subseteq \mathbb{R}^m$. The set of all Pareto set approximations is denoted by Ψ and the set of all Pareto front approximations is denoted by Ω.

By definition, the Pareto front approximation corresponding to a given Pareto set approximation possesses only distinct elements, i.e., two different elements of the Pareto set approximation cannot map to the same point in the objective space. Consequently, for all $X_N \in \Psi$, $|X_N| = |Y_N|$.

Remark. We use the terms *Pareto set approximation* and *Pareto front approximation* in the remaining of the paper.

Zitzler et al. [280] propose an extension of the relation order for objective vectors to Pareto front approximations. They are summarized in Table 4.2. These relations orders can be naturally extended to Pareto set approximations.

Table 4.2 Comparison relations between Pareto front approximations [280].
Note that $Y_N^1 \prec\prec Y_N^2 \Longrightarrow Y_N^1 \prec Y_N^2 \Longrightarrow Y_N^1 \lhd Y_N^2 \Longrightarrow Y_N^1 \preceq Y_N^2$.

Relation	Objective vectors $\mathbf{y}^1$ and $\mathbf{y}^2$		Pareto front approximations Y_N^1 and Y_N^2	
Strictly dominates	$\mathbf{y}^1 < \mathbf{y}^2$	$\mathbf{y}^1$ is better than $\mathbf{y}^2$ in all objectives	$Y_N^1 \prec\prec Y_N^2$	Every $\mathbf{y}^2 \in Y_N^2$ is strictly dominated by at least one $\mathbf{y}^1 \in Y_N^1$
Dominates	$\mathbf{y}^1 \leq \mathbf{y}^2$	$\mathbf{y}^1$ is not worse than $\mathbf{y}^2$ in all objectives and better in at least one objective	$Y_N^1 \prec Y_N^2$	Every $\mathbf{y}^2 \in Y_N^2$ is dominated by at least one $\mathbf{y}^1 \in Y_N^1$
Weakly dominates	$\mathbf{y}^1 \leqq \mathbf{y}^2$	$\mathbf{y}^1$ is not worse than $\mathbf{y}^2$ in all objectives	$Y_N^1 \preceq Y_N^2$	Every $\mathbf{y}^2 \in Y_N^2$ is weakly dominated by at least one $\mathbf{y}^1 \in Y_N^1$
Is better			$Y_N^1 \lhd Y_N^2$	Every $\mathbf{y}^2 \in Y_N^2$ is weakly dominated by at least one $\mathbf{y}^1 \in Y_N^1$ and $Y_N^1 \neq Y_N^2$
Is incomparable		Neither $\mathbf{y}^1$ weakly dominates y^2 nor y^2 weakly dominates $\mathbf{y}^1$	$Y_N^1 \parallel Y_N^2$	Neither Y_N^1 weakly dominates Y_N^2 nor Y_N^1 weakly dominates Y_N^2

Measures are defined on Pareto front approximations. They are designed as quality indicators or performance indicators [280].

Definition 28 (Performance indicator [280])**.** A *k-ary* performance indicator is a function $I : \Omega^k \rightarrow \mathbb{R}$ which assigns to each collection $Y_N^1, Y_N^2, \ldots, Y_N^k$ of k Pareto front approximations a real value $I(Y_N^1, Y_N^2, \ldots, Y_N^k)$.

A performance indicator may consider several Pareto front approximations. The most common ones are mappings that take only one or two Pareto front approximations as arguments. They are known respectively as *unary* and *binary* performance indicators. With such a quality indicator, one can define a relation order between different Pareto front approximations. The interesting indicators are those that capture the Pareto dominance in the objective space.

Definition 29 (Monotonicity [277])**.** Assuming a greater indicator value is preferable, a performance indicator $I : \Omega \rightarrow \mathbb{R}$ is *monotonic* if and only if

$$\text{for all } Y_N^1, Y_N^2 \in \Omega,\ Y_N^1 \preceq Y_N^2 \implies I(Y_N^1) \geq I(Y_N^2).$$

Similarly, assuming a greater indicator value is preferable, a performance indicator $I : \Omega \rightarrow \mathbb{R}$ is *strictly monotonic* if and only if

$$\text{for all } Y_N^1, Y_N^2 \in \Omega,\ Y_N^1 \lhd Y_N^2 \implies I\left(Y_N^1\right) > I(Y_N^2).$$

Once the notion of performance indicator is defined, the definition of comparison method can be introduced.

Definition 30 (Comparison method [280])**.** Let $Y_N^1, Y_N^2 \in \Omega$ be two Pareto front approximations, $I = (I_1, I_2, \ldots, I_k)$ a combination of performance indicators and $E : \mathbb{R}^k \times \mathbb{R}^k \rightarrow \{true,\ false\}$ a Boolean function taking two vectors of size k as arguments. If all I_i for

$i = 1, 2, \ldots, k$ are unary, the *comparison method* $C_{I,E}(Y_N^1, Y_N^2)$ is defined as a Boolean function by the following formula:

$$C_{I,E}(Y_N^1, Y_N^2) = E\left(I(Y_N^1),\ I(Y_N^2)\right)$$

where for all $Y_N \in \Omega$, $I(Y_N) = \ (I_1(Y_N), I_2(Y_N), \ldots, I_k(Y_N))$.

If every I_i for $i = 1, 2, \ldots, k$ is binary, the comparison method $C_{I,E}(Y_N^1, Y_N^2)$ is defined as a Boolean function by

$$C_{I,E}(Y_N^1, Y_N^2) = E\left(I(Y_N^1, Y_N^2), I(Y_N^2, Y_N^1)\right)$$

where for all $Y_N^1,\ Y_N^2 \in \Omega$, $I(Y_N^1, Y_N^2) = (I_1(Y_N^1, Y_N^2),\ I_2(Y_N^1, Y_N^2), \ldots, I_k(Y_N^1, Y_N^2))$.

If I is composed of a single indicator I_0, we adopt the notation $C_{I_0,E}\left(Y_N^1, Y_N^2\right)$ instead of $C_{I,E}(Y_N^1, Y_N^2)$.

Informally, a comparison method is a true/false answer to: "Is a Pareto front approximation better than another one according to the combination of performance indicators I?" A simple comparison method is the following: given an unary performance indicator I and two Pareto front approximations $Y_N^1, Y_N^2 \in \Omega$,

> if the proposition $(C_{I,E}(Y_N^1,\ Y_N^2) = (I(Y_N^1) > I(Y_N^2)))$ is true, then Y_N^1 is said to be better than Y_N^2 according to the indicator I, assuming a greater indicator scalar value is preferable.

To compare several Pareto front approximations, one can be interested in defining comparison methods that capture the Pareto dominance in the objective space, i.e. given two Pareto front approximations $Y_N^1, Y_N^2 \in \Omega$ provided by Algorithms 1 and 2,

$$Y_N^1 \text{ weakly dominates/strictly dominates/is better than } Y_N^2 \Rightarrow (C_{I,E}(Y_N^1, Y_N^2) \text{ is true}).$$

More precisely, good comparison methods should always be compliant with the $\lhd$- relation [280]. On a given problem, Algorithm 1 should not be considered as less performant than Algorithm 2 if Y_N^1 is better than Y_N^2. The following definition summarizes these points:

Definition 31 (Compatibility and completeness [280])**.** Let $\mathcal{R}$ be an arbitrary binary relation on Pareto front approximations (typically, $\mathcal{R} \in \{\prec, \prec\prec, \preceq, \lhd\}$). The comparison method $C_{I,E}$ is denoted as $\mathcal{R}$-*compatible* if either for any Y_N^1, Y_N^2 Pareto front approximations, we have:

$$C_{I,E}(Y_N^1, Y_N^2) \Rightarrow Y_N^1\ \mathcal{R}\ Y_N^2$$

or for any Y_N^1, Y_N^2 Pareto front approximations, we have:

$$C_{I,E}(Y_N^1, Y_N^2) \Rightarrow Y_N^2 \ \mathcal{R} \ Y_N^1.$$

The comparison method is denoted as $\mathcal{R}$-*complete* if either for any Y_N^1, Y_N^2 Pareto front approximations, we have:

$$Y_N^1 \ \mathcal{R} \ Y_N^2 \Rightarrow \ C_{I,E}(Y_N^1, Y_N^2)$$

or for any Y_N^1, Y_N^2 Pareto front approximations, we have:

$$Y_N^2 \ \mathcal{R} \ Y_N^1 \Rightarrow C_{I,E}(Y_N^1, Y_N^2).$$

For any Pareto front approximations $Y_N^1, Y_N^2 \in \Omega$, there are no combination I of unary performance indicators such that $Y_N^1 \lhd Y_N^2 \ \Leftrightarrow \ C_{I,E}(Y_N^1, Y_N^2)$ [280].

The mathematical properties of the performance indicators mentioned in this survey are summarized in Tables 4.3, 4.4 and 4.5 in the appendices.

Remark. Throughout the rest of the paper, the following notations will be used. A discrete representation of the Pareto front is denoted by $Y_P \subseteq \mathcal{Y}_P$, called the *Pareto optimal solution set* [204]. The Pareto front approximation at iteration k will be denoted by $Y_N(k)$. In many cases, the Pareto front is unknown. The user needs to specify a set of objective vectors in the objective space, called a *reference set* and denoted by $Y_R \subseteq \mathbb{R}^m$. Note that a Pareto front (approximated or not) contains only feasible objective vectors, i.e. each element of a Pareto front approximation belongs to $\mathcal{Y}$. It implies that if an algorithm does not find any feasible points then $Y_N(k)$ is empty. For the following definitions to apply, we impose that the iteration counter k is set to 0 at the iteration where a first feasible point has been found.

4.3 A classification of performance indicators

We classify performance indicators into the four following groups [151, 204, 218]:

- *Cardinality indicators* 4.3.1: Quantify the number of non-dominated points generated by an algorithm.
- *Convergence indicators* 4.3.2: Quantify how close a set of non-dominated points is from the Pareto front in the objective space.

- *Distribution and spread indicators* 4.3.3: Quantify the distribution of a Pareto front approximation. Coverage measures how well every region of the objective space is represented. Spread focuses on the aspect that points should be far away from each other (typically this drives them to the boundary). The difference is discussed in [102, 155]. Extent refers to a more precise property, i.e., if the Pareto front approximation contains the extreme points of the Pareto front. Uniformity only considers how well the points are equally spaced [110, 225]. Spread and uniformity properties constitute the diversity of a Pareto front approximation [151].
- *Convergence and distribution indicators* 4.3.4: Capture both the properties of convergence and distribution.

4.3.1 Cardinality indicators

These indicators focus on the number of non-dominated points generated by a given algorithm. Some of them require the knowledge of the Pareto front.

Overall Non-dominated vector generation ($ONVG$) [249]

$ONVG$ returns the number of elements in the Pareto front approximation generated by the algorithm:

$$\text{for all } Y_N \in \Omega,\ ONVG(Y_N) = |Y_N|.$$

This indicator is to be maximized. Nonetheless, this is not a pertinent measure. For example, consider a Pareto front approximation Y_N^1 composed of one million non-dominated points and a Pareto front approximation Y_N^2 with only one point, such as this point dominates all the other points of Y_N^1. Y_N^1 outperforms Y_N^2 for this indicator but Y_N^2 is clearly better than Y_N^1 [159].

Overall Non-dominated vector generation ratio ($ONVGR$) [249]

$ONVGR$ represents the ratio of a number of elements in the Pareto front approximation with respect to a Pareto optimal solution set. Formally,

$$ONVGR(Y_N; Y_P) = \frac{|Y_N|}{|Y_P|}$$

where $|Y_P|$ is the cardinality of a Pareto optimal solution set and $|Y_N|$ the number of points of the Pareto front approximation. A higher value is considered to be better. Note that this

indicator is just $ONVG$ (4.3.1) divided by a scalar. Consequently, it suffers from the same drawbacks as the previous indicator.

Generational indicators ($GNVG$, $GNVGR$ and NVA) [249]

$GNVG(Y_N;k)$ (generational non-dominated vector generation) returns the number of non-dominated points $|Y_N(k)|$ generated at iteration k for a given iterative algorithm.
$GNVGR(Y_N;Y_P,k)$ (generational non-dominated vector generation ratio) is the ratio of non-dominated points $|Y_N(k)|$ generated at iteration k over the cardinality of Y_P where Y_P is a set of points from the Pareto front. $NVA(Y_N;k)$ (non-dominated vector addition) represents the variation of non-dominated points in the objective space generated between successive iterations. It is given by:

$$NVA\,(Y_N;k) = |Y_N(k)| - |Y_N(k-1)| \text{ for } k > 0.$$

These indicators can be used to follow the evolution of the generation of non-dominated points along iterations of a given algorithm. It seems difficult to use them as a stopping criterion as the number of non-dominated points can evolve drastically between two iterations.

Error ratio (ER) [249]

This indicator considers the number of non-dominated objective vectors which belong to the Pareto front. It is given by the following formula:

$$E(Y_N) = \frac{1}{|Y_N|} \sum_{\mathbf{y} \in Y_N} \mathbb{1}_{\mathbb{R}^m \setminus \mathcal{Y}_P}(\mathbf{y})$$

where for all $\mathbf{y} \in \mathbb{R}^m$,

$$\mathbb{1}_{\mathbb{R}^m \setminus \mathcal{Y}_P}(\mathbf{y}) = \begin{cases} 0 & \text{if } \mathbf{y} \text{ belongs to the Pareto front.} \\ 1 & \text{otherwise.} \end{cases}$$

The lower the indicator value, the better it is considered.

The author of [249] does not mention the presence of rounding errors in their indicator. A suggestion should be to consider an external accuracy parameter ϵ, quantifying the belonging of an element of the Pareto front approximation to the Pareto front with ϵ near to correct rounding errors.

This indicator requires the analytical expression of the Pareto front. Consequently, an user

can only use it on analytical benchmark tests. Moreover, this indicator depends mostly on the cardinality of the Pareto front approximation, which can misguide interpretations. [159] illustrate this drawback with the following example. Consider two Pareto front approximations. The first one has 100 elements, one in the Pareto front and the others close to it. Its error ratio is equal to 0.99. The second one has only two elements, one in the Pareto front, the other far from it. Its ratio is equal to 0.5. It is obvious that the first Pareto front approximation is better, even if its error ratio is bad. However, it is straightforward to compute.

Similarly to the error ratio measure, [139] defines the C_{1R} metric (called also *ratio of the reference points*). Given a reference set Y_R (chosen by the user) in the objective space, it is the ratio of the number of objectives vectors found in Y_R over the cardinality of the reference set Y_R. A higher value is desirable.

C-metric or coverage of two sets (C) [274]

Let Y_N^1 and Y_N^2 be two Pareto front approximations. The C-metric captures the proportion of points in a Pareto front approximation Y_N^2 weakly dominated by the Pareto front approximation Y_N^1. This binary performance indicator maps the ordered pair (Y_N^1, Y_N^2) to the interval $[0, 1]$ and is defined by:

$$C(Y_N^1, Y_N^2) = \frac{|\{\mathbf{y}^2 \in Y_N^2, \text{ there exists } \mathbf{y}^1 \in Y_N^1 \text{ such that } \mathbf{y}^1 \leqq \mathbf{y}^2\}|}{|Y_N^2|}.$$

If $C(Y_N^1, Y_N^2) = 1$, all the elements of Y_N^2 are dominated by (or equal to) the elements of Y_N^1. If $C(Y_N^1, Y_N^2) = 0$, no element of Y_N^2 is weakly dominated by an element of Y_N^1. Both orderings have to be computed, as $C(Y_N^1, Y_N^2)$ is not always equal to $1 - C(Y_N^2, Y_N^1)$.

Knowles et al. [159] point out the limits of this measure. If $C(Y_N^1, Y_N^2) \neq 1$ and if $C(Y_N^2, Y_N^1) \neq 1$, the two sets are incomparable. If the distribution of the sets or the cardinality is not the same, it gives some unreliable results. Moreover, it does not give an indication of "how much" a Pareto front approximation strictly dominates another.

Similarly to the C-metric, given a reference set Y_R in the objective space, the C_{2R} metric (*Ratio of non-dominated points by the reference set*) introduced in [139] is given by:

$$C_{2R}(Y_N; Y_R) = \frac{|\{\mathbf{y} \in Y_N : \text{ there does not exist } \mathbf{r} \in Y_R \text{ such that } \mathbf{r} \leq \mathbf{y}\}|}{|Y_N|}.$$

A higher C_{2R} value is considered to be better. This indicator has the same drawbacks as the C-metric.

Mutual domination rate (MDR) [187]

The authors of [187] use this performance indicator in combination with a Kalman filter to monitor the progress of evolutionary algorithms along iterations and thus providing a stopping criterion. At a given iteration k, MDR captures how many non-dominated points at iteration $k-1$ are dominated by non-dominated points generated at iteration k and reciprocally. Given two Pareto front approximations Y_N^1 and Y_N^2, the function $\Delta(Y_N^1,\ Y_N^2)$ returns the set of elements of Y_N^1 that are dominated by at least one element of Y_N^2. Formally,

$$MDR(Y_N;k) = \frac{|\Delta\left(Y_N(k-1),\ Y_N(k)\right)|}{|Y_N(k-1)|} - \frac{|\Delta\left(Y_N(k),\ Y_N(k-1)\right)|}{|Y_N(k)|}$$

where $Y_N(k)$ is the Pareto front approximation generated at iteration k. If $MDR(Y_N;k)=1$, the set of non-dominated points at iteration k totally dominates its predecessor at iteration $k-1$. If $MDR(Y_N;k)=0$, no significant progress has been observed. $MDR(Y_N;k)=-1$ is the worst case, as it results in a total loss of domination at the current iteration.

Cardinality indicators have a main drawback. They fail to quantify how well-distributed the Pareto front approximation is, or to quantify how it converges during the course of an algorithm.

4.3.2 Convergence indicators

Most of these measures require the knowledge of the Pareto Front to be evaluated. They capture the degree of proximity between a Pareto front and its approximation.

Generational distance (GD) [249]

The GD indicator captures the average distance between each element of a Pareto front approximation and its closest neighbor in a discrete representation of the Pareto front. This indicator is given by the following formula:

$$GD(Y_N;Y_P) = \frac{1}{|Y_N|}\left(\sum_{\mathbf{y}^1\in Y_N}\min_{\mathbf{y}^2\in Y_P}\|\mathbf{y}^1-\mathbf{y}^2\|^p\right)^{\frac{1}{p}}$$

where $|Y_N|$ is the number of points in a Pareto front approximation and $Y_P\subseteq\mathcal{Y}_P$ a discrete representation of the Pareto front. Generally, $p=2$. In this case, it is similar to the $M_1^\star$-measure defined in [276]. When $p=1$, it is equivalent to the γ-metric defined in [84]. For all

these indicators, a lower value is considered to be better.

GD is straightforward to compute but very sensitive to the number of points found by a given algorithm. In fact, if the algorithm identifies a single point in the Pareto front, the generational distance will equal 0. An algorithm can then miss an entire portion of the Pareto front without being penalized by this indicator. This measure favors algorithms returning a few non-dominated points close to the Pareto front versus those giving a more distributed representation of the Pareto front. As suggested by Colette and Siarry [65], it could be used as a stopping criteria. A slight variation of the generational distance $GD\left(Y_N(k), Y_N(k+1)\right)$ between two successive iterations, as long as the algorithm is running, could mean a convergence towards the Pareto front. It can be applied on continuous and discontinuous Pareto front approximations.

Standard deviation from the generational distance ($STDGD$) [249]

It measures the deformation of the Pareto front approximation according to a discrete representation of the Pareto front. It is given by the following formula:

$$STDGD(Y_N; Y_P) = \frac{1}{|Y_N|} \sum_{\mathbf{y}^1 \in Y_N} \left(\min_{\mathbf{y}^2 \in Y_P} \|\mathbf{y}^1 - \mathbf{y}^2\| - GD(Y_N; Y_P) \right)^2.$$

The same critics as with the generational distance apply.

Seven points average distance ($SPAD$) [226]

As it is not practical to obtain the Pareto front, an alternative is to use a reference set Y_R in the objective space. The $SPAD$ indicator defined for biobjective optimization problems uses a reference set composed of seven points [1]:

$$Y_R = \{(0,0)\} \cup \left\{ \left(\frac{j}{3} \max_{\mathbf{x} \in \mathcal{X}} f_1(\mathbf{x}), 0 \right)_{1 \leq j \leq 3} \right\} \cup \left\{ \left(0, \ \frac{l}{3} \max_{\mathbf{x} \in \mathcal{X}} f_2(\mathbf{x}) \right)_{1 \leq l \leq 3} \right\}.$$

$SPAD$ captures the average distance of the elements of the reference set Y_R to their closest neighbor in the Pareto front approximation. Formally,

$$SPAD(Y_N; Y_R) = \frac{1}{|Y_R|} \sum_{\mathbf{r} \in Y_R} \min_{\mathbf{y} \in Y_N} \|\mathbf{y} - \mathbf{r}\|.$$

[1] The formula written in the article is wrong. It has been corrected.

A lower value is considered to be better.

This indicator raises same critics as above. Note that the computational cost to solve the single-objective problems $\max_{\mathbf{x}\in\mathcal{X}} f_i(\mathbf{x})$ for $i = 1, 2$ is not negligible. Also, the points in the reference set can fail to capture the whole form of the Pareto front. Its limitation to two objectives is also an inconvenient. Nonetheless, it does not require the knowledge of Pareto front.

Maximum Pareto front error ($MPFE$) [249]

This indicator, defined in [249], is another measure that evaluates the distance between a discrete representation of the Pareto front and the Pareto front approximation obtained by a given algorithm. It corresponds to the largest minimal distance between elements of the Pareto front approximation and their closest neighbors belonging to the Pareto front. This indicator is to be minimized. It is expressed with the following formula (generally, $p = 2$):

$$MPFE(Y_N; Y_P) = \max_{\mathbf{y}^1\in Y_N} \left(\min_{\mathbf{y}^2\in Y_P} \sum_{i=1}^{m} \left| \mathbf{y}_i^1 - \mathbf{y}_i^2 \right|^p \right)^{\frac{1}{p}}.$$

When $Y_N \subseteq Y_P$, the value $MPFE(Y_N; Y_P)$ is zero. The indicator is not relevant, as pointed out in [159]. Consider two Pareto fronts approximations in which the first possesses only one element in the Pareto front and the second has ten elements: nine of them belong to the Pareto front and one is at some positive distance from it. As $MPFE$ considers only largest minimal distances, it favors the first Pareto front approximation. But the second is clearly better. However, it is straightforward and cheap to compute and may be used on continuous and discontinuous problems.

Progress metric (P_g) [249]

This indicator introduced in [34] measures the progression of the Pareto front approximation given by an algorithm towards the Pareto front in function of the number of iterations for a given objective function i. It is defined by:

$$P_g(Y_N; i, k) = \ln \sqrt{\frac{\min\limits_{\mathbf{y}\in Y_N(0)} \mathbf{y}_i}{\min\limits_{\mathbf{y}\in Y_N(k)} \mathbf{y}_i}}.$$

The author of [249] modifies this metric to take into account whole Pareto front approximations:

$$RP_g(Y_N; Y_P, k) = \ln \sqrt{\frac{GD(Y_N(0); Y_P)}{GD(Y_N(k); Y_P)}}$$

where $GD(Y_N(k); Y_P)$ is the generational distance (4.3.2) of the Pareto front approximation Y_N at iteration k.

P_g is not always defined, for example when values of $\min_{\mathbf{y} \in Y_N(0)} \mathbf{y}_i$ or $\min_{\mathbf{y} \in Y_N(k)} \mathbf{y}_i$ are negative or null. If GD is positive[2], RP_g is well defined, but it requires the knowledge of the Pareto front.

P_g, when it exists, provides an estimation of the speed of convergence of the associated algorithm. RP_g captures only the variations of the generational distance along the number of iterations. The drawbacks of the generational distance do not apply in this case. Finally, a bad measure of progression does not necessarily mean that an algorithm performs poorly. Some methods less deeply explore the objective space, but reach the Pareto front after a more important number of iterations.

ϵ-indicator (I_ϵ) [280]

An objective vector $\mathbf{y}^1 \in \mathbb{R}^m$ is ϵ-dominating, for $\epsilon > 0$, an objective vector $\mathbf{y}^2 \in \mathbb{R}^m$ if:

$$\text{for all } i = 1, 2, \ldots, m, \quad \mathbf{y}_i^1 \leq \epsilon\, \mathbf{y}_i^2.$$

The multiplicative ϵ-indicator for two Pareto front approximations Y_N^1 and Y_N^2 is defined as the minimum factor one has to multiply a Pareto front approximation to make it weakly dominate another one. It is given by

$$I_{\epsilon_\times}(Y_N^1, Y_N^2) = \inf_{\epsilon > 0} \left\{ \mathbf{y}^2 \in Y_N^2 :\ \exists \mathbf{y}^1 \in Y_N^1 \text{ such that } \mathbf{y}^1 \text{ is } \epsilon\text{-dominating } \mathbf{y}^2 \right\}.$$

It can be calculated the following way:

$$I_{\epsilon_\times}(Y_N^1, Y_N^2) = \max_{\mathbf{y}^2 \in Y_N^2} \min_{\mathbf{y}^1 \in Y_N^1} \max_{1 \leq i \leq m} \frac{\mathbf{y}_i^1}{\mathbf{y}_i^2}.$$

Given a discrete representation of the Pareto front Y_P, the unary metric $I_{\epsilon_\times}(Y_N; Y_P)$ (with a semicolon) is defined as $I_{\epsilon_\times}(Y_P, Y_N)$.

Similarly, Zitzler et al. [280] define an additive ϵ-indicator based on the following additive

[2]This first part of the sentence has been modified according to the article.

ϵ-domination. It is said that an objective vector $\mathbf{y}^1$ is additively ϵ-dominating an objective vector $\mathbf{y}^2$ for $\epsilon > 0$ if for all $i = 1, 2, \ldots, m$, $\mathbf{y}_i^1 \leq \epsilon + \mathbf{y}_i^2$. This indicator is then calculated by:

$$I_{\epsilon_+}(Y_N^1, Y_N^2) = \max_{\mathbf{y}^2 \in Y_N^2} \min_{\mathbf{y}^1 \in Y_N^1} \max_{1 \leq i \leq m} \mathbf{y}_i^1 - \mathbf{y}_i^2.$$

A value inferior to 1 (respectively 0) for the binary multiplicative ϵ-indicator (respectively additive ϵ-indicator) indicates that Y_N^1 weakly dominates Y_N^2.

Binary additive and multiplicative ϵ-indicators possess desirable properties. They are Pareto compliant and compatible [280], do not require the knowledge of the Pareto front, and represent natural extensions for approximation schemes in optimization theory. However, the main problem with the ϵ-indicator is that its evaluation value involves only one particular element in each Pareto front approximation, which can misguide quality comparison between different Pareto front approximations. Furthermore, the ϵ-indicator focuses only on one objective when comparing different objective vectors, as noticed in [174]. For example, consider $\mathbf{y}^1 = (0, 1, 1)$ and $\mathbf{y}^2 = (1, 0, 0)$ in a tri-objective maximization problem. Although $\mathbf{y}^1$ performs better than $\mathbf{y}^2$ in two different objectives, the additive ϵ-indicators are identical:

$$I_{\epsilon_+}\left(\{\mathbf{y}^1\}, \{\mathbf{y}^2\}\right) = I_{\epsilon_+}\left(\{\mathbf{y}^2\}, \{\mathbf{y}^1\}\right).$$

On the contrary, it is straightforward to compute. It can be used for continuous and discontinuous approximations of Pareto fronts.

Degree of approximation (*DOA*) [93]

By taking into account the dominance relation in the objective space, DOA captures an average degree of proximity from a discrete representation of the Pareto front to a Pareto front approximation. A lower value is considered to be better. This indicator is proved to be $\prec$-complete (see Definition 31). It aims to compare algorithms when the Pareto fronts are known.

Given $\mathbf{y}^2$ an objective vector belonging to Y_P, the set $\mathcal{D}(\mathbf{y}^2, Y_N)$ in the objective space is defined as the subset of points belonging to the Pareto front approximation Y_N dominated by the objective vector $\mathbf{y}^2$. The minimum Euclidean distance between $\mathbf{y}^2 \in Y_P$ and $\mathcal{D}(\mathbf{y}^2, Y_N)$ (which may be empty) is computed with

$$d(\mathbf{y}^2, Y_N) = \begin{cases} \min\limits_{\mathbf{y}^1 \in \mathcal{D}(\mathbf{y}^2, Y_N)} \|\mathbf{y}^2 - \mathbf{y}^1\| & \text{if } |\mathcal{D}(\mathbf{y}^2, Y_N)| > 0 \\ \infty & \text{if } |\mathcal{D}(\mathbf{y}^2, Y_N)| = 0. \end{cases}$$

Similarly, $d^+\left(\mathbf{y}^2, Y_N \setminus \mathcal{D}(\mathbf{y}^2, Y_N)\right)$ is defined for $\mathbf{y}^2 \in Y_P$ by considering the set of points of Y_N that do not belong to $\mathcal{D}(\mathbf{y}^2, Y_N)$ as:

$$d^+\left(\mathbf{y}^2, Y_N \setminus \mathcal{D}(\mathbf{y}^2, Y_N)\right) = \begin{cases} \min\limits_{\mathbf{y}^1 \in Y_N \setminus \mathcal{D}(\mathbf{y}^2, Y_N)} \|(\mathbf{y}^1 - \mathbf{y}^2)_+\| & \text{if } |Y_N \setminus \mathcal{D}(\mathbf{y}^2, Y_N)| > 0 \\ \infty & \text{if } |Y_N \setminus \mathcal{D}(\mathbf{y}^2, Y_N)| = 0 \end{cases}$$

where $(\mathbf{y}^1 - \mathbf{y}^2)_+ = (\max(0, \mathbf{y}_i^1 - \mathbf{y}_i^2))_{i=1,2,\ldots,m}$.

The DOA indicator is finally given by

$$DOA(Y_N; Y_P) = \frac{1}{|Y_P|} \sum_{\mathbf{y}^2 \in Y_P} \min\left\{d(\mathbf{y}^2, Y_N), d^+\left(\mathbf{y}^2, Y_N \setminus \mathcal{D}(\mathbf{y}^2, Y_N)\right)\right\}.$$

The value of DOA does not depend on the number of points of Y_P, i.e. if $|Y_P| \gg |Y_N|$ [93]. In fact, this indicator partitions Y_N into subsets in which each element is dominated by a point $\mathbf{y} \in Y_P$. Its computational cost is quite low (in $\mathcal{O}(m\ |Y_N| \times |Y_P|)$). It can be used for discontinuous and continuous approximations of Pareto fronts.

4.3.3 Distribution and spread indicators

According to [74], "*the spread metrics try to measure the extents of the spread achieved in a computed Pareto front approximation*". They are not really useful to **evaluate the convergence of an algorithm, or at comparing algorithms**, but rather the distribution of the points along Pareto front approximations. They only make sense when the Pareto set is composed of several solutions corresponding to distinct objective vectors.

Spacing (SP) [226]

The SP indicator captures the variation of the distance between elements of a Pareto front approximation. A lower value is considered to be better. This indicator is computed with

$$SP(Y_N) = \sqrt{\frac{1}{|Y_N| - 1} \sum_{j=1}^{|Y_N|} \left(\bar{d} - d^1\left(\mathbf{y}^j,\ Y_N \setminus \{\mathbf{y}^j\}\right)\right)^2}$$

where $d^1\left(\mathbf{y}^j,\ Y_N \setminus \{\mathbf{y}^j\}\right) = \min_{\mathbf{y} \in Y_N \setminus \{\mathbf{y}^j\}} \|\mathbf{y} - \mathbf{y}^j\|_1$ is the l_1 distance of $\mathbf{y}^j \in Y_N$ to the set $Y_N \setminus \{\mathbf{y}^j\}$ and $\bar{d}$ is the mean of all $d^1(\mathbf{y}^j, Y_N \setminus \{\mathbf{y}^j\})$ for $j = 1, 2, \ldots, |Y_N|$.

This method cannot account for holes in the Pareto front approximation as it takes into account the distance between an objective vector and its closest neighbor. The major issue

with this indicator is it gives some limited information when points given by the algorithm are clearly separated, but spread into multiple groups. On the contrary, it is straightforward to compute.

Delta indexes (Δ', Δ and $\Delta^\star$) [84, 272]

Deb et al. [84] introduce the Δ' index for biobjective problems, which captures the variation of distance between consecutive elements of the Pareto front approximation into the biobjective space. Formally,

$$\Delta'(Y_N) = \sum_{j=1}^{|Y_N|-1} \frac{\left| d^c\left(\mathbf{y}^j, Y_N \setminus \{\mathbf{y}^j\}\right) - \bar{d^c} \right|}{|Y_N| - 1}$$

where $d^c\left(\mathbf{y}^j, Y_N \setminus \{\mathbf{y}^j\}\right)$ is the Euclidean distance between consecutive elements of the Pareto front approximation Y_N , and $\bar{d^c}$ the mean of the $d^c\left(\mathbf{y}^j, Y_N \setminus \{\mathbf{y}^j\}\right)$ for $j = 1, 2, \ldots, |Y_N| - 1$. As this indicator considers Euclidean distances between consecutive objective vectors, it can be misleading if the Pareto front approximation is piecewise continuous. The Δ' index does not generalize to more than 2 objectives, as it uses lexicographic order in the biobjective objective space to compute the $d^c\left(\mathbf{y}^j, Y_N \setminus \{\mathbf{y}^j\}\right)$. In addition, it does not consider the extent of the Pareto front approximation, i.e. distances between extreme points of the Pareto front.

The Δ index is an indicator derived from the Δ' index to take into account the extent of the Pareto front approximation for biobjective problems:

$$\Delta(Y_N; Y_P) = \frac{\sum_{i=1}^{2} \min_{y \in Y_N} \|\mathbf{y}^{i,\star} - \mathbf{y}\| + \sum_{j=1}^{|Y_N|-1} \left| d^c\left(\mathbf{y}^j, Y_N \setminus \{\mathbf{y}^j\}\right) - \bar{d^c} \right|}{\sum_{k=1}^{2} \min_{\mathbf{y} \in Y_N} \|\mathbf{y}^{k,\star} - \mathbf{y}\| + \left(|Y_N| - 1\right) \bar{d^c}}$$

where $\min_{\mathbf{y} \in Y_N} \|\mathbf{y}^{i,\star} - \mathbf{y}\|$ for $i = 1, 2$ are the Euclidean distances between the extreme solutions of the Pareto front (i.e. $\mathbf{y}^{i,\star} = f(\mathbf{x}^{i,\star}) \in Y_P$ where $\mathbf{x}^{i,\star}$ is solution to the i-th single-objective problem) and the boundary solutions of the Pareto front approximation. The other notations remain the same as before. This indicator requires the resolution of each single-objective optimization problem. This indicator is extended to Pareto fronts with more than two objectives by [272] to the generalized $\Delta^\star$-index:

$$\Delta^\star(Y_N; Y_P) = \frac{\sum_{i=1}^{m} d^2(\mathbf{y}^{i,\star}, Y_N) + \sum_{j=1}^{|Y_N|} \left| d^2(\mathbf{y}^j, Y_N \setminus \{\mathbf{y}^j\}) - \bar{d^2} \right|}{\sum_{i=1}^{m} d^2(\mathbf{y}^{i,\star}, Y_N) + |Y_N| \;\; \bar{d^2}}$$

where $d^2(\mathbf{y}^{i,\star}, Y_N) = \min_{\mathbf{y}\in Y_N} \|y^{i,\star} - y\|$ with $y^{i,\star} = F(x^{i,\star}) \in Y_P$ the extreme objective vector corresponding to $\mathbf{x}^{i,\star}$ solution to the i-th single-objective problem and $d^2(\mathbf{y}^j, Y_N \setminus \{\mathbf{y}^j\}) = \min_{\mathbf{y}\in Y_N\setminus\{\mathbf{y}^j\}} \|\mathbf{y}^j - \mathbf{y}\|$ the minimal Euclidean distance between two points of the Pareto front approximation. $\bar{d^2}$ is the mean of all $d^2(\mathbf{y}^j, Y_N \setminus \{\mathbf{y}^j\})$ for $j = 1, 2, \ldots, |Y_N|$. As it considers the shortest distances between elements of the Pareto front approximation, the $\Delta^\star$ index suffers from the same drawbacks as the spacing metric. Moreover, it requires the knowledge of the extreme solutions of the Pareto front.

For these three indicators, a lower value is considered to be better.

Two measures proposed by [74] (Γ and Δ)

Given a Pareto set approximation $X_N = \{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^N\}$ to which two additional , "extreme" points indexed by 0 and $N+1$ are added, for each objective i for $i = 1, 2, \ldots, m$, elements $\mathbf{x}^j$ for $j = 0, 1, \ldots, N+1$ of the Pareto set approximation are sorted so that,

$$f_i(\mathbf{x}^0) \leq f_i(\mathbf{x}^1) \leq f_i(\mathbf{x}^2) \leq \ldots \leq f_i(\mathbf{x}^{N+1}).$$

Custódio et al. [74] introduce the following metric $\Gamma > 0$ defined by:

$$\Gamma(Y_N) = \max_{i\in\{1,2,\ldots,m\}} \max_{j\in\{0,1,\ldots,N\}} \delta_{i,j}$$

where $\delta_{i,j} = f_i(\mathbf{x}^{j+1}) - f_i(\mathbf{x}^j)$ and $Y_N = f(X_N)$. When considering a biobjective problem ($m = 2$), the metric reduces to consider the maximum distance in the infinity norm between two consecutive points in the Pareto front approximation as it is shown in Figure 4.2.

To take into account the extent of the Pareto front approximation, the authors of [74] define the following indicator by

$$\Delta(Y_N) = \max_{i\in\{1,2,\ldots,m\}} \left\{ \frac{\delta_{i,0} + \delta_{i,N} + \sum_{j=1}^{N-1} \left|\delta_{i,j} - \overline{\delta_i}\right|}{\delta_{i,0} + \delta_{i,N} + (N-1)\overline{\delta_i}} \right\}$$

where $\overline{\delta_i}$, for $i = 1, 2, \ldots, m$, is the mean of all distances $\delta_{i,j}$ for $j = 1, 2 \ldots, N-1$.

The Γ and Δ indicators do not use the closest distance between two points in the objective space. Consequently, they do not have the same drawbacks as the spacing metric. However, the $\delta_{i,j}$ distance captures holes in the Pareto front if this one is piecewise discontinuous. For these two indicators, a lower value is desirable. These two metrics are more adapted to continuous Pareto front approximations.

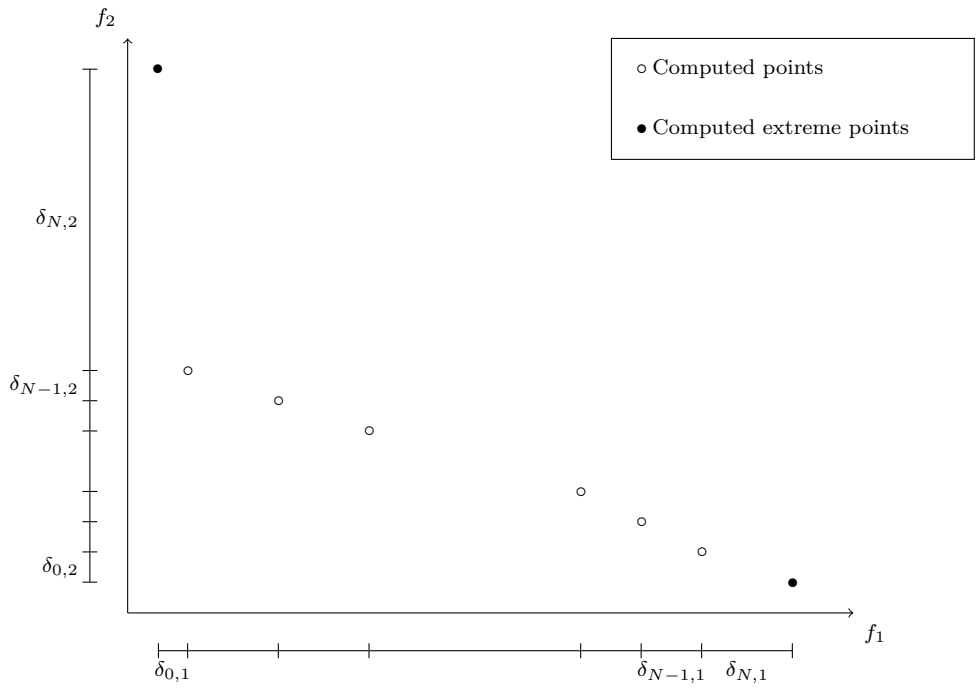


Figure 4.2 Illustration of the Γ metric for a biobjective problem (inspired by [74]).

Remark. The authors of [74] suggest two ways to compute extreme points $\mathbf{x}^0$ and $\mathbf{x}^{N+1}$. For benchmark tests, the Pareto front is known and extreme points correspond to the ones of the Pareto set. Otherwise, the Γ and Δ indicators use the extreme points of the Pareto set approximation X_N.

Hole relative size (HRS) [64]

This indicator identifies the largest hole in a Pareto front approximation for a biobjective problem. It is given by

$$HRS(Y_N) = (1/\bar{d}) \max_{j=1,2,\ldots,|Y_N|-1} d^j$$

where Y_N is a Pareto front approximation whose elements have been sorted in ascendant order according to the first objective, $d^j = \|\mathbf{y}^j - \mathbf{y}^{j+1}\|_2$ is the l_2 distance between the two adjacent objective vectors $\mathbf{y}^j \in Y_N$ and $\mathbf{y}^{j+1} \in Y_N$ and $\bar{d}$ the mean of all d^j for $j = 1, 2, \ldots, |Y_N| - 1$.

A lower indicator value is desirable. As it takes into account holes in the objective space, this indicator is more adapted to continuous Pareto front approximations.

Zitzler's metrics $M_2^\star$ and $M_3^\star$ [276]

The $M_2^\star$ metric returns a value in the interval $[0; |Y_N|]$ where Y_N is the Pareto front approximation. It reflects the number of subsets of the Pareto front approximation Y_N of a certain radius (σ). A higher value is considered to be better. Its expression is given by

$$M_2^\star(Y_N; \sigma) = \frac{1}{|Y_N| - 1} \sum_{\mathbf{y}^2 \in Y_N} |\{\mathbf{y}^1 \in Y_N, \|\mathbf{y}^2 - \mathbf{y}^1\| > \sigma\}|.$$

If $M_2^\star(Y_N; \sigma) = |Y_N|$, it means that for each objective vector, no other objective vector within the distance σ can be found. It is straightforward to compute but it can be difficult to interpret.

The authors of [239] introduce the Uniform distribution indicator, based too on the search of niches of size σ, given by

$$UD(Y_N; \sigma) = \frac{1}{1 + D_{nc}(Y_N, \sigma)}$$

where $D_{nc}(Y_N, \sigma)$ is the standard deviation of the number of niches around all the points of the Pareto front approximation Y_N defined as

$$D_{nc}(Y_N, \sigma) = \sqrt{\frac{1}{|Y_N| - 1} \left(\sum_{j=1}^{|Y_N|} \left(nc(\mathbf{y}^j, \sigma) - \frac{1}{|Y_N|} \sum_{l=1}^{|Y_N|} nc(\mathbf{y}^l, \sigma) \right)^2 \right)}$$

with $nc(\mathbf{y}^j, \sigma) = |\{\mathbf{y} \in Y_N, \|\mathbf{y} - \mathbf{y}^j\| < \sigma\}| - 1$. The UD indicator is to be minimized.

Finally, the $M_3^\star$ metric defined by Zitzler [276], considers the extent of the front:

$$M_3^\star(Y_N) = \sqrt{\sum_{i=1}^{m} \max_{j \in \{1,2,\ldots,|Y_N|\}} \max_{\mathbf{y} \in Y_N \setminus \{y^j\}} \|\mathbf{y}^j - \mathbf{y}\|}.$$

A higher value is considered to be better. The $M_3^\star$ metric only takes into account the extremal points of the computed Pareto front approximation. Consequently, it is sufficient for two different algorithms to have the same extremal points to be considered as equivalent according to this metric. It can be used on continuous and discontinuous approximations of Pareto fronts as it only gives information on the extent of the Pareto front.

Uniformity (δ) [225]

This is the minimal distance between two points of the Pareto front approximation. This measure is straightforward to compute and easy to understand. However, it does not really provide pertinent information on the distribution of the points along the Pareto front approximation.

Evenness (ξ) [189]

The ξ-evenness indicator captures the uniformity quality of a Pareto front approximation by integrating distance values between its elements into a coefficient of variation. More specifically, two scalar values are associated to each element $\mathbf{y} \in Y_N$ of the Pareto front approximation. $d^l(\mathbf{y}, Y_N \setminus \{\mathbf{y}\})$ is the minimum Euclidean distance between objective vector $\mathbf{y}$ and its closest neighbor in the objective space. $d^u(\mathbf{y}, Y_N \setminus \{\mathbf{y}\})$ is the maximal Euclidean distance between an element $\mathbf{y} \in Y_N$ and another element of Y_N such that no other point of Y_N is within the (hyper)sphere of diameter $d^u(\mathbf{y}, Y_N \setminus \{\mathbf{y}\})$. ξ is then defined as

$$\xi(Y_N) = \frac{\sigma_D}{\widehat{D}}$$

where σ_D and $\widehat{D}$ are respectively the standard deviation and the mean of the set of minimum distances $d^l(\mathbf{y}, Y_N \setminus \{\mathbf{y}\})$ and maximum diameters $d^u(\mathbf{y}, Y_N \setminus \{\mathbf{y}\})$ for each element $\mathbf{y}$ of Y_N. The closest ξ is to 0, the better the uniformity is.

It can be considered as a coefficient of variation. It is straightforward to compute. In the case of continuous Pareto front, it can account for holes in the Pareto front approximation.

Reference [128] also defines the evenness as

$$E(Y_N) = \frac{\max\limits_{\mathbf{y}^1 \in Y_N} \min\limits_{\mathbf{y}^2 \in Y_N \setminus \{\mathbf{y}^1\}} \|\mathbf{y}^1 - \mathbf{y}^2\|}{\min\limits_{\mathbf{y}^1 \in Y_N} \min\limits_{\mathbf{y}^2 \in Y_N \setminus \{\mathbf{y}^1\}} \|\mathbf{y}^1 - \mathbf{y}^2\|}.$$

The lower the value, the better the distribution with a lower bound $E(Y_N) = 1$.

Binary uniformity (SP_l) [188]

Contrary to others indicators, this indicator aims to compare the uniformity of two Pareto front approximations. This indicator is inspired by the wavelet theory.

Let Y_N^1 and Y_N^2 be two Pareto front approximations. The algorithm is decomposed in several steps:

Set $l = 1$.

1. For each element $\mathbf{y}^1 \in Y_N^1$ and $\mathbf{y}^2 \in Y_N^2$, compute the respective distances to their closest neighbor $d_l^2(\mathbf{y}^1, Y_N^1 \setminus \{\mathbf{y}^1\})$ and $d_l^2(\mathbf{y}^2, Y_N^2 \setminus \{\mathbf{y}^2\})$ given by

$$d_l^2(\mathbf{y}, Y_N \setminus \{\mathbf{y}\}) = \min_{\mathbf{y}^v \in Y_N \setminus \{\mathbf{y}\}} \|\mathbf{y} - \mathbf{y}^v\|.$$

2. For both sets, compute the average distances $\overline{d_l^2}(Y_N^1)$ and $\overline{d_l^2}(Y_N^2)$ between neighbor points given by

$$\overline{d_l^2}(Y_N) = \frac{1}{|Y_N|} \sum_{\mathbf{y} \in Y_N} d_l^2(\mathbf{y}, Y_N \setminus \{\mathbf{y}\}).$$

3. For each set, compute the spacing measures $SP^l(Y_N^1)$ and $SP^l(Y_N^2)$ given by

$$SP^l(Y_N) = \sqrt{\sum_{\mathbf{y} \in Y_N} \frac{\left(1 - \psi\left(d_l^2(\mathbf{y}, Y_N \setminus \{\mathbf{y}\}), \overline{d_l^2}(Y_N)\right)\right)^2}{|Y_N| - 1}}$$

   with $\psi(a, b) = \begin{cases} \frac{a}{b} & \text{if } a > b \\ \frac{b}{a} & \text{otherwise.} \end{cases}$

4. If $SP^l(Y_N^1) < SP^l(Y_N^2)$, then Y_N^1 has better uniformity than Y_N^2 and reciprocally. If $SP^l(Y_N^1) = SP^l(Y_N^2)$ and $l \geq \min(|Y_N^1| - 1, |Y_N^2| - 1)$ then Y_N^1 has the same uniformity as Y_N^2. Else if $SP^l(Y_N^1) = SP^l(Y_N^2)$ and $l < \min(|Y_N^1| - 1, |Y_N^2| - 1)$, then increment l by 1, and recompute the previous steps by removing the smallest distance $d_l^2(\mathbf{y}^1, Y_N^1 \setminus \{\mathbf{y}^1\})$ and $d_l^2(\mathbf{y}^2, Y_N^2 \setminus \{\mathbf{y}^2\})$ until the end.

The value of the binary uniformity indicator is difficult to interpret but can be computed easily. It does not take into account the extreme points of the Pareto front.

U-measure (U) [169]

This indicator measures the uniformity of a Pareto front approximation based on distance between its elements according to each objective. For each objective vector in the Pareto front approximation Y_N, Euclidean distance to their nearest neighbors with respect to each objective axis is determined, as well as distances of reference objectives (objective vectors corresponding to solutions of the single-objective optimization problems or determined by the user) to their nearest neighbors. Let χ be the set of distances between nearest neighbors and $\bar{\chi}^0$ the set of distances between reference objective vectors and their closest neighbor.

Small variability in the set χ would reflect uniform distribution, and values close to 0 for the set $\bar{\chi}^0$ would reflect good coverage properties. For ease of computation, each distance in $\bar{\chi}^0$ is summed up with the average value of the distances in χ, resulting in the new set $\bar{\chi}$. The U-measure captures the discrepancy among the scalar elements of the set $\bar{\chi}$ and is given by

$$U(Y_N) = \frac{1}{|\bar{\chi}|} \sum_{d' \in \bar{\chi}} \left| \frac{d'}{d_{\text{ideal}}} - 1 \right|$$

where $d_{\text{ideal}} = \dfrac{1}{|\bar{\chi}|} \displaystyle\sum_{d' \in \bar{\chi}} d'$.

$\dfrac{d'}{d_{\text{ideal}}} - 1$ can be interpreted as the percentage deviation from the ideal distance if it is multiplied by 100%. The U-measure is then the mean of this ratio along all elements of the set $\bar{\chi}$. A small U indicator value can be interpreted as a better uniformity.

It attempts to quantify the uniformity of found points along the Pareto front approximation.

The same problems as for the previous indicators can be raised. Especially, the formula works only if there are several points. Moreover, this indicator can take time to compute when computing the minimal distances. As for the spacing metric (4.3.3), this last one does not account for holes in the Pareto front approximation as it takes only into account closest neighbors. It is then more pertinent on continuous Pareto front approximations.

Overall Pareto spread (OS) [263]

This indicator only captures the extent of the front covered by the Pareto front approximation. The larger the better it is. It is given by

$$OS(Y_N) = \prod_{i=1}^{m} \frac{\left| \max\limits_{\mathbf{y} \in Y_N} \mathbf{y}_i - \min\limits_{\mathbf{y} \in Y_N} \mathbf{y}_i \right|}{\left| \widetilde{\mathbf{y}}_i^I - \widetilde{\mathbf{y}}_i^M \right|}$$

where $\widetilde{\mathbf{y}}^M$ is an approximation of the maximum objective vector (objective vector composed of the maxima of each single-objective optimization problem assuming they exist) or the maximum objective vector and $\widetilde{y}^I$ an approximation of the ideal objective vector or the ideal objective vector.

This is an indicator for which the values are among the values 0 and 1. The maximum and ideal objective vectors need to be computed for more precise results, resulting in 2 m single-objective problems to solve. The indicator does not take into account the distribution of points along the Pareto front approximation.

Outer diameter (I_{OD}) [277]

Analogously to the overall Pareto spread metric, the outer diameter indicator returns the maximum distance along all objective dimensions pondered by weights $\mathbf{w} \in \mathbb{R}_+^m$ chosen by the user. A higher indicator value is desirable. It is given by:

$$I_{OD}(Y_N) = \max_{1 \leq i \leq m} \mathbf{w}_i \left(\max_{\mathbf{y} \in Y_N} \mathbf{y}_i - \min_{\mathbf{y} \in Y_N} \mathbf{y}_i \right).$$

The weights can be used to impose an order on criteria importance relatively to the modeling of a specific problem but it is not mandatory. Although this indicator is cheap to compute, it only takes into account the extent of the Pareto front approximation. By the way, it can result in an information loss of the extent of the Pareto front approximation, as it focuses only on the largest distance along a single dimension.

Distribution metric (DM) [271]

This indicator introduced by [271] aims to correct several drawbacks of the spacing measure [226] and add some information about the extent of the Pareto front. As it is mentioned, the "*spacing metric does not adopt normalized distance, which may result in a bias conclusion, especially when the orders of magnitudes of the objectives differ considerably*". Moreover, it cannot account for holes in the Pareto front, as it considers only closest neighbors. An example pointing out the drawbacks of the spacing metric (4.3.3) is given in Figure 4.3.

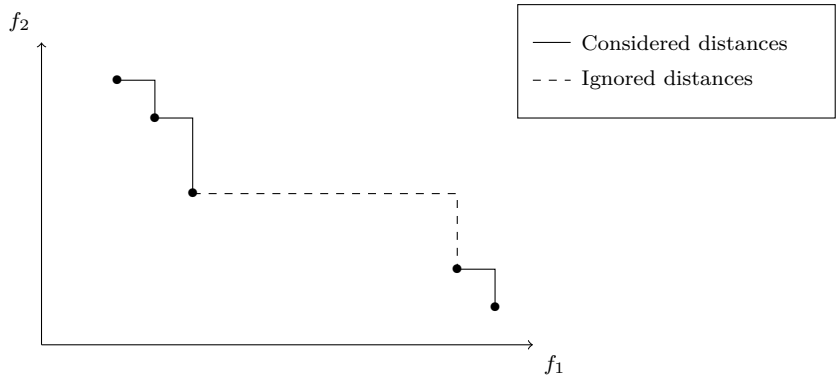


Figure 4.3 An example showing the weaknesses of the spacing metric (inspired by [271]): the spacing metric ignores the gap drawn in dashed lines

.

The DM indicator is given by

$$DM(Y_N) = \frac{1}{|Y_N|} \sum_{i=1}^{m} \left(\frac{\sigma_i}{\mu_i}\right) \left(\frac{\left|\mathbf{y}_i^I - \mathbf{y}_i^N\right|}{\max\limits_{\mathbf{y} \in Y_N} \mathbf{y}_i - \min\limits_{\mathbf{y} \in Y_N} \mathbf{y}_i}\right)$$

with $\sigma_i = \dfrac{1}{|Y_N| - 2} \sum\limits_{j=1}^{|Y_N|-1} \left(d_i^j - \overline{d_i}\right)^2$, $\mu_i = \dfrac{1}{|Y_N| - 1} \sum\limits_{j=1}^{|Y_N|-1} d_i^j$ where $|Y_N|$ is the number of non-dominated objective vectors, $\mathbf{y}^I$ and $\mathbf{y}^N$ are respectively the ideal and nadir objective vectors. d_i^j is the distance of the j-th interval between two adjacent solutions corresponding to the i-th objective, σ_i and μ_i are the standard deviation and mean of the distances relative to the i-th objective, and $\dfrac{\sigma_i}{\mu_i}$ is the coefficient of variance relative to the i-th objective.
A smaller DM indicates better distributed solutions. It takes into account the extent and distribution of the points along the Pareto front approximation. However, it requires the nadir and ideal objective vectors, which may be computationally expensive. As it accounts for holes, this indicator is more relevant for continuous Pareto front approximations.

Uniform assessment metric (I_D) [176]

The I_D indicator measures the variation of distances between elements of a Pareto front approximation based on the construction of a minimum spanning tree. The indicator value is comprised between 0 and 1. The closest to 1, the better. Let Y_N be a Pareto front approximation such that $|Y_N| > 2$. The computation of this indicator is decomposed into several steps:

1. A minimum spanning tree T_G covering all the elements of Y_N based on the Euclidean distance in the objective space is built.

2. Each element $\mathbf{y} \in Y_N$ has at least one neighbor in the spanning set, i.e a vertex adjacent to $\mathbf{y}$. Let $N_{T_G}(\mathbf{y})$ be the set of adjacent vertices to $\mathbf{y}$ in the spanning tree T_G.
   For each $\mathbf{y}^v \in N_{T_G}(\mathbf{y})$, we define a "neighborhood" [176]

   $$N_{y^v}(\mathbf{y}) = \left\{\mathbf{y}^2 \in Y_N, \|\mathbf{y}^2 - \mathbf{y}\| \leq \|\mathbf{y}^v - \mathbf{y}\|\right\}$$

   which corresponds to the subset of Y_N contained in the closed ball of radius $\|\mathbf{y}^v - \mathbf{y}\|$ and centered in $\mathbf{y}$. Note that $\{\mathbf{y}, \mathbf{y}^v\} \in N_{y^v}(\mathbf{y})$. The neighborhoods that contain only two elements, i.e. $\mathbf{y}$ and $\mathbf{y}^v$, are not considered.

3. For all $\mathbf{y} \in Y_N$ and $\mathbf{y}^v \in N_{T_G}(\mathbf{y})$, a distribution relation is defined by

$$\psi(\mathbf{y}, \mathbf{y}^v) = \begin{cases} 0 & \text{if } |N_{y^v}(\mathbf{y})| = 2, \\ \prod_{\mathbf{y}^2 \in N_{y^v}(\mathbf{y}) \setminus \{\mathbf{y}\}} \frac{\|\mathbf{y} - \mathbf{y}^2\|}{\|\mathbf{y} - \mathbf{y}^v\|} & \text{otherwise.} \end{cases}$$

4. There are $2|Y_N| - 2$ neighborhoods. Among them, N_r corresponds to the number of neighborhoods that only contain two elements. The uniform assessment metric is then defined by

$$I_D(Y_N) = \frac{1}{2|Y_N| - N_r - 2} \sum_{\mathbf{y} \in Y_N} \sum_{\mathbf{y}^v \in N_{T_G}(\mathbf{y})} \psi(\mathbf{y}, \mathbf{y}^v)$$

which corresponds to the mean of the distribution relation for neighborhoods containing more than two elements.

This indicator does not require external parameters. Due to the definition of the neighborhood, it takes into account holes in the Pareto front. Indeed, contrary to the spacing metric, it does not consider only closest distances between objective vectors.

Extension measure (EX) [188]

This indicator aims to measure the extent of the Pareto front approximation. It is given by

$$EX(Y_N; Y_P) = \frac{1}{m} \sqrt{\sum_{i=1}^{m} d^2(\mathbf{y}^{i,\star}, Y_N)^2}$$

where $d^2(\mathbf{y}^{i,\star}, Y_N)$ is the minimal Euclidean distance between the objective vector corresponding to the solution to the ith single-objective problem and the set of non-dominated points obtained by a given algorithm in the objective space.

This indicator requires the resolution of m single-objective optimization problems. It penalizes well-distributed Pareto front approximations neglecting the extreme values. It is straightforward to compute.

Diversity indicator based on reference vectors (DIR) [56]

As its name indicates, this indicator uses reference vectors in the objective space to measure the diversity of a Pareto front approximation. The lower this indicator is, the better. Let $Y_R = \{\mathbf{r}^1, \mathbf{r}^2, \ldots, \mathbf{r}^{|Y_R|}\}$ be a set of uniformly generated reference vectors in $\mathbb{R}^m$. For each element of a Pareto front approximation $\mathbf{y} \in Y_N$, the closeness between $\mathbf{y}$ and the reference

vector $\mathbf{r}^j$, for $j = 1, 2, \ldots, |Y_R|$, is given by

$$\text{angle}(\mathbf{r}^j, \mathbf{y}) = \arccos \frac{(\mathbf{r}^j)^T(\mathbf{y} - \mathbf{y}^I)}{\|\mathbf{r}^j\|\|\mathbf{y} - \mathbf{y}^I\|}.$$

If a reference vector $\mathbf{r}^j$ is the closest to an element $\mathbf{y}$ of Y_N relatively to the closeness metric, it is said that $\mathbf{y}$ "*covers the reference vector* $\mathbf{r}^j$" [56]. The coverage vector $\mathbf{c}$ of size $|Y_N|$ represents for each $\mathbf{y} \in Y_N$ the number of reference vectors that $\mathbf{y}$ covers. DIR is the normalized standard deviation of the coverage vector $\mathbf{c}$, defined as

$$DIR(Y_N; Y_R) = \sqrt{\frac{1}{|Y_N|}\sum_{i=1}^{|Y_N|}(\mathbf{c}_i - \bar{c})^2} \;\div\; \left(\frac{|Y_R|}{|Y_N|}\sqrt{|Y_N| - 1}\right)$$

where $\bar{c}$ is the mean of all $\mathbf{c}_i$ for $i = 1, 2, \ldots, |Y_N|$. It is intuitive to understand and cheap to compute (in $\mathcal{O}(m\,|Y_N| \times |Y_R|)$ [56]). It captures both the distribution and the spreading. Nonetheless, it requires the knowledge of the ideal point. The number of reference vectors to choose (at least greater than $|Y_N|$ to be more pertinent) equally plays an important role. It can be biased when the Pareto front is piecewise continuous.

The Riesz s-energy indicator (E_s) [105, 140]

The Riesz s-energy indicator [140] aims at quantifying a good distribution of points in d-dimensional manifolds. Given a Pareto front approximation Y_N, this indicator is defined as:

$$E_s(Y_N) = \sum_{\mathbf{y}^1 \in Y_N} \sum_{\mathbf{y}^2 \in Y_N \setminus \{\mathbf{y}^1\}} \frac{1}{\|\mathbf{y}^1 - \mathbf{y}^2\|^s}$$

where $s > 0$ is a fixed external parameter which controls the degree of uniformity of the elements of Y_N.

An uniformly distributed Pareto front approximation must have a minimal Riesz s-energy value. In [141], it is proved that configurations of points in a rectifiable $d-$dimensional manifold that have minimum Riesz s-energy possess asymptotically uniformly distribution properties for $s \geq d$. Moreover, s does not depend on the shape of the manifold [141].

Use of the Riesz s-energy indicator to assess generation of an uniformly distributed Pareto front approximation can be found in [105].

Laumanns metric (I_L) [165, 166]

The Laumanns metric measures the normalized volume of the objective space dominated by a Pareto front approximation and bounded above by an approximated nadir objective vector. Given a vector $\mathbf{y}$ in the objective space $\mathbb{R}^m$, let $\mathcal{D}(\mathbf{y}) = \{\mathbf{y}^2 \in \mathbb{R}^m,\ \mathbf{y} \leq \mathbf{y}^2\}$ be the set of objective vectors dominated by $\mathbf{y}$. Given a Pareto front approximation Y_N, $\mathcal{D}(Y_N)$ is designed as the dominated space by the set Y_N and is defined as

$$\mathcal{D}(Y_N) = \bigcup_{\mathbf{y} \in Y_N} \mathcal{D}(\mathbf{y}).$$

Let $\mathbf{y}^{j,\star}$ be the j-th outer point of the Pareto front approximation Y_N defined by: for all $i = 1, 2, \ldots, m$,

$$(\mathbf{y}_i^{j,\star}) = \begin{cases} \max\limits_{\mathbf{y} \in Y_N} \mathbf{y}_i & \text{if } i \neq j, \\ \min\limits_{\mathbf{y} \in Y_N} \mathbf{y}_i & \text{otherwise.} \end{cases}$$

We introduce the hypercube $H(Y_N) = \left\{ \mathbf{y} \in \mathbb{R}^m : \mathbf{y} = \mathbf{y}^I + \sum_{i=1}^{m} a_i(\mathbf{y}^{i,\star} - \mathbf{y}^I), a_i \in [0; 1] \right\}$ where $\mathbf{y}^I$ is the ideal point. The Laumanns metric is defined as the ratio of the Lebesgue measure of the intersection of $\mathcal{D}$ and H, with the Lebesgue measure of H:

$$I_L(Y_N) = \frac{\lambda_m(\mathcal{D}(Y_N) \cap H(Y_N)))}{\lambda_m(H(Y_N))}$$

where $\lambda_m(A)$ is the m-dimensional Lebesgue measure of the bounded set $A \subset \mathbb{R}^m$. The metric returns a value between 0 and 1. The higher the better. An illustration is given in Figure 4.4.

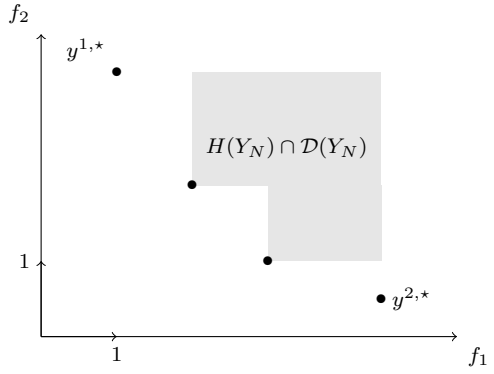


Figure 4.4 The intersection of $H(Y_N)$ and $\mathcal{D}(Y_N)$ for a biobjective minimization problem.

This indicator is biased in favor of convex and extended fronts. Moreover, its computational

complexity in $\mathcal{O}(|Y_N|^{\frac{m}{3}} \operatorname{poly}\log|Y_N|)$ [57] explodes when the objective space dimension increases: in fact, it is similar to the hypervolume indicator (4.3.4) when the reference point is chosen such as $\widetilde{y}^N$.

Similarly, the convex hull surface indicator [270] measures the volume of the convex hull formed by the Pareto front approximation and a reference point $r \in \mathbb{R}^m$ dominated by all the elements of the Pareto front approximation. The greater the value of this indicator is, the better. Computational cost increases exponentially with the number of objectives. As it only considers points on the boundaries of the convex hull, it is more pertinent to use it on convex Pareto front approximations.

Other distribution indicators

Some other indicators are mentioned in this subsection. They require external parameters chosen by the user that can be crucial to their performance. The reader can consult the provided references.

1. Entropy measure [108]: For each element of Y_N, an influential function (a Gaussian function centered in $\mathbf{y}$ for $\mathbf{y} \in Y_N$) is defined, which enables the creation of a density function considered as the sum of influential functions for each element $\mathbf{y} \in Y_N$. Peaks and valleys in the objective space are considered as places where information can be measured. A "good" Pareto front approximation should have an uniform density function in the objective space. The subset of the objective space bounded by the nadir and ideal objective vectors is firstly normalized, then divided into boxes, whose the number is decided by the user. Based on this discretization of the objective space, the indicator is computed using the values of the density function for each center of each box and the Shannon formula of entropy [231]. The higher the value, the better.

2. Cluster CL_μ and Number of Distinct Choices NDC_μ [263]: Given two respective good (ideal objective vector) and bad (maximum objective vector assuming it exists) objective vectors $\widetilde{\mathbf{y}}^I$ and $\widetilde{\mathbf{y}}^M$, the objective space (preliminary normalized) is divided into hyperboxes of size μ ($\in (0;\ 1]$). NDC_μ is defined as the number of hyperboxes containing elements of the Pareto front approximation. For this indicator, a higher value is desirable. CL_μ is then defined as $CL_\mu(Y_N) = \frac{|Y_N|}{NDC_\mu(Y_N)}$. A higher value of the CLU_μ indicator is the consequence of a more clustered distribution of the elements of a Pareto front approximation and so a lower value is considered to be better.

3. Sigma diversity metrics σ and $\overline{\sigma}$ [195]: The objective space is divided into zones delimited by uniformly distributed reference lines starting from the origin whose the number

equals $|Y_N|$. The indicator value is the ratio of the number of objective vectors that are sufficiently close to the reference lines according to the Euclidean norm with a threshold d chosen by the user, over the total number of reference lines. The higher the value, the better.

4. Diversity comparison indicator DCI [172]: It is a k-ary spread indicator. The zone of interest in the objective space delimited by lower and upper bounds is divided into a number of hyperboxes. For each Pareto front approximation, a contribution coefficient is computed relatively to the hyperboxes where non-dominated points are found. For each Pareto front approximation, DCI returns the mean of contribution coefficients relatively to all hyperboxes of interest. A variant is the $M{-}DI$ indicator [8] (Modified Diversity Indicator) which considers a distributed reference set in the objective space instead of the set of non-dominated points from the union of the k Pareto front approximations.

A drawback of these indicators is the choice of external parameters (d threshold, μ size, number of hyperboxes) that can wrongly favor Pareto front approximations over others. σ and CL_μ can be considered as cardinal indicators too and therefore suffer from the same drawbacks as the above cardinal indicators.

4.3.4 Convergence and distribution indicators

These indicators are of two types: some enable to compare several Pareto approximations in term of distribution and Pareto dominance. The others give a value that capture distribution, spreading and convergence at the same time.

Inverted generational distance (IGD) [62]

IGD has a quite similar form than GD. It captures the average minimal distance from an element of a discrete representation of the Pareto front to the closest point in the Pareto front approximation. It is given by

$$IGD(Y_N; Y_P) = \frac{1}{|Y_P|} \left(\sum_{\mathbf{y}^2 \in Y_P} \left(\min_{\mathbf{y}^1 \in Y_N} \|\mathbf{y}^1 - \mathbf{y}^2\| \right)^p \right)^{\frac{1}{p}}.$$

Generally, $p = 2$. A lower value is considered to be better. Pros and cons are the same as for the GD indicator (4.3.2).

When an element of the Pareto front approximation Y_N does not belong to the set of nearest points to the Pareto optimal set Y_P, it is ignored by the IGD indicator. The authors of [243] propose a variant of the IGD indicator, named IGD–NS, which takes into account these non-contributed elements in an Euclidean distance-based indicator. Let

$$Y_{NC} = \left\{ \mathbf{y} \in Y_N : \forall \mathbf{y}^2 \in Y_P, \|\mathbf{y} - \mathbf{y}^2\| \neq \min_{\mathbf{y}^1 \in Y_N} \|\mathbf{y}^1 - \mathbf{y}^2\| \right\}$$

be the set of non-contributed elements of Y_N. The IGD–NS indicator is defined by

$$IGD\text{–}NS(Y_N; Y_P) = \sum_{\mathbf{y}^2 \in Y_P} \min_{\mathbf{y}^1 \in Y_N} \|\mathbf{y}^1 - \mathbf{y}^2\| + \sum_{\mathbf{y}^2 \in Y_P} \min_{\mathbf{y}^1 \in Y_{NC}} \|\mathbf{y}^1 - \mathbf{y}^2\|.$$

A lower indicator value is desirable.

Averaged Hausdorff distance (Δ_p) [227]

In [227], the authors combine GD (4.3.2) and IGD (4.3.4) into a new indicator, called the averaged Hausdorff distance Δ_p defined by

$$\Delta_p(Y_N; Y_P) = \max \{ GD_p(Y_N; Y_P),\ IGD_p(Y_N; Y_P) \}$$

where GD_p and IGD_p are slightly modified versions of the GD and IGD indicators defined as

$$GD_p(Y_N; Y_P) = \left(\frac{1}{|Y_N|} \sum_{\mathbf{y}^1 \in Y_N} \left(\min_{\mathbf{y}^2 \in Y_P} \|\mathbf{y}^1 - \mathbf{y}^2\| \right)^p \right)^{\frac{1}{p}}$$

and

$$IGD_p(Y_N; Y_P) = \left(\frac{1}{|Y_P|} \sum_{\mathbf{y}^2 \in Y_P} \left(\min_{\mathbf{y}^1 \in Y_N} \|\mathbf{y}^1 - \mathbf{y}^2\| \right)^p \right)^{\frac{1}{p}}.$$

This indicator is to be minimized. It is straightforward to compute and to understand. On the contrary, it requires the knowledge of the Pareto front. The authors of [227] introduce this new indicator to correct the drawbacks of the GD and IGD indicators. It can be used to compare continuous and discontinuous approximations of Pareto fronts.

In [251], the authors propose an extension of the averaged Hausdorff distance indicator, called the p, q-averaged distance $\Delta_{p,q}$ for $p, q \in \mathbb{R} \setminus \{0\}$. Given two finite sets of objective vectors

$Y^1 \subset \mathbb{R}^m$, $Y^2 \subset \mathbb{R}^m$, the generational distance $GD_{p,q}$ for $p, q \in \mathbb{R} \setminus \{0\}$ is defined as

$$GD_{p,q}(Y^1, Y^2) = \left(\frac{1}{|Y^1|} \sum_{\mathbf{y}^1 \in Y^1} \left(\frac{1}{|Y^2|} \sum_{\mathbf{y}^2 \in Y^2} \|\mathbf{y}^1 - \mathbf{y}^2\|^q \right)^{\frac{p}{q}} \right)^{\frac{1}{p}}.$$

When $p < 0$ or $q < 0$, $GD_{p,q}$ exists if and only if $Y^1 \bigcap Y^2 = \emptyset$. Given a Pareto front approximation Y_N and a discrete representation of the Pareto front $Y_P \subseteq \mathcal{Y}_p$, the p, q-averaged distance indicator is defined as

$$\Delta_{p,q}(Y_N; Y_P) = \max\left(GD_{p,q}\left(Y_N, Y_P \setminus Y_N\right), GD_{p,q}\left(Y_P, Y_N \setminus Y_P\right)\right).$$

As the averaged Hausdorff distance, this indicator is not Pareto compliant [251]. However, once that the values of p and q are selected, it is straightforward to compute and to understand.

Modified inverted generational distance (IGD^+) [147]

Although the GD (4.3.2) and IGD (4.3.4) indicators are commonly used due to their low computational cost [218], one of their major drawbacks is that they are non monotone [147]. The Δ_p indicator (4.3.4) has the same problem [227].

Also, the authors of [147] propose a slightly different version of the IGD indicator named IGD^+ integrating the dominance relation computable in $\mathcal{O}(m\ |Y_N| \times |Y_P|)$ where Y_P is a fixed Pareto optimal solution set. It is weakly Pareto compliant, i.e. :

$$\text{for all } Y_N^1, Y_N^2 \in \Omega \text{ such that } Y_N^1 \preceq Y_N^2,\ IGD^+(Y_N^1; Y_P) \leq IGD^+(Y_N^2; Y_P).$$

The IGD^+ indicator is defined by

$$IGD^+(Y_N; Y_P) = \frac{1}{|Y_P|} \sum_{\mathbf{y}^2 \in Y_P} \min_{\mathbf{y}^1 \in Y_N} \|(\mathbf{y}^1 - \mathbf{y}^2)_+\|$$

where $(\mathbf{y}^1 - \mathbf{y}^2)_+ = (\max(0, \mathbf{y}_i^1 - \mathbf{y}_i^2))_{i=1,2,\ldots,m}$.

As opposed to the IGD indicator, IGD^+ takes into account the dominance relation between an element of Y_P and an element of Y_N when computing their Euclidean distance. A reference set Y_R can also be used instead of Y_P: the authors of [146] analyze the choice of such reference sets. This indicator can be used with discontinuous and continuous Pareto fronts.

Similarly to IGD^+, given a reference set $Y_R \subset \mathbb{R}^m$, Dist_{1R} [78] is given by

$$\text{Dist}_{1R}(Y_N; Y_R) = \frac{1}{|Y_R|} \sum_{\mathbf{r} \in Y_R} \min_{\mathbf{y} \in Y_N} \max_{i=1,2,\ldots,m} \{0,\ w_i(\mathbf{y}_i - \mathbf{r}_i)\}$$

with w_i a relative weight assigned to objective i. For all these indicators, a lower value is desirable.

R_1 and R_2 indicators [139]

Let Y_N^1 and Y_N^2 be two Pareto front approximations, $\mathcal{U}$ a set of utility functions $u : \mathbb{R}^m \to \mathbb{R}$ mapping each point in the objective space into a measure of utility, and p a probability distribution on the set $\mathcal{U}$. To each $u \in \mathcal{U}$ are associated $u^\star(Y_N^1) = \max_{\mathbf{y} \in Y_N^1} u(\mathbf{y})$ and $u^\star(Y_N^2) = \max_{\mathbf{y} \in Y_N^2} u(\mathbf{y})$. The two indicators measure to which extent Y_N^1 is better than Y_N^2 over the set of utility functions $\mathcal{U}$. The R_1 indicator is given by

$$R_1(Y_N^1, Y_N^2; \mathcal{U}, p) = \int_{u \in \mathcal{U}} C(Y_N^1, Y_N^2, u)\ p(u)\ \text{du}$$

where

$$C(Y_N^1, Y_N^2, u) = \begin{cases} 1 & \text{if } u^\star(Y_N^1) > u^\star(Y_N^2), \\ 1/2 & \text{if } u^\star(Y_N^1) = u^\star(Y_N^2), \\ 0 & \text{if } u^\star(Y_N^1) < u^\star(Y_N^2). \end{cases}$$

The R_2 indicator defined as

$$R_2(Y_N^1, Y_N^2; \mathcal{U}, p) = E\left(u^\star(Y_N^1)\right) - E\left(u^\star(Y_N^2)\right) = \int_{u \in \mathcal{U}} \left(u^\star(Y_N^1) - u^\star(Y_N^2)\right) p(u)\text{du}.$$

is the expected difference in the utility of a Pareto front approximation Y_N^1 with another one Y_N^2. In practice, these two indicators use a discrete and finite set $\mathcal{U}$ of utility functions associated with an uniform distribution over $\mathcal{U}$ [277]. The two indicators can then be rewritten as

$$R_1(Y_N^1, Y_N^2; \mathcal{U}) = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} C(Y_N^1, Y_N^2, u) \text{ and } R_2(Y_N^1, Y_N^2; \mathcal{U}) = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} u^\star\left(Y_N^1\right) - u^\star(Y_N^2).$$

If $R_2(Y_N^1, Y_N^2; \mathcal{U}) > 0$, then Y_N^1 is considered as better than Y_N^2. Else if $R_2(Y_N^1, Y_N^2; \mathcal{U}) \geq 0$, Y_N^1 is considered as not worse than Y_N^2.

The authors of [139] recommend to use the utility set $\mathcal{U}_\infty = (u_w)_{\mathbf{w} \in W}$ of weighted Tchebycheff

utility functions, with

$$u_w(y) = -\max_{i=1,2,\ldots,m} (\mathbf{w}_i \, |\mathbf{y}_i - \mathbf{r}_i|)$$

for $\mathbf{y} \in \mathbb{R}^m$ where $\mathbf{r}$ is a reference objective vector chosen so that any objective vector of the feasible objective set does not dominate $\mathbf{r}$ (or as an approximation of the ideal point [51, 52, 277]) and $\mathbf{w} \in W$ a weight vector such that for all $\mathbf{w} \in W$ and $i = 1, 2, \ldots, m$,

$$\mathbf{w}_i \geq 0 \text{ and } \sum_{i=1}^{m} \mathbf{w}_i = 1.$$

Zitzler et al. [277] suggest using the set of augmented weighted Tchebycheff utility functions defined by

$$u_w(\mathbf{y}) = -\left(\max_{i=1,2,\ldots,m} \mathbf{w}_i \, |\mathbf{y}_i - \mathbf{r}_i| + \rho \sum_{i=1}^{m} |(\mathbf{y}_i - \mathbf{r}_i| \right)$$

where ρ is a sufficiently small positive real number.

As given in [51], for $m = 2$ objectives, W can be chosen such that:

1. $W = \left\{ (0,1), \left(\frac{1}{k-1}, \; 1 - \frac{1}{k-1}\right), \left(\frac{2}{k-1}, \; 1 - \frac{2}{k-1}\right), \ldots, (1,0) \right\}$ is a set of k weights uniformly distributed in the space $[0;1]^2$.
2. $W = \left\{ \left(\frac{1}{1+\tan\varphi}, \frac{\tan\varphi}{1+\tan\varphi} \right), \varphi \in \Phi_k \right\}$ where $\Phi_k = \left\{ 0, \frac{\pi}{2(k-1)}, \frac{2\pi}{2(k-1)}, \ldots, \frac{\pi}{2} \right\}$ is a set of weights uniformly distributed over the trigonometric circle.

The I_{R_2} indicator [51] is an unary indicator derived from R_2 defined as (in the case of weighted Tchebycheff utility functions)

$$I_{R_2}(Y_N; W) = \frac{1}{|W|} \sum_{\mathbf{w} \in W} \min_{\mathbf{y} \in Y_N} \left\{ \max_{i=1,2,\ldots,m} (\mathbf{w}_i \, |\mathbf{y}_i - \mathbf{r}_i|) \right\}.$$

The lower this index, the better.

As Knowles et al. [159] remark, *"the application of R_2 depends up on the assumption that it is meaningful to add the values of different utility functions from the set $\mathcal{U}$. This simply means that each utility function in $\mathcal{U}$ must be appropriately scaled with respect to the others and its relative importance"*. R-indicators are only monotonic, i.e. $I(Y_N^1) \geq I(Y_N^2)$ in case Y_N^1 weakly dominates Y_N^2. They do not require important computations as the number of objectives increases. The reference point has to be chosen carefully. Studies concerning the properties of the R_2 indicator can be found in [51, 52, 254].

G-metric [184]

This indicator enables to compare k Pareto front approximations based on two criteria: the repartition of their elements and the level of domination in the objective space. It is compliant with the weak dominance relation as defined above. Its computation decomposes into several steps: given k Pareto front approximations $(Y_N^1, Y_N^2, \ldots, Y_N^k)$:

1. Scale the values of the objective vectors corresponding to the images of the decision vectors in the k sets, i.e. extract the non-dominated objective vectors from the union $\bigcup_{j=1}^{k} Y_N^j$, then normalize all objective vectors according to the extreme values of the objective vectors of this set.

2. Group the Pareto front approximations according to their degree of dominance. In level L^1 will be put all Pareto front approximations which are not dominated by the union of the k Pareto front approximations; remove them from the considered Pareto front approximations; then in L^2, will be put the Pareto front approximations which are not dominated by the union of the remaining Pareto front approximations, and so on.

3. For each level of dominance L^q for $q = 1, 2, \ldots, Q$, where Q is the number of levels, dominated points belonging in the set $\bigcup_{Y_N \in L^q} Y_N$ are removed. Each objective vector in each set of the same level possesses a zone of influence. It is a ball of radius ρ centered in this last one. The radius ρ considers distances between neighbors objective vectors [169] from the k Pareto front approximations. For each Pareto front approximation belonging to the same level of dominance, a measure of dispersion is computed. This last one takes into account the zone of influence that the union of non-dominated elements of the set cover in the objective space. The smaller the value, the closer the points are.

4. The G-metric associated to an Pareto front approximation is the summation of the dispersion measure of this set and the largest dispersion measure of Pareto front approximations of lower dominance degree for each level. The bigger, the better.

The computation cost is quite important (in $\mathcal{O}(k^3 \times \max_{j=1,2,\ldots,k} |Y_N^j|^2)$ [184]) but the cost can be decreased when one considers a small number of Pareto front approximations. Note that this indicator highly depends on the computation of the radius ρ when defining zones of influence. This indicator can be used for continuous and discontinuous Pareto fronts, especially to compare two Pareto front approximations, in terms of dominance and distribution in the objective space.

Performance comparison indicator (PCI) [173]

The Performance Comparison indicator [173] PCI was conceived to rectify the main drawback of the ϵ-indicator. PCI enables to compare k Pareto front approximations taking into account their level of dominance and their distribution in the objective space. PCI uses a reference set composed of all non-dominated points taken from the union of the k Pareto front approximations. Using extreme values of the reference set, all objective vectors of the k Pareto front approximations are normalized. Then PCI divides the set into different clusters based on a distance threshold σ. PCI estimates the minimum distance move of the points in the Pareto front approximations to weakly dominate all points in the clusters. A lower value reflects better closeness of the Pareto front approximation to the reference set.

This indicator possesses a quadratic computational cost and is Pareto compliant when only two Pareto front approximations are considered [173]. The authors propose the following choice for the threshold, i.e.

$$\sigma \approx \frac{1}{\sqrt[m-1]{|Y_R|(m-1)!} - (m/2)}$$

where $|Y_R|$ is the size of the reference set, which enables this indicator to be external-parameter-free.

The recent binary dominance move indicator [174] DoM is a generalization of the PCI indicator, as it computes the minimum distance move from one Pareto front approximation to weakly dominate another. A polynomial algorithm is proposed in [174] in the biobjective case. To the best of our knowledge, no extension of the DoM indicator to more objectives has been proposed yet.

Hyperarea/hypervolume metrics (HV) [274]

Named also *S-metric*, the hypervolume indicator is described as the volume of the space in the objective space dominated by the Pareto front approximation Y_N and delimited from above by a reference objective vector $r \in \mathbb{R}^m$ such that for all $\mathbf{y} \in Y_N$, $\mathbf{y} \leq \mathbf{r}$. The hypervolume indicator is given by

$$HV(Y_N; \mathbf{r}) = \lambda_m \left(\bigcup_{\mathbf{y} \in Y_N} [\mathbf{y}; \mathbf{r}] \right)$$

where λ_m is the m-dimensional Lebesgue measure. An illustration is given in Figure 4.5 for the biobjective case ($m = 2$). This indicator is to be maximized.

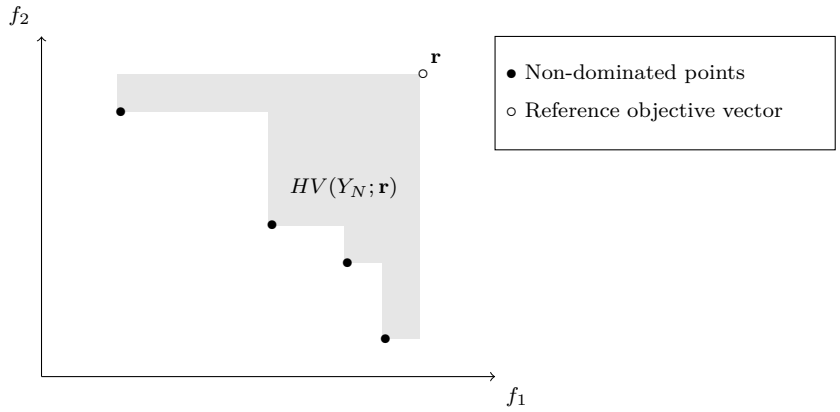


Figure 4.5 Illustration of the hypervolume indicator for a biobjective problem.

If the Pareto front is known, *the Hyperarea ratio* is given by

$$HR(Y_N, Y_P; r) = \frac{HV(Y_N;\mathbf{r})}{HV(Y_P;\mathbf{r})}.$$

The greater the ratio is (converges toward 1), the better the approximation is.

The hypervolume indicator and some closely related metrics are the only known unary indicators to be strictly monotonic [106, 123, 275, 277], i.e. if a Pareto front approximation Y_N^1 is better than another Pareto front approximation Y_N^2, $HV(Y_N^1;\mathbf{r}) > HV(Y_N^2;\mathbf{r})$ assuming that all elements of the two Pareto front approximations are in the interior of the region which dominates the reference point. The best known complexity upper cost is $\mathcal{O}(|Y_N|^{\frac{m}{3}} \operatorname{poly}\log|Y_N|)$ [57]. To the best of our knowledge, no implementation of this algorithm is available. Practically, efficient implementations of the exact hypervolume indicator can be found in [37,150,163,221,222,258]. The second drawback is the choice of the reference point, as illustrated in Figure 4.6. A practical guide to specify the reference point can be found in [145].

For the biobjective case, it was shown [31] that the optimal distribution of non-dominated points which maximizes the hypervolume indicator depends on the slope of the Pareto front. Consequently, if the Pareto front is highly nonlinear, a non-uniform Pareto front approximation may have a higher hypervolume value according to another incomparable Pareto front approximation. Other theoretical results can be found in [46–48]. Due to its properties, it is widely used in the evolutionary community in the search of potential interesting new points or to compare algorithms.

Similarly, [274] introduces the *Difference D of two sets* Y_N^1 and Y_N^2. $D(Y_N^1, Y_N^2)$ enables to

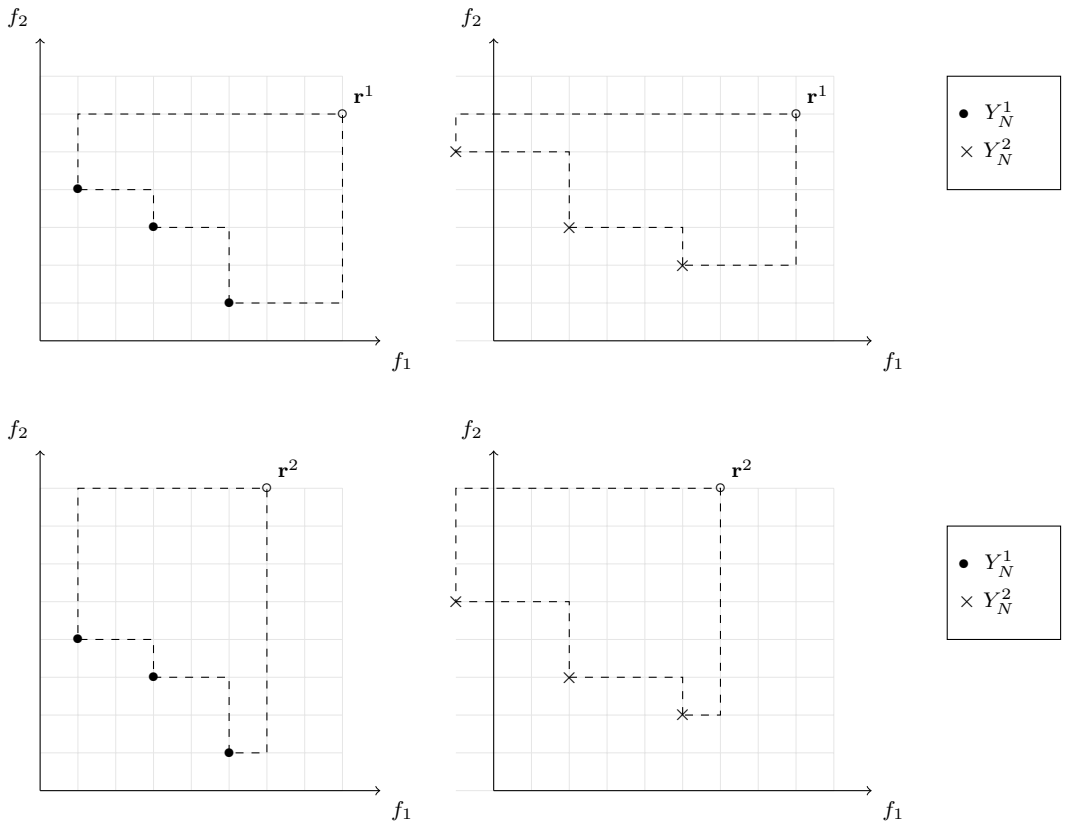


Figure 4.6 The relative value of the hypervolume metric depends on the chosen reference point $\mathbf{r}^1$ or $\mathbf{r}^2$ (inspired by [159]). On the top, two non-dominated Y_N^1 and Y_N^2 sets are shown, with $HV(Y_N^1;\mathbf{r}^1) > HV(Y_N^2;\mathbf{r}^1)$. On the bottom, $HV(Y_N^2;\mathbf{r}^2) > HV(Y_N^1;\mathbf{r}^2)$.

measure the size of the area dominated by Y_N^1 but not by Y_N^2 in the objective space.

The *Hyperarea Difference* was suggested by [263] to compensate the lack of information about the theoretical Pareto front. Given a good objective vector $\widetilde{\mathbf{y}}^I$ and a bad objective vector $\widetilde{\mathbf{y}}^M$, we can approximate the size of the area dominated by the Pareto front (or circumvent the objective space by a rectangle). The Hyperarea Difference is just the normalization of the dominated space by the Pareto front approximation in the objective space over the given rectangle.

More recently, a pondered hypervolume by weights indicator was introduced by [275] to give a preference of an objective according to another. More volume indicators can be found in [263]. Some other authors [152] (for biobjective optimization problems) suggest to compute the hypervolume defined by a reference point and the projection of the points belonging to the Pareto front approximation on the line delimited by the two extreme points. This measure enables to better estimate the distribution of the points along the Pareto front

(in fact, it can be shown that for a linear Pareto front, an uniform distribution of points maximizes the hypervolume indicator: see [30–32, 234] for more details about the properties of the hypervolume indicator). A logarithmic version of the hypervolume indicator called the logarithmic hypervolume indicator [123] is defined by

$$\log HV(Y_N; \mathbf{r}) = \lambda_m \left(\bigcup_{\mathbf{y} \in Y_N} [\log \mathbf{y}; \log \mathbf{r}] \right)$$

with the same notations as previously. Note that this indicator can only be used with positive vectors in $\mathbb{R}^m$. Finally, we can mention a generalization of the hypervolume indicator called the cone-based hypervolume indicator that was introduced recently by [100] and an extension of the hypervolume indicator to reference point free weighted hypervolume indicators based on set monotonic and desirability functions [103].

Hypervolume Sharpe-Ratio indicator I_{HSR} [265]

The conception of this indicator proposed by [265] lies on an analogy between the financial Portfolio Selection problem [186] and the quality of a Pareto front approximation. This analogy interprets the elements of the approximation set as assets with expected returns. An investor must allocate capital to maximize the expected return of the assembled portfolio while minimizing the expected variance (associated to risk).

To solve the financial Portfolio Selection problem, it is common to look for a strategy which maximizes the financial performance index known as reward-to-volatility ratio or Sharpe ratio [68]. Let $A = \{a^1, a^2, \ldots, a^{|A|}\}$ be a non-empty set of assets, let $\mathbf{r}^a \in \mathbb{R}^{|A|}$ be the expected return of these assets and $\mathbf{Q} \in \mathbb{R}^{|A| \times |A|}$ the covariance matrix of asset returns. The Sharpe-Ratio maximization problem is defined as

$$\begin{array}{lll} \max\limits_{\mathbf{z} \in [0,1]^{|A|}} & h^A(\mathbf{z}) & = \dfrac{(\mathbf{r}^a)^\top \mathbf{z} - r^f}{\sqrt{\mathbf{z}^\top \mathbf{Q} \mathbf{z}}} \\ \text{such that} & \sum\limits_{i=1}^{|A|} \mathbf{z}_i & = 1 \end{array}$$

where $r^f \in \mathbb{R}$ is the return of a riskless asset. This non-linear problem which may be difficult to solve can be transformed (see [68] for more details) into the following convex problem

(under the condition that Q be symmetric positive)

$$\begin{array}{lll} \min\limits_{\mathbf{w}\in\mathbb{R}_+{}^{|A|}} & g^A(\mathbf{w}) & = \mathbf{w}^T\mathbf{Q}\mathbf{w} \\ \text{such that} & \sum\limits_{i=1}^{|A|}(\mathbf{r}_i^a - r^f)\mathbf{w}_i & = 1. \end{array}$$

The optimal strategy $z^\star$ of the first problem is given by $\mathbf{z}^\star = \mathbf{w}^\star / \left(\sum_{i=1}^{|A|} \mathbf{w}_i^\star\right)$ where $\mathbf{w}^\star$ is the solution of the constrained quadratic problem defined above.

The authors of [265] define the Hypervolume Sharpe-Ratio indicator as

$$I_{HSR}(Y_N; \mathbf{y}^l, \mathbf{y}^u) = \max_{\mathbf{z}\in\mathcal{Z}^{sr}} h^{Y_N}(\mathbf{z})$$

where $\mathcal{Z}^{sr} \subset [0;1]^{|Y_N|}$ is the set of feasible strategies of the Sharpe ratio maximization problem and $\mathbf{y}^l \in \mathbb{R}^m$ and $\mathbf{y}^u \in \mathbb{R}^m$ two reference points such that $\mathbf{y}^l < \mathbf{y}^u$. The definition of the covariance matrix and the asset returns inspired by the hypervolume indicator are given by

$$\mathbf{r}_j^{|Y_N|} = \frac{\lambda_m\left([\mathbf{y}^l;\mathbf{y}^u] \cap [\mathbf{y}^j;+\infty[\right)}{\lambda_m\left([\mathbf{y}^l;\mathbf{y}^u]\right)} = \frac{\prod_{k=1}^m \max\left(\mathbf{y}_k^u - \max(\mathbf{y}_k^j, \mathbf{y}_k^l), 0\right)}{\prod_{k=1}^m (\mathbf{y}_k^u - \mathbf{y}_k^l)}$$

and

$$\mathbf{Q}_{i,j} = \frac{\lambda_m\left([\mathbf{y}^l;\mathbf{y}^u] \cap [\mathbf{y}^i;+\infty[\cap [\mathbf{y}^j;+\infty[\right)}{\lambda_m\left([\mathbf{y}^l;\mathbf{y}^u]\right)} = \frac{\prod_{k=1}^m \max\left(\mathbf{y}_k^u - \max(\mathbf{y}_k^i, \mathbf{y}_k^j, \mathbf{y}_k^l), 0\right)}{\prod_{k=1}^m (\mathbf{y}_k^u - \mathbf{y}_k^l)}$$

for $i, j \in \{1, 2, \ldots, |Y_N|\}$ where λ_m is the m-dimensional Lebesgue measure. The riskless asset value is set to $r^f = 0$. The greater the indicator value is, the better.

The I_{HSR} indicator has desirable properties: providing that $\mathbf{y}^u \leqq \mathbf{y} < \mathbf{y}^l$ for all $\mathbf{y} \in Y_N$, the I_{HSR} indicator is proved to be monotonic [136, 137], but not strictly monotonic [137]. Other theoretical results can be found in [136, 137]. Due to its relation with a financial model, it is interpretative. However, it is sensitive to the choice of the reference point $\mathbf{y}^u$ [137]. Its main drawback is its computational cost, directly linked to the resolution of the quadratic formulation of the Sharpe ratio maximization problem. Assuming the associated correlation matrix $\mathbf{Q}$ is symmetric positive, this indicator can be computed in $\mathcal{O}(|Y_N|^3)$ operations (for theoretical complexity results, the reader is invited to refer to [199]; see also [201, chapter 16]). Practically, existing quadratic constrained solvers can efficiently compute the indicator value for a given Pareto front approximation (see [201, chapter 16] for a list of quadratic solvers).

4.3.5 Quality assessment of Pareto set approximations in decision space

By definition, performance indicators enable to characterize properties of Pareto front approximations with respect to their diversity, spread and convergence. Moreover, it is possible to build indicators which assess the quality of Pareto set approximations in the decision space, i.e. to design mappings $I : \Psi \to \mathbb{R}$. Some research works explore the design of such indicators. For example, Zitzler et al. [276] propose the M_1, M_2 and M_3 indicators to respectively assess convergence, diversity and extension properties of Pareto set approximations. In [225], the author suggests using a cardinality indicator that returns the number of non-dominated points $|X_N|$, coverage indicator or uniformity indicator in the feasible set. In [246], the authors conceive diversity indicators based on diversity preference relations in the feasible set. In [87], the authors define diversity crowding-distance indicator in the decision space. Indicators to take into account diversity both in the decision space and the objective space can be found in [233, 247]. Finally, in [102], the authors present some measures to qualify approximation sets in level set approximations, which are subsets of the feasible set.

Sayin [225] states that the decision maker is firstly interested by the quality of the best trade-off solutions found in the objective space as long as corresponding decision variables satisfy the constraints. Furthermore, the number of objectives is usually smaller than the number of variables, which makes the Pareto front approximation easier to study/visualize.

4.4 Some usages of performance indicators

This section focuses on four applications of performance indicators: comparison of algorithms for multiobjective optimization, embedding performance indicators in multiobjective optimization algorithms, definition of stopping criteria, and the use of relevant distribution and spread indicators for assessing the diversity characterization of a Pareto front approximation.

4.4.1 Comparison of algorithms

The first use of performance indicators is to evaluate the performance of algorithms on a multiobjective problem. In single-objective optimization, the most used graphical tools to compare algorithms include performance profiles [97] and data profiles [192] (see also [36] for a detailed survey on the tools to compare single-optimization algorithms). More specifically, let $\mathcal{S}$ be a set of solvers and $\mathcal{P}$ the set of benchmarking problems. Let $t_{p,s} > 0$ be a performance measure of solver $s \in \mathcal{S}$ on problem $p \in \mathcal{P}$: the lower, the better. Performance and data profiles combine performance measures of solvers $t_{p,s}$ to enable a general

graphic representation of the performance of each solver relatively to each other on the set of benchmarking problems $\mathcal{P}$.

To the best of our knowledge, Custódio et al. [74] are the first to use data and performance profiles for multiobjective optimization. For each problem $p \in \mathcal{P}$, they build a Pareto front approximation $Y_N^p = \bigcup_{s \in \mathcal{S}} Y_N^{p,s}$ composed of the union of all Pareto front approximations $Y_N^{p,s}$ generated by each solver $s \in \mathcal{S}$ for the problem p. All dominated points are then removed. Pareto front approximations and relative Pareto optimal solution sets are then compared using cardinality and Γ and Δ metrics proposed by [74].

One of the critics we can make with this approach is the use of distribution and cardinality indicators that do not **capture order relations between two different sets**. The choice of (weakly) monotonic indicators or ($\prec$-complete / $\prec$-compatible) $\lhd$-complete / $\lhd$-compatible comparisons methods is more appropriated in this context [139, 159, 277, 280]. Among them, dominance move (4.3.4), G-metric (4.3.4), binary ϵ-indicator (4.3.2), Hypervolume Sharpe-Ratio indicator (4.3.4) and volume-space metrics (4.3.4) have properties corresponding to these criteria. Mathematical proofs can be found in [31, 51, 52, 136, 137, 159, 174, 184, 280]) and are synthesized in Appendices. An example of performance profile using the hypervolume indicator (4.3.4) can be found in [180]. The use of performance indicators such as GD (4.3.2) or IGD (4.3.4) as it is done in [4, 52] is not a pertinent choice due to their inability to capture dominance relation. Instead, we suggest to use their weakly monotonic counterpart IGD^+ (4.3.4) or DOA (4.3.2), that can be cheaper to compute than for example the hypervolume indicator when the number of objectives is high. It is equally possible to build Pareto-compliant indicators by considering a combination of weakly Pareto compliant indicators with at least one strictly Pareto compliant indicator as it is proposed in [106].

The attainment function [122] is another tool for the performance assessment of multiobjective (stochastic) solvers. Assuming a multiobjective solver has produced k Pareto front approximations Y_N^j for $j = 1, 2, \ldots, k$ on a given problem, the empirical attainment function $\alpha : \mathbb{R}^m \to [0, 1]$ is defined as

$$\alpha(\mathbf{y}) = \frac{1}{k} \sum_{j=1}^{k} \mathbb{1}\{Y_N^j \preceq \{\mathbf{y}\}\}.$$

For a given $\mathbf{y} \in \mathbb{R}^m$, the attainment function estimates the probability that the multiobjective solver reaches (in term of dominance in the objective space) the objective vector $\mathbf{y}$. The interested reader can refer to [49, 119, 120, 122, 277] for additional information.

4.4.2 Embedding performance indicators in multiobjective optimization algorithms

Performance indicators are able to quantify properties a good Pareto front approximation should possess. It is then logical to incorporate them into multiobjective optimization methods. By optimizing directly the indicator, one can hope to obtain approximations of the Pareto front satisfying demanding properties. For these last years, the evolutionary multiobjective community has frequently adopted this approach.

For example, performance indicators such as R_2 [245] or HV [38], I_ϵ [278], IGD^+ [105, 185], IGD–NS [243] are used in selection mechanisms in evolutionary algorithms.The reader is invited to consult the recent survey [104] for more information on indicator-based multiobjective evolutionary algorithms. Similarly, the Γ-indicator [74] enables to identify holes in the Pareto front approximation around which the algorithm can explore to improve diversity. In global stochastic optimization, some methods integrate hypervolume indicator [44, 101, 111] and its variants [112] or R_2 [91] to better explore the decision space. In [3], the authors use Radial Basis models and the hypervolume indicator to identify next promising points to evaluate. In [5], the authors propose a multiobjective optimistic algorithm using the additive ϵ-indicator (4.3.2) and analyse its link with the weighted Tchebysheff approach.

The transformation of a multiobjective optimization problem into a single-objective quality indicator based problem implies a loss of information. Indeed, the choice of a specific performance indicator reflects the personal preferences of the decision user. It is then important to understand the bias of this choice on the solution set found. Given a performance indicator, the concept of optimal μ-distribution [31] refers to the study of the optimal distributions of non-dominated points of size μ which belong to the Pareto front and maximize (or minimize) the performance indicator for a given multiobjective problem. Their study enables to understand bias of considered indicators and analyze the behavior of bounded size indicator-based algorithms. The first ones were done for the hypervolume indicator and some of its variants in the biobjective case [31, 32, 100] then extended to more objectives in [30, 234]. Theoretical results for the R_2 indicator [51], the Δ_p indicator [220] or the Hypervolume Sharpe-Ratio indicator [137] for the biobjective case exist too.

4.4.3 Stopping criteria of multiobjective algorithms

To generate a Pareto front approximation, two approaches are currently considered. The first category, named as *scalarization methods*, consists in aggregating the objective functions and to solve a series of single-objective problems. Surveys about scalarization algorithms

can be found for example in [259]. The second class, designed as *a posteriori articulations of preferences* [74] methods, aims at obtaining the whole Pareto front without combining any objective function in a single-objective framework. Evolutionary algorithms, Bayesian optimization methods [101] or deterministic algorithms such as DMS [74] belong to this category.

For scalarization methods, under some assumptions, solutions to single-objective problems can be proved to belong to the Pareto front or a local one. So, defining stopping criteria results in choosing the number of single-objective problems to solve via the choice of parameters and a single-objective stopping criterion for each of them. Stopping at a predetermined number of function evaluations is often used in the context of blackbox optimization [27]. The use of performance indicators also is not relevant.

A posteriori methods consider a set of points in the objective space (a population) that is brought to head for the Pareto front along iterations. Basically, a maximum number of evaluations is still given as a stopping criterion but it remains crucial to give an estimation to how far from a (local) Pareto front the approximation set is. For multiobjective Bayesian optimization [101], the goal is to find at next iteration the point that maximizes the hyperarea difference between old non-dominated set of points and the new one. The performance indicator is directly embedded into the algorithm and could be used as a stopping criterion. For evolutionary algorithms, surveys on stopping criteria for multiobjective optimization can be found in [187, 255]. The approach is to measure the progression of the current population combining performance indicators (hypervolume, MDR, etc.) and statistic tools (Kalman filter [187], χ^2-variance test [256], etc.) These last ones enable to detect a stationary state reached by the evolving population.

We believe that the use of monotonic performance indicators or binary ones that capture the dominance property seems to be the most efficient one in the years to come to follow the behavior of population-based algorithms along iterations.

4.4.4 Distribution and spread

The choice of spread and distribution indicators has only a sense when one wants to measure the distribution of points in the objective space, no matter how close from the Pareto front the approximated set is. Spread and distribution metrics can put forward global properties (for example statistics on the distribution of the points or extent of the front) or local properties such as the largest distance between closest non-dominated points that can be used to conduct search such as Γ indicator. Typically, the construction of a distribution or spread indicator requires two steps. The first consists in defining a distance between two points in the objective

space. Many distribution indicators in the literature use minimum Euclidean or Manhattan distance between points such as the SP metric (4.3.3), the Δ index (4.3.3), HRS (4.3.3), and so on. The DM (4.3.3) and Γ-metric (4.3.3) indicators use a "sorting distance"; I_D (4.3.3) a "neighborhood distance" based on a spanning tree, and so on. Once this is done, many of the existing distribution indicators are built by using statistic tools on this distance: mean (Δ (4.3.3), U measure (4.3.3), DM (4.3.3) for example), mean square (SP (4.3.3), D_{nc} (4.3.3)), and so on.

To use a distribution or spread indicator, it should satisfy the following properties:

1. The support of scaled functions, which enables to compare all objectives in an equivalent way (DM (4.3.3), OS (4.3.3), I_{OD} (4.3.3), Δ (4.3.3), Γ (4.3.3)).

2. For piecewise continuous or discontinuous Pareto front approximations, a good distribution indicator should not be based on the distance between closest neighbors, as it can hide some holes [271]. Some indicators possess this property such as DM (4.3.3), Γ (4.3.3), Δ (4.3.3), E_s (4.3.3) or evenness indicators (4.3.3).

3. Distribution and spread performance indicators should not be based on external parameters, such as Zitzler's metric $M_2^\star$ (4.3.3), UD (4.3.3), or entropy measure (4.3.3).

4. An easy interpretation: a value returned by an indicator has to be "intuitive" to understand. For example, the binary uniformity (4.3.3) is extremely difficult to interpret and should not be used. This remark applies for all types of performance indicators.

One could directly include spread control parameters in the design of new algorithms. The Normal Boundary Intersection method [80] controls the spread of a Pareto front approximation. This method is also used in the context of blackbox optimization [28].

4.5 Discussion

In this work, we give a review of performance indicators for the quality of Pareto front approximations in multiobjective optimization, as well as some usages of these indicators.

The most important application of performance indicators is to allow comparison and analysis of results of different algorithms. In this optic, among all these indicators, the hypervolume indicator and its binary counterpart, the hyperarea difference can be considered until now as the most relevant. The hypervolume indicator possesses good mathematical properties, it can capture dominance properties and distribution and does not require the knowledge of the Pareto front. Empirical studies [151, 204] have confirmed its efficiency compared to

other performance indicators. That is why it has been deeply used in the evolutionary community [218]. However, it has some limitations: the exponential cost as the number of objectives increases and the choice of the reference point. To compare algorithms, it can be replaced with other indicators capturing lower dominance relation such as dominance move, G-metric, binary ϵ-indicator, Hypervolume Sharpe-Ratio indicator, modified inverted generated distance or degree of approximation whose computational cost is less important.

Future research can focus on the discovery of new performance indicators that correct some drawbacks of the hypervolume indicator but keeps its good properties, and the integration of performance indicators directly into algorithms for multiobjective optimization.

Acknowledgments This research was financed by Le Digabel's NSERC discovery grant RGPIN-2018-05286, and also by the NSERC CRD RDCPJ 490744-15 grant and an InnovÉÉ grant, both in collaboration with Hydro-Québec and Rio Tinto. We also would like to thank the anonymous reviewers for their suggestions and comments which helped to greatly improve this work.

4.6 A summary of performance indicators

Table 4.3 draws a summary of all indicators described in Section 4.3. Most of complexity cost indications for computing indicators are drawn from [151]. $Y_P \subseteq \mathcal{Y}_P$ corresponds to the *Pareto optimal solution set* and Y_N is a Pareto front approximation returned by a given algorithm. The symbol "✗" indicates that the performance indicator does not satisfy the monotony property. The "-" symbol corresponds to binary indicators, for which monotonicity has no meaning.

Table 4.3 A summary of performance indicators.

Category	Performance indicators	Section	Symbol	Parameters	Comparison sets	Computational complexity	Monotone
Cardinality	C-metric/Two sets Coverage [279]	4.3.1	C	None	Binary indicator	$\mathcal{O}(m \ \lvert Y_N^1\rvert \times \lvert Y_N^2\rvert)$	-
4.3.1	Error ratio [249]	4.3.1	ER	None	Pareto front $\mathcal{Y}_P$	Low	✗
	Generational non dominated vector generation [250]	4.3.1	$GNVG$	None	None	Low	✗
	Generational non dominated vector generation ratio [250]	4.3.1	$GNVGR$	None	Pareto front Y_P	Low	✗
	Mutual domination rate [187]	4.3.1	MDR	None	None	Low	✗
	Nondominated vector additional [250]	4.3.1	NVA	None	None	Low	✗
	Overall nondominated vector generation [249]	4.3.1	$ONVG$	None	None	Low	✗
	Overall nondominated vector generation ratio [249]	4.3.1	$ONVGR$	None	Pareto front Y_P	Low	✗
	Ratio of non-dominated points by the reference set [139]	4.3.1	C_{2R}	None	Reference set Y_R	$\mathcal{O}(m \ \lvert Y_N\rvert \times \lvert Y_R\rvert)$	✗
	Ratio of the reference points [139]	4.3.1	C_{1R}	None	Reference set Y_R	$\mathcal{O}(m \ \lvert Y_N\rvert \times \lvert Y_R\rvert)$	✗
Convergence	Degree of Approximation [93]	4.3.2	DOA	None	Pareto front Y_P	$\mathcal{O}(m \ \lvert Y_N\rvert \times \lvert Y_P\rvert)$	Not strictly
4.3.2	ϵ-family [280]	4.3.2	I_ϵ	None	Pareto front Y_P	$\mathcal{O}(m \ \lvert Y_N\rvert \times \lvert Y_P\rvert)$	Not strictly
	Generational distance [249]	4.3.2	GD	None	Pareto front Y_P	$\mathcal{O}(m \ \lvert Y_N\rvert \times \lvert Y_P\rvert)$	✗
	γ-metric [84]	4.3.2	γ	None	Pareto front Y_P	$\mathcal{O}(m \ \lvert Y_N\rvert \times \lvert Y_P\rvert)$	✗
	Maximum Pareto front error [249]	4.3.2	$MPFE$	None	Pareto front Y_P	$\mathcal{O}(m \ \lvert Y_N\rvert \times \lvert Y_P\rvert)$	✗
	$M_1^\star$-metric [276]	4.3.2	$M_1^\star$	None	Pareto front Y_P	$\mathcal{O}(m \ \lvert Y_N\rvert \times \lvert Y_P\rvert)$	✗
	Progression metric [249]	4.3.2	-	None	None	$\mathcal{O}(m \ \lvert Y_N\rvert)$	✗
	Seven points average distance [226]	4.3.2	$SPAD$	None	Reference set Y_R	$\mathcal{O}(m \ \lvert Y_N\rvert)$	✗
	Standard deviation from the Generational distance [249]	4.3.2	$STDGD$	None	Pareto front Y_P	$\mathcal{O}(m \ \lvert Y_N\rvert \times \lvert Y_P\rvert)$	✗
Distribution and spread	Cluster [263]	4.3.3	CL_μ	A parameter μ	None	High	✗
	Δ-index [84]	4.3.3	Δ	None	Pareto front Y_P	$\mathcal{O}(m \ \lvert Y_N\rvert^2 + m \ \lvert Y_N\rvert \times \lvert Y_P\rvert)$	✗
4.3.3	Δ'-index [84]	4.3.3	Δ'	None	None	$\mathcal{O}(m \ \lvert Y_N\rvert^2)$	✗
	$\Delta^\star$ spread metric [272]	4.3.3	$\Delta^\star$	None	Pareto front Y_P	$\mathcal{O}(m \ \lvert Y_N\rvert^2 + m \ \lvert Y_N\rvert \times \lvert Y_P\rvert)$	✗
	Distribution metric [271]	4.3.3	DM	None	None	$\mathcal{O}(m \ \lvert Y_N\rvert^2)$	✗
	Diversity comparison indicator [172]	4.3.3	DCI	A parameter div	k-ary indicator comparing $Y_N^1, Y_N^2, \ldots, Y_N^k$ non-dominated sets	$\mathcal{O}\left(m \ (k \ \lvert Y_N^{\max}\rvert)^2\right)$	✗
	Diversity indicator [56]	4.3.3	DIR	None	Reference set Y_R	$\mathcal{O}(m \ \lvert Y_N\rvert \times \lvert Y_R\rvert)$	✗

Table 4.3 A summary of performance indicators.

Category	Performance indicators	Section	Symbol	Parameters	Comparison sets	Computational complexity	Monotone
	Entropy metric [108]	4.3.3	-	A parameter $grids$	None	High	✗
	Evenness [189]	4.3.3	ξ	None	None	$\mathcal{O}(m\ \|Y_N\|^2)$	✗
	Extension [188]	4.3.3	EX	None	Pareto front Y_P	$\mathcal{O}(m\ \|Y_N\| \times \|Y_P\|)$	✗
	Γ-metric [74]	4.3.3	Γ	None	None	$\mathcal{O}(m\ \|Y_N\|^2)$	✗
	Hole Relative Size [64]	4.3.3	HRS	None	None	$\mathcal{O}(m\ \|Y_N\|^2)$	✗
	Laumanns metric [166]	4.3.3	-	None	None	$\mathcal{O}(\|Y_N\|^{\frac{m}{3}} \operatorname{poly}\log\|Y_N\|)$	✗
	Modified Diversity indicator [8]	4.3.3	$M{-}DI$	A parameter δ	Reference set Y_R	$\mathcal{O}(m\ \|Y_N\|^2 \times \|Y_R\|)$	
	$M_2^\star$-metric [276]	4.3.3	$M_2^\star$	Niche radius σ	None	$\mathcal{O}(m\ \|Y_N\|^2)$	✗
	$M_3^\star$-metric [276]	4.3.3	$M_3^\star$	None	None	$\mathcal{O}(m\ \|Y_N\|^2)$	✗
	Number of distinct choices [263]	4.3.3	NDC_μ	A parameter μ	None	High	✗
	Outer diameter [277]	4.3.3	I_{OD}	None	None	$\mathcal{O}(m\ \|Y_N\|)$	✗
	Overall Pareto Spread [263]	4.3.3	OS	None	$\tilde{\mathbf{y}}^I$ and $\tilde{\mathbf{y}}^M$	$\mathcal{O}(m\ \|Y_N\|)$	✗
	Riesz S-energy [140]	4.3.3	E_S	A parameter s	None	$\mathcal{O}(m\ \|Y_N\|^2)$	✗
	Sigma diversity metric [195]	4.3.3	σ	A parameter $lines$	None	High	✗
	Spacing [226]	4.3.3	SP	None	None	$\mathcal{O}(m\ \|Y_N\|^2)$	✗
	U-measure [169]	4.3.3	U	None	None	$\mathcal{O}(m\ \|Y_N\|^2)$	✗
	Uniform assessment metric [176]	4.3.3	I_D	None	None	$\mathcal{O}(m\ \|Y_N\|^2)$	✗
	Uniform distribution [239]	4.3.3	UD	Niche radius σ	None	$\mathcal{O}(m\ \|Y_N\|^2)$	✗
	Uniformity [225]	4.3.3	δ	None	None	$\mathcal{O}(m\ \|Y_N\|^2)$	✗
	Uniformity [188]	4.3.3	-	None	Binary	Quadratic	✗
Convergence and distribution	Averaged Hausdorff distance [227]	4.3.4	Δ_q	None	Pareto front Y_P	$\mathcal{O}(m\ \|Y_N\| \times \|Y_P\|)$	✗
	Cone-based hypervolume [100]	4.3.4	-	Angle γ and Reference point $\mathbf{r}$	None	$\mathcal{O}(\|Y_N\|^{\frac{m}{3}} \operatorname{poly}\log\|Y_N\|)$	Strictly
4.3.4	Dominance move [174]	4.3.4	DoM	None	Binary indicator	$\mathcal{O}(\|Y_N\| \log\|Y_N\|)$	-
	D-metric/Difference coverage of two sets [274]	4.3.4	-	Reference point $\mathbf{r}$	Binary indicator	$\mathcal{O}(\|Y_N\|^{\frac{m}{3}} \operatorname{poly}\log\|Y_N\|)$	-
	D_R-metric [78]	4.3.4	-	None	Reference set Y_R	$\mathcal{O}(m\ \|Y_N\| \times \|Y_R\|)$	Not strictly
	Hyperarea difference [263]	4.3.4	HD	Reference point $\mathbf{r}$	None	$\mathcal{O}(\|Y_N\|^{\frac{m}{3}} \operatorname{poly}\log\|Y_N\|)$	Strictly
	Hypervolume indicator (or S-metric) [276]	4.3.4	HV	Reference point $\mathbf{r}$	None	$\mathcal{O}(\|Y_N\|^{\frac{m}{3}} \operatorname{poly}\log\|Y_N\|)$	Strictly
	Hypervolume Sharpe-ratio indicator [265]	4.3.4	I_{HSR}	Reference points $\mathbf{y}^l$ and $\mathbf{y}^u$	None	Polynomial	Not strictly
	Inverted generational distance [62]	4.3.4	IGD	None	Pareto front Y_P	$\mathcal{O}(m\ \|Y_N\| \times \|Y_P\|)$	✗
	Inverted generation distance with non contributed solutions detection [243]	4.3.4	$IGD{-}NS$	None	Pareto front Y_P	$\mathcal{O}(m\ \|Y_N\| \times \|Y_P\|)$	✗
	G-metric [184]	4.3.4	-	None	k-ary indicator comparing $Y_N^1, Y_N^2, \ldots, Y_N^k$ non-dominated sets	$\mathcal{O}(k^3\ \|Y_N^{\max}\|^2)$	Not strictly
	Logarithmic hypervolume indicator [123]	4.3.4	$\log HV$	Reference point $\mathbf{r}$	None	$\mathcal{O}(\|Y_N\|^{\frac{m}{3}} \operatorname{poly}\log\|Y_N\|)$	Strictly
	Modified inverted generational distance [147]	4.3.4	IGD^+	None	Pareto front Y_P	$\mathcal{O}(m\ \|Y_N\| \times \|Y_P\|)$	Not strictly
	Performance comparison indicator [173]	4.3.4	PCI	σ distance	k-ary indicator comparing $Y_N^1, Y_N^2, \ldots, Y_N^k$ non-dominated sets	Quadratic	Not strictly
	p,q-averaged distance [251]	4.3.4	$\Delta_{p,q}$	None	Pareto front Y_P	Quadratic	✗
	R-metric [139]	4.3.4	Y_R	A set W of weights vectors	Reference set Y_R	$\mathcal{O}(m\ \|Y_N\| \times \|Y_R\| \times \|W\|)$	Not strictly

4.7 Compatibility and completeness

Tables 4.4 and 4.5 summarize compatibility and completeness properties. Only the strongest relationships are kept. Some of them are drawn from [280]. All spread and distribution indicators are not compatible with Pareto front approximation relations.

Table 4.4 Compatibility and completeness of unary performance indicators.

Category	Performance indicators	Section	Symbol	Boolean function	Compatible	Complete
Cardinality	Error ratio [249]	4.3.1	ER	$ER(Y_N^1) < ER(Y_N^2)$	✗	✗
4.3.1	Generational non dominated vector generation [250]	4.3.1	$GNVG$	-	-	-
	Generational non dominated vector generation ratio [250]	4.3.1	$GNVGR$	-	-	-
	Mutual domination rate [187]	4.3.1	MDR	-	-	-
	Nondominated vector additional [250]	4.3.1	NVA	-	-	-
	Overall nondominated vector generation [249]	4.3.1	$ONVG$	$ONVG(Y_N^1) > ONVG(Y_N^2)$	✗	✗
	Overall nondominated vector generation ratio [249]	4.3.1	$ONVGR$	$ONVGR(Y_N^1;Y_P) > ONVGR(Y_N^2;Y_P)$	✗	✗
	Ratio of non-dominated points by the reference set [139]	4.3.1	C_{2R}	$C_{2R}(Y_N^1;Y_R) > C_{2R}(Y_N^2;Y_R)$	✗	✗
	Ratio of the reference points [139]	4.3.1	C_{1R}	$C_{1R}(Y_N^1;Y_R) > C_{1R}(Y_N^2;Y_R)$	✗	✗
Convergence	Degree of Approximation [93]	4.3.2	DOA	$DOA(Y_N^1;Y_P) < DOA(Y_N^2;Y_P)$	Not better than	$\prec$
4.3.2	Generational distance [249]	4.3.2	GD	$GD(Y_N^1;Y_P) < GD(Y_N^2;Y_P)$	✗	✗
	γ-metric [84]	4.3.2	γ	$\gamma(Y_N^1;Y_P) < \gamma(Y_N^2;Y_P)$	✗	✗
	Maximum Pareto front error [249]	4.3.2	$MPFE$	$MPFE(Y_N^1;Y_P) < MPFE(Y_N^2;Y_P)$	✗	✗
	$M_1^\star$-metric [276]	4.3.2	$M_1^\star$	$M_1^\star(Y_N^1;Y_P) < M_1^\star(Y_N^2;Y_P)$	✗	✗
	Progression metric [249]	4.3.2	-	-	-	-
	Seven points average distance [226]	4.3.2	$SPAD$	$SPAD(Y_N^1;Y_P) < SPAD(Y_N^2;Y_P)$	✗	✗
	Standard deviation from the Generational distance [249]	4.3.2	$STDGD$	-	-	-
Distribution	Cluster [263]	4.3.3	CL_μ	-	-	-
and spread	Δ-index [84]	4.3.3	Δ	$\Delta(Y_N^1;Y_P) < \Delta(Y_N^2;Y_P)$	✗	✗
4.3.3	Δ'-index [84]	4.3.3	Δ'	$\Delta'(Y_N^1) < \Delta'(Y_N^2)$	✗	✗
	$\Delta^\star$ spread metric [272]	4.3.3	$\Delta^\star$	$\Delta^\star(Y_N^1;Y_P) < \Delta^\star(Y_N^2;Y_P)$	✗	✗
	Distribution metric [271]	4.3.3	DM	$DM(Y_N^1) < DM(Y_N^2)$	✗	✗
	Diversity indicator [56]	4.3.3	DIR	$DIR(Y_N^1) < DIR(Y_N^2)$	✗	✗
	Entropy metric [108]	4.3.3	-	-	-	-
	Evenness [189]	4.3.3	ξ	$\xi(Y_N^1) < \xi(Y_N^2)$	✗	✗
	Extension [188]	4.3.3	EX	$EX(Y_N^1;Y_P) < EX(Y_N^2;Y_P)$	✗	✗
	Γ-metric [74]	4.3.3	Γ	$\Gamma(Y_N^1) < \Gamma(Y_N^2)$	✗	✗
	Hole Relative Size [64]	4.3.3	HRS	$HRS(Y_N^1) < HRS(Y_N^2)$	✗	✗
	Laumanns metric [166]	4.3.3	-	$I_L(Y_N^1) > I_L(Y_N^2)$	✗	✗
	Modified Diversity indicator [8]	4.3.3	$M{-}DI$	$M{-}DI(Y_N^1;Y_R) > M{-}DI(Y_N^2;Y_R)$	✗	✗
	$M_2^\star$-metric [276]	4.3.3	$M_2^\star$	$M_2^\star(Y_N^1;\sigma) > M_2^\star(Y_N^2;\sigma)$	✗	✗
	$M_3^\star$-metric [276]	4.3.3	$M_3^\star$	$M_3^\star(Y_N^1) > M_3^\star(Y_N^2)$	✗	✗
	Number of distinct choices [263]	4.3.3	NDC_μ	$NDC_\mu(Y_N^1) > NDC_\mu(Y_N^2)$	✗	✗
	Outer diameter [277]	4.3.3	I_{OD}	$I_{OD}(Y_N^1) > I_{OD}(Y_N^2)$	✗	✗
	Overall Pareto Spread [263]	4.3.3	OS	$OS(Y_N^1) > OS(Y_N^2)$	✗	✗
	Riesz S-energy [140]	4.3.3	E_S	$E_S(Y_N^1) < E_S(Y_N^2)$	✗	✗
	Sigma diversity metric [195]	4.3.3	σ	$\sigma(Y_N^1;d) > \sigma(Y_N^2;d)$	✗	✗
	Spacing [226]	4.3.3	SP	$SP(Y_N^1) < SP(Y_N^2)$	✗	✗
	U-measure [169]	4.3.3	U	$U(Y_N^1) < U(Y_N^2)$	✗	✗
	Uniform assessment metric [176]	4.3.3	I_D	$I_D(Y_N^1) > I_D(Y_N^2)$	✗	✗
	Uniform distribution [239]	4.3.3	UD	$UD(Y_N^1;\sigma) < UD(Y_N^2;\sigma)$	✗	✗
	Uniformity [225]	4.3.3	δ	$\delta(Y_N^1) < \delta(Y_N^2)$	✗	✗
Convergence	Averaged Hausdorff distance [227]	4.3.4	Δ_q	$\Delta_q(Y_N^1;Y_P) < \Delta_q(Y_N^2;Y_P)$	✗	✗

Table 4.4 Compatibility and completeness of unary performance indicators.

Category	Performance indicators	Section	Symbol	Boolean function	Compatible	Complete
and distribution 4.3.4	Cone-based hypervolume [100]	4.3.4	-	$\chi(Y_N^1) > \chi(Y_N^2)$	Not better than	$\lhd$
	D_R-metric [78]	4.3.4	-	$D_R(Y_N^1;Y_R) < D_R(Y_N^2;Y_R)$	Not better than	$\prec\prec$
	Hyperarea difference [263]	4.3.4	HD	$HD(Y_N^1) < HD(Y_N^2)$	Not better than	$\lhd$
	Hypervolume indicator (or S-metric) [276]	4.3.4	HV	$HV(Y_N^1;\mathbf{r}) > HV(Y_N^2;\mathbf{r})$	Not better than	$\lhd$
	Hypervolume Sharpe-ratio indicator [265]	4.3.4	I_{HSR}	$I_{HSR}(Y_N^1;\mathbf{y}^l,\mathbf{y}^u) > I_{HSR}(Y_N^2;\mathbf{y}^l,\mathbf{y}^u)$	Not better than	$\prec$
	Inverted generational distance [62]	4.3.4	IGD	$IGD(Y_N^1;Y_P) < IGD(Y_N^2;Y_P)$	✗	✗
	Inverted generation distance with non contributed solutions detection [243]	4.3.4	IGD–NS	IGD–$NS(Y_N^1;Y_P) < IGD$–$NS(Y_N^2;Y_P)$	✗	✗
	Logarithmic hypervolume indicator [123]	4.3.4	$\log HV$	$\log HV(Y_N^1;\mathbf{r}) > \log HV(Y_N^2;\mathbf{r})$	Not better than	$\lhd$
	Modified inverted generational distance [147]	4.3.4	IGD^+	$IGD^+(Y_N^1;Y_P) < IGD^+(Y_N^2;Y_P)$	Not better than	$\preceq$
	p,q-averaged distance [251]	4.3.4	$\Delta_{p,q}$	$\Delta_{p,q}(Y_N^1;Y_P) < \Delta_{p,q}(Y_N^1;Y_P)$	✗	✗

Table 4.5 Compatibility and completeness of binary performance indicators (inspired by [280]): a - means there is no comparison method which is complete and compatible for the given relation, a ✗ that the indicator is not even monotone.

Category	Performance indicators	Section	Symbol	Relation			
				$\lhd$	$\preceq$	$=$	$\parallel$
Cardinality 4.3.1	C-metric/Two sets Coverage [279]	4.3.1	C	$C(Y_N^1,Y_N^2)=1$ $C(Y_N^2,Y_N^1)<1$	$C(Y_N^1,Y_N^2)=1$	$C(Y_N^1,Y_N^2)=1$ $C(Y_N^2,Y_N^1)=1$	$C(Y_N^1,Y_N^2)>1$ $C(Y_N^2,Y_N^1)>1$
Convergence 4.3.2	Additive ϵ-indicator [280]	4.3.2	I_ϵ	$I_\epsilon(Y_N^1,Y_N^2)\leq 0$ $I_\epsilon(Y_N^2,Y_N^1)>0$	$I_\epsilon(Y_N^1,Y_N^2)\leq 0$	$I_\epsilon(Y_N^1,Y_N^2)=0$ $I_\epsilon(Y_N^2,Y_N^1)=0$	$I_\epsilon(Y_N^1,Y_N^2)>0$ $I_\epsilon(Y_N^2,Y_N^1)>0$
Distribution and spread	Diversity comparison indicator [172]	4.3.3	DCI	✗	✗	✗	✗
4.3.3	Uniformity [188]	4.3.3	-	✗	✗	✗	✗
Convergence and	Dominance move [174]	4.3.4	DoM	$DoM(Y_N^1,Y_N^2)=0$ $DoM(Y_N^2,Y_N^1)>0$	$DoM(Y_N^1,Y_N^2)=0$ $DoM(Y_N^2,Y_N^1)\geq 0$	$DoM(Y_N^1,Y_N^2)=0$ $DoM(Y_N^2,Y_N^1)=0$	$DoM(Y_N^1,Y_N^2)>0$ $DoM(Y_N^2,Y_N^1)>0$
distribution 4.3.4	D-metric/Difference coverage of two sets [274]	4.3.4	-	$D(Y_N^1,Y_N^2)>0$ $D(Y_N^2,Y_N^1)=0$	$D(Y_N^1,Y_N^2)\geq 0$ $D(Y_N^2,Y_N^1)=0$	$D(Y_N^1,Y_N^2)=0$ $D(Y_N^2,Y_N^1)=0$	$D(Y_N^1,Y_N^2)>0$ $D(Y_N^2,Y_N^1)>0$
	G-metric [184]	4.3.4	-	-	-	-	-
	Performance comparison indicator [173]	4.3.4	PCI	-	-	-	-
	R-metric [139]	4.3.4	R	-	-	-	-

CHAPTER 5 ARTICLE 2: DMULTI-MADS: MESH ADAPTIVE DIRECT MULTISEARCH FOR BOUND-CONSTRAINED BLACKBOX MULTIOBJECTIVE OPTIMIZATION

Nanos gigantum umeris insidentes

Jean De Salisbury, *Metalogicon*

Authors Jean Bigeon, Sébastien Le Digabel and Ludovic Salomon

Journal Published in *Computational Optimization and Applications*

Abstract The context of this research is multiobjective optimization where conflicting objectives are present. In this work, these objectives are only available as the outputs of a blackbox for which no derivative information is available. This work proposes a new extension of the mesh adaptive direct search (MADS) algorithm to multiobjective derivative-free optimization with bound constraints. This method does not aggregate objectives and keeps a list of non dominated points which converges to a (local) Pareto set as long as the algorithm unfolds. As in the single-objective optimization MADS algorithm, this method is built around a search step and a poll step. Under classical direct search assumptions, it is proved that the so-called DMulti-MADS algorithm generates multiple subsequences of iterates which converge to a set of local Pareto stationary points.

Finally, computational experiments suggest that this approach is competitive compared to the state-of-the-art algorithms for multiobjective blackbox optimization.

Keywords Multiobjective optimization, derivative-free optimization, blackbox optimization, mesh adaptive direct search, Clarke analysis.

5.1 Introduction

This work considers the following multiobjective optimization problem:

$$MOP : \min_{\mathbf{x}\in\Omega} f(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x}))^\top$$

where $\Omega = [\mathbf{l}; \mathbf{u}]$ is the *feasible decision set* and $\mathbf{l}, \mathbf{u} \in \mathbb{R}^n$ with $\mathbf{l}_i < \mathbf{u}_i$ for $i = 1, 2, \ldots, n$. The functions $f_i : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ for $i = 1, 2, \ldots, m \geq 2$, are the outputs of a blackbox, which means that no analytical form is known. Derivatives are not available so that gradient-based techniques cannot be considered. Allowing f to take infinity values refers to the possibility that evaluations of f can fail. In these cases, blackbox or derivative-free optimization techniques [20, 67] are particularly relevant. The mapping of Ω by the objective function f is designed as the *feasible objective set*. The sets $\mathbb{R}^n$ and $\mathbb{R}^m$ are denoted the *decision space* and the *objective space*, respectively.

The goal is then to find the best set of trade-off solutions in the objective space, named as the Pareto front, given a finite budget of functions evaluations. These solutions can then be presented to the decision maker, who can decide of the most adequate design dependently for her/his problem [63, 89, 98].

Multiobjective heuristics such as evolutionary/genetic algorithms [85] or particule-swarm optimization [142] are commonly used. However, they do not possess mathematical convergence background and require a significantly large amount of functions evaluations, which is not affordable in this research context where problems involve costly blackbox functions. This last limitation has partly been removed: by using cheaper surrogate models such as radial basis functions [196] or krigging metamodeling [158], one can identify the most promising points to be evaluated with the true objective function.

Among supported convergence-based methods, a first approach is to aggregate all objective functions in one parameterized single-objective formulation. By solving the resulting problem with convergence-proved derivative-free techniques, one is able to get a local optimal Pareto solution. The procedure can be used again to obtain different Pareto solutions by changing the parameters of the current formulation. The BiMADS [27] and MultiMADS [28] methods follow these approaches. They are both based on the Mesh Adaptive Direct Search (MADS) algorithm [15] for single-objective constrained optimization. BiMADS is only designed for biobjective problems contrary to MultiMADS that takes into account more objectives. Several issues are raised with such scalarization-based methods. A first one is the number of evaluations to allocate to each single-objective problem: too few and no promising points can be found; too many and the algorithm can lack budget to explore potential promising zones in the objective space. A second drawback is the large amount of evaluations that the user can fix to obtain points close to the Pareto front. Function evaluations could rather be used to enrich the non dominated set of points constituting the approximated Pareto front returned by the algorithm.

Recently, new convergence-proved methods for derivative-free multiobjective optimization

have emerged, which keep a population of non dominated points that gets closer to the Pareto front as long as the algorithm unfolds [74,126,180]. To the best of our knowledge, [74] is the first to propose the DMS framework that extends single-objective direct search algorithms to multiobjective optimization. They prove the existence of at least one subsequence of iterates that converges to a local Pareto point. This approach is used again to build constrained line-search methods for multiobjective optimization [180], implicit-filtering methods for multiobjective optimization [126] and derivative-free trust-region methods for biobjective optimization [223].

Inspired by the works of [74] and [180], this work proposes a new way to extend the MADS algorithm to nonsmooth constrained multiobjective optimization with alternative[1] convergence results than the DMS algorithm. More precisely, under mild assumptions, the DMS algorithm generates at least a sequence of points which converges to a local Pareto optimal solution. This research goes a step further. It is proved that under the same assumptions, the proposed method generates sequences of points which converge to a set of Pareto stationary points. This result is equally stronger than the one proposed in [180], as their proof requires that their objective functions be Lipschitz continuous.

This work is organized as follows. Section 5.2 introduces the notations and definitions relative to multiobjective optimization. Section 5.3 summarizes the core elements of the MADS algorithm. Section 5.4 presents the new extension of the MADS algorithm to multiobjective optimization. Section 5.5 is dedicated to the convergence analysis of the proposed method. Finally, Section 5.6 reports computational experiments and discussions, followed by the conclusion.

5.2 Multiobjective optimization and Pareto dominance

In order to compare objective vectors, the following relation order is used [98]:

$$\forall (\mathbf{y}^1, \mathbf{y}^2) \in (\mathbb{R}^m)^2 \,,\; \mathbf{y}^1 \leq \mathbf{y}^2 \Leftrightarrow \mathbf{y}^2 - \mathbf{y}^1 \in \mathbb{R}^m_+ \Leftrightarrow \forall i = 1, 2, \ldots, m,\; \mathbf{y}^1_i \leq \mathbf{y}^2_i.$$

The relation notations $<$, $>$ and $\geq$ are similarly defined according to the cone $\mathbb{R}^m_+$.
The concept of Pareto dominance can now be introduced:

Definition 32. Given two decision vectors $\mathbf{x}^1$ and $\mathbf{x}^2$ in the feasible decision set, it is said that:

- $\mathbf{x}^1 \preceq \mathbf{x}^2$ ($\mathbf{x}^1$ weakly dominates $\mathbf{x}^2$) if and only $\forall i = 1, 2, \ldots, m,\; f_i(\mathbf{x}^1) \leq f_i(\mathbf{x}^2)$.

[1] We replace the term "stronger" by alternative.

- $\mathbf{x}^1 \prec \mathbf{x}^2$ ($\mathbf{x}^1$ dominates $\mathbf{x}^2$) if and only $\forall i = 1, 2, \ldots, m$, $f_i(\mathbf{x}^1) \leq f_i(\mathbf{x}^2)$ and it exists at least one index i_0 such that $f_{i_0}(\mathbf{x}^1) < f_{i_0}(\mathbf{x}^2)$.
- $\mathbf{x}^1 \sim \mathbf{x}^2$ ($\mathbf{x}^1$ and $\mathbf{x}^2$ are indifferent) if $\mathbf{x}^1$ does not dominate $\mathbf{x}^2$ and $\mathbf{x}^2$ does not dominate $\mathbf{x}^1$.

This definition is illustrated in Figure 5.1 for a biobjective minimization problem in the feasible objective set which is a subset of $\mathbb{R}^2$ delimited by the closed curve. Depending on the $\mathbf{x}^1$ point, three zones in the objective space are considered. The dominance zone is the set of feasible points which dominate $\mathbf{x}^1$. The dominated zone is the set of feasible points which are dominated by $\mathbf{x}^1$. The indifference zone is the set of points which are indifferent to $\mathbf{x}^1$. In this case, $\mathbf{x}^4 \prec \mathbf{x}^1$, $\mathbf{x}^1 \prec \mathbf{x}^2$ and $\mathbf{x}^3 \sim \mathbf{x}^1$.

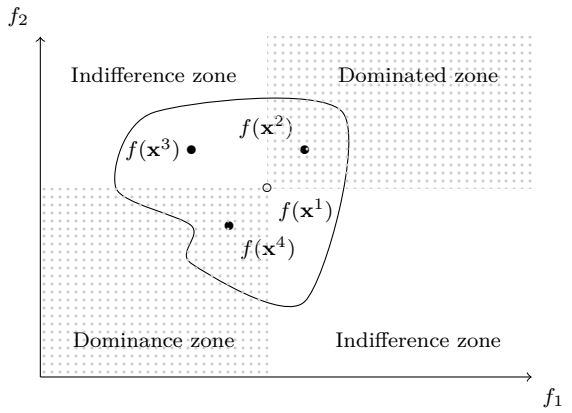


Figure 5.1 An illustration of the Pareto dominance for a minimization biobjective problem. $\mathbf{x}^4 \prec \mathbf{x}^1$, $\mathbf{x}^1 \prec \mathbf{x}^2$, $\mathbf{x}^4 \prec \mathbf{x}^2$ and $\mathbf{x}^3 \sim \mathbf{x}^4$.

The above definition enables to define optimality for multiobjective optimization problems.

Definition 33 (Global Pareto optimal solution)**.** A point $\mathbf{x}^* \in \Omega$ is a global Pareto optimal solution of MOP if there does not exist any other point $\mathbf{x} \in \Omega$ such that $\mathbf{x} \prec \mathbf{x}^*$.

Definition 34 (Local Pareto optimal solution)**.** A point $\mathbf{x}^* \in \Omega$ is a local Pareto optimal solution of MOP if there does not exist any other point $\mathbf{x} \in \Omega \cap \mathcal{N}(\mathbf{x}^*)$ such that $\mathbf{x} \prec \mathbf{x}^*$, where $\mathcal{N}(\mathbf{x}^*)$ is a neighbourhood of $\mathbf{x}^*$.

The set of global Pareto optimal solutions in the feasible decision set Ω is called the *Pareto set* denoted by $\mathcal{X}_P$ and its mapping by the objective function f is the *Pareto front* denoted by $\mathcal{Y}_P$. The image of a set of local Pareto optimal points is called a *local Pareto front*.

The Pareto set is usually composed of many elements [98], which cannot be all enumerated. Solving a multiobjective optimization problem aims at finding a good representative subset of the Pareto front [225]. It is then convenient to introduce the concept of *Pareto approximation set* [280].

Definition 35 (Pareto set and front approximation)**.** A set of vectors X_N in the feasible decision set Ω is called a Pareto set approximation if no element of this set is dominated by another. Its image in the objective space is called a Pareto front approximation.

All elements of a Pareto set approximation have to be non dominated relatively to each other. A Pareto set approximation should ideally contain elements of the Pareto set. The algorithm described in this work guarantees convergence towards a Pareto set approximation whose elements are locally Pareto optimal.

Ideally, a Pareto set approximation should contain *extreme points of the Pareto set*. Extreme points of the Pareto set are decision vectors that are the solutions of each single-objective subproblem $\min_{\mathbf{x}\in\Omega} f_i(\mathbf{x})$ for $i = 1, 2, \ldots, m$, which are non dominated. With the knowledge of extreme Pareto points, one can get the *ideal objective vector* $\mathbf{y}^I$ of MOP, defined by

$$\mathbf{y}^I = \left(\min_{\mathbf{x}\in\Omega} f_1(\mathbf{x}), \min_{\mathbf{x}\in\Omega} f_2(\mathbf{x}), \ldots, \min_{\mathbf{x}\in\Omega} f_m(\mathbf{x}) \right).$$

Figure 5.2 illustrates these concepts.

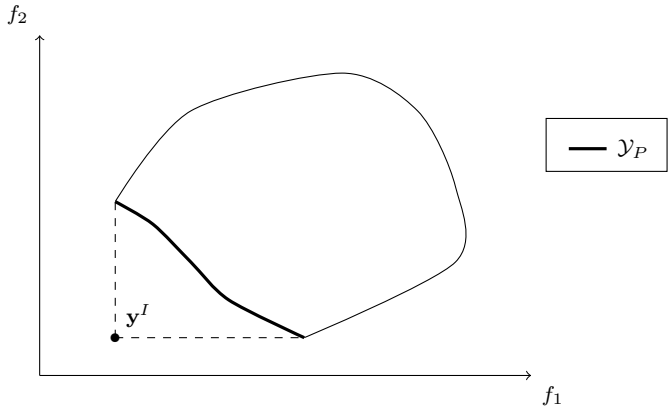


Figure 5.2 Objective space, Pareto front $\mathcal{Y}_P$ (represented in black) and ideal objective vector $\mathbf{y}^I$ for a minimization biobjective problem.

5.3 The MADS algorithm

The proposed algorithm is an extension of the MADS algorithm [15] to multiobjective optimization. It differs from the existing BiMADS and MultiMADS methods [27, 28], as it does not rely on a scalarization-based approach. This section summarizes the main steps of the MADS algorithm. All the definitions come from [20]. The reader is invited to consult [15] for more details.

The MADS algorithm is a direct search method initially designed to solve optimization single-objective optimization blackbox problems, i.e., $\min_{\mathbf{x}\in\Omega} f(\mathbf{x})$, with $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ a scalar-valued function and Ω the same subset of $\mathbb{R}^n$ as defined for MOP.

Definition 36. Let $\mathbf{G} \in \mathbb{R}^{n\times n}$ be a non-singular matrix and $\mathbf{Z} \in \mathbb{Z}^{n\times n_D}$ be such that the columns of $\mathbf{Z}$ form a positive spanning set for $\mathbb{R}^n$. Define $\mathbf{D} = \mathbf{GZ}$. At iteration k, the current *mesh of coarseness* $\delta^k > 0$, generated by D is defined by

$$M^k = \bigcup_{\mathbf{x}\in V^k} \left\{\mathbf{x} + \delta^k \mathbf{D}\mathbf{z} : \mathbf{z} \in \mathbb{N}^{n_D}\right\}$$

where V^k is the set of points already evaluated by the start of iteration k.

At each iteration k, MADS attempts to find a better point in the decision space, belonging to the current mesh defined above.

V^0 represents the set of starting points indicated by the user. The mesh is generated with a finite set of n_D directions $D \subset \mathbb{R}^n$ scaled with the mesh size parameter $\delta^k > 0$. Generally, one considers the positive spanning set $\mathbf{D} = [\mathbf{I}_n, -\mathbf{I}_n]$ where $\mathbf{I}_n$ is the identity matrix of size n but other choices are possible [15].

Each iteration is composed of two steps: the search and the poll. The search step enables the user to define its own search strategy as long as new evaluated points remain on the mesh M^k. If a better point is found during the search step, the poll step is not executed. As the convergence analysis depends on the poll step, it is more rigidly defined. It generates points on the mesh around an incumbent solution $\mathbf{x}^k \in V^k$. The poll points must belong to the frame of extent Δ^k centered at $\mathbf{x}^k$ defined below.

Definition 37. Let $\mathbf{G} \in \mathbb{R}^{n\times n}$ be a non-singular matrix and $\mathbf{Z} \in \mathbb{Z}^{n\times n_D}$ be such that the columns of $\mathbf{Z}$ form a positive spanning set for $\mathbb{R}^n$. Let $\delta^k > 0$ be the mesh size parameter and let Δ^k be such that $\delta^k \leq \Delta^k$. At iteration k, the frame of extent Δ^k generated by $\mathbf{D}$, centered at $\mathbf{x}^k$ is defined by

$$F^k = \left\{\mathbf{x} \in M^k : \|\mathbf{x} - \mathbf{x}^k\|_\infty \leq \Delta^k b\right\}$$

with $b = \max\{\| \mathbf{d}' \|_\infty : \mathbf{d}' \in \mathbf{D}\}$ and Δ^k is the frame size parameter such that $\delta^k \leq \Delta^k$.

The initial frame size parameter $\Delta^0 \in \mathbb{R}_+$ can be provided by the user or automatically fixed according to the bound constraints of the optimization problem or the coordinates of an initial starting point [18, 26].

The new candidates in the poll step must belong to the poll set P^k defined by

$$P^k = \left\{ \mathbf{x}^k + \delta^k \mathbf{d} : \mathbf{d} \in \mathbb{D}_\Delta^k \right\} \subset F^k$$

where $\mathbb{D}_\Delta^k$ is a positive spanning set of directions. To satisfy these properties, the authors of [15] propose the following relation between δ^k and Δ^k:

$$\delta^k = \min\left\{ \Delta^k, \left(\Delta^k\right)^2 \right\}.$$

Poll points can be evaluated opportunistically (as soon as a better solution is found, the poll step is interrupted) or completely (all candidates are evaluated) as it does not affect the convergence analysis. If a better point is found after the search and poll steps, the iteration is marked as successful. If not, it is considered as unsuccessful. In the first case, the frame size parameter is increased or kept constant. In the second case, the frame size parameter is reduced, increasing the mesh resolution and reducing the exploration field around the current incumbent point $\mathbf{x}^k$. Directions $\mathbf{d} \in \mathbb{D}_\Delta^k$ may be generated according to the OrthoMADS instantiation of MADS [2]. Algorithm 5 summarizes the main steps of MADS. More details can be found in [16, 26].

Under mild assumptions, the MADS convergence analysis provided in [15] guarantees the existence of an accumulation point $\hat{\mathbf{x}}$ such that its Clarke generalized derivative $f^0(\hat{\mathbf{x}}; \mathbf{d})$ is non negative [59] for all the directions $\mathbf{d} \in \mathbb{R}^n$ belonging to the hypertangent cone $\mathcal{T}_\Omega^H(\hat{\mathbf{x}})$ [59]. More details are given in the convergence analysis of the new algorithm in Section 5.5.

5.4 The mesh adaptive direct multisearch algorithm (DMulti-MADS) for multiobjective optimization

This section presents the new bound-constrained blackbox algorithm for multiobjective optimization, named DMulti-MADS. It is divided into three subsections. The first subsection gives a high-level description of DMulti-MADS. The two other subsections address specific points: the updating of the mesh size and frame size parameters of the list of non dominated points and the choice of the current incumbent point.

Algorithm 5 The mesh adaptive direct search algorithm (MADS)

Input : Choose a set of initial starting points $V^0 \subset \mathbb{R}^n$, $\Delta^0 > 0$ and $\mathbf{D} = \mathbf{GZ}$ be a positive spanning set matrix.

for $k = 0, 1, 2, \ldots$ **do**

Set $\delta^k = \min\left\{\Delta^k, \left(\Delta^k\right)^2\right\}$.

1. **Search step** (optional): Evaluate f at a finite set of points S^k on the mesh M^k. If successful, go to 3.

2. **Poll step** : Select a positive spanning set $\mathbb{D}_\Delta^k \subset \mathbf{D}$. Evaluate f at the set of poll points $P^k \subset F^k$ where F^k is the frame of extent Δ^k.

3. **Parameter update**: Update the cache V^{k+1}, the incumbent $\mathbf{x}^{k+1}$ and the frame size parameter Δ^k.

end for

Figure 5.3 A simplified version of the MADS algorithm.

5.4.1 The DMulti-MADS algorithm

DMulti-MADS deals with constraints via the extreme barrier approach [74]. Specifically, the objective function f is extended to an extreme barrier function by setting

$$f_\Omega(\mathbf{x}) = \begin{cases} f(x) & \text{if } \mathbf{x} \in \Omega, \\ (+\infty, +\infty, \ldots, +\infty)^\top & \text{otherwise.} \end{cases}$$

Concretely, all the points that do not satisfy constraints are affected an infinite objective value.

Similarly to the DMS [74] and DFMO [180] algorithms, DMulti-MADS generates a Pareto set approximation at each iteration. More specifically, at each iteration k, DMulti-MADS keeps a finite set L^k which stores all feasible non dominated points found until iteration k, called an *iterate list*. For each k, L^k is a finite set defined as

$$L^k = \left\{(\mathbf{x}^j, \Delta^j) : \mathbf{x}^j \in \Omega \text{ and } \Delta^j > 0, j = 1, 2, \ldots, l^k\right\}$$

where $l^k = |L^k|$ and Δ^j is the frame size parameter associated to the j-th non dominated element $\mathbf{x}^j$ of the list L^k. $\delta^j = \min\left\{\Delta^j, (\Delta^j)^2\right\}$ is the mesh size parameter associated with $\mathbf{x}^j$.

The DMulti-MADS algorithm is an extension of the direct search method MADS. As in single-objective optimization, its functioning is organized around a poll step and a search step, this

last one being optional as the convergence analysis does not depend on it. All notations P^k, M^k, F^k have the same mathematical meaning as in single-objective optimization (see Section 5.3). The algorithm is described in Algorithm 6.

Algorithm 6 DMulti-MADS algorithm with extreme barrier

Input : Choose $\mathbf{x}^0 \in \Omega$, $\Delta^0 > 0$, $\mathbf{D} = \mathbf{GZ}$ be a positive spanning set matrix, $\tau \in (0;1) \cap \mathbb{Q}$ the frame size adjustment parameter and $w^+ \in \mathbb{N}$ a fixed integer parameter. Initialize the list of non dominated points $L^0 = \{(\mathbf{x}^0, \Delta^0)\}$.

for $k = 0, 1, 2, \ldots$ **do**

Selection of the current incumbent point: Select $\left(\mathbf{x}^k, \Delta^k\right)$ element of L^k such that $(\mathbf{x}^k, \Delta^k) :=$ `selectCurrentIncumbent`(L^k, w^+, τ) (see Algorithm 8).

Set $\delta^k = \min\left\{\Delta^k, \left(\Delta^k\right)^2\right\}$. Initialize $L^{add} := \emptyset$.

Search step (optional): Evaluate f_Ω at a finite set of points S^k on the mesh $M^k = \left\{\mathbf{x}^k + \delta^k \mathbf{D}\mathbf{z} : \mathbf{z} \in \mathbb{N}^p\right\}$. Set $L^{add} := \{(\mathbf{x}, \Delta^k) : \mathbf{x} \in S^k\}$.

If $\mathbf{t} \prec \mathbf{x}^k$ for some $\mathbf{t} \in S^k$, declare the iteration as successful and skip the poll step.

Poll step : Select a positive spanning set $\mathbb{D}_\Delta^k \subset \mathbf{D}$. Evaluate f_Ω at the set of poll points $P^k = \{\mathbf{x}^k + \delta^k \mathbf{d} : \mathbf{d} \in \mathbb{D}_\Delta^k\}$ subset of the frame F^k of extent Δ^k. Set $L^{add} := \{(\mathbf{x}, \Delta^k) : \mathbf{x} \in P^k\} \cup L^{add}$.

If $\mathbf{t} \prec \mathbf{x}^k$ for some $\mathbf{t} \in P^k$, declare the iteration as successful. Otherwise declare the iteration as unsuccessful.

Parameter update: Remove all dominated points of L^{add}. Call the procedure $L^{k+1} :=$ `updateList`(L^k, L^{add}, τ) (see Algorithm 7).

If the iteration is unsuccessful, replace the poll center $(\mathbf{x}^k, \Delta^k)$ by $(\mathbf{x}^k, \Delta^{k+1})$ with $\Delta^{k+1} < \Delta^k$, i.e., $\Delta^{k+1} := \tau\Delta^k$.

end for

Figure 5.4 Description of the DMulti-MADS algorithm with extreme barrier.

At the beginning of iteration k, an element $\left(\mathbf{x}^k, \Delta^k\right)$ of the list L^k is selected as the current incumbent point at iteration k. The choice of the current incumbent point is crucial in the convergence analysis and will be detailed later on. A temporary list of points L^{add} is initialized to keep track of all the new generated points during iteration k with the associated frame size parameter Δ^k.

As for the MADS algorithm for single-objective optimization, the search step is optional: it aims at improving the performance of the algorithm by evaluating points on the mesh of coarseness δ^k. The poll step obeys to the same rules as in single-objective optimization. To guarantee convergence, evaluated points during the poll step must belong to the poll set P^k.

An iteration is said to be successful as soon as a new point dominating the current incumbent

x^k is found. Otherwise, it is said to be unsuccessful. As in single-objective optimization, one can choose the opportunistic or the complete polling strategy.

The two next subsections address main details left open during the description of DMulti-MADS.

5.4.2 Updating the list L^k

At the end of iteration k, L^k is updated as described in Algorithm 7.

Algorithm 7 `updateList`(L^k, L^{add}, τ)

1: **for** $j = 0, 1, 2, \ldots, |L^{add}|$ **do**
2: **if** there exists at least an element $(\mathbf{x}, \Delta) \in L^k$ such that $\mathbf{x}^j \prec \mathbf{x}$ **then**
3: set $L^k := L^k \setminus \left\{(\mathbf{x}, \Delta) \in L^k : \mathbf{x}^j \prec \mathbf{x}\right\} \cup (\mathbf{x}^j, \tau^{-1}\Delta^j)$.
4: **else if** there exists $i = 1, 2, \ldots, m$ such that $f_i(\mathbf{x}^j) < \min_{\mathbf{x} \in L^k} f_i(\mathbf{x})$ **then**
5: set $L^k := L^k \cup (\mathbf{x}^j, \tau^{-1}\Delta^j)$.
6: **else if** $\mathbf{x}^j \sim \mathbf{x}$ for all $(\mathbf{x}, \Delta) \in L^k$ **then**
7: set $L^k := L^k \cup (\mathbf{x}^j, \Delta^j)$.
8: **end if**
9: **end for**
10: **return** L^k.

Figure 5.5 Procedure to update the iterate list L^k.

The `updateList` procedure successively adds new points found during iteration k to the current list L^k and remove dominated points from L^k. At the end of the procedure, the updated L^{k+1} list contains only non dominated points. Let emphasize that before calling the `updateList` procedure, L^{add} has been filtered to remove dominated points; in Algorithm 7, L^{add} contains only non dominated points relatively to each other. By construction, all elements of L^{add} have the same associated frame size parameter value Δ^k.

At iteration k, the DMulti-MADS algorithm attempts to find at least a new point dominating the current incumbent $\mathbf{x}^k$. If found, the iteration is marked as successful. In this case, L^{add} contains at least a point dominating the current incumbent $\mathbf{x}^k$. The first condition of Lines 2−3 of Algorithm 7 ensures the replacing of element $(\mathbf{x}^k, \Delta^k)$ by the new element $(\mathbf{x}^j, \Delta^j)$ with $\mathbf{x}^j \prec \mathbf{x}^k$ and $\Delta^j > \Delta^k$. In case of an unsuccessful iteration, no element of L^{add} dominates the current incumbent $\mathbf{x}^k$. Consequently, the element $(\mathbf{x}^k, \Delta^k) \in L^k$ is substituted by $(\mathbf{x}^k, \Delta^{k+1})$ with $\Delta^{k+1} < \Delta^k$ as described in Algorithm 6.

During the search and poll steps, the algorithm can generate points which improve the Pareto set approximation L^k without dominating the current poll center $\mathbf{x}^k$. Typically, a good Pareto set approximation should verify three criteria [11, 279]:

- Its representation in the objective space should be as close as possible to the Pareto front.
- A good (uniform) distribution of the non dominated points in the objective space should be assessed.
- The extent of its representation in the objective space should be maximized, i.e., single-objective non dominated solutions should be part of the Pareto set approximation.

By increasing the mesh and frame size parameters for new promising elements by a factor τ^{-1}, $\tau \in (0,1) \cap \mathbb{Q}$, DMulti-MADS enables a larger exploration in the zone around these new points if they are selected as poll centers in the following iterations (Lines 3 and 5 of Algorithm 7). The `updateList` procedure considers the following points as promising:

- The ones that dominate a portion of the actual list of non dominated points L^k (Line 2 of Algorithm 7). The images of these points are closer to the Pareto front.
- The ones which improve the extent of the Pareto set approximation in the objective space (Line 4 of Algorithm 7), i.e., that reach a better value for at least one of the objectives.

On the contrary, the `updateList` procedure does not consider as promising new points that fill the approximated Pareto front, i.e. new non dominated points that neither dominate the current points or extend the approximated Pareto front (Line 6 of Algorithm 7). For these indifferent points, the frame size parameter value is kept as Δ^j (Line 7 of Algorithm 7). One may hope to find new non dominated points that locally improve the density of the Pareto front approximation around these new points. Figure 5.6 illustrates these concepts.

Let emphasize that the convergence analysis requires that new elements added to the list L^k must have a frame size parameter $\Delta \geq \Delta^k$. The `updateList` procedure satisfies these requirements.

5.4.3 Choice of the current incumbent $\mathbf{x}^k$

Contrary to the DMS algorithm [74], the choice of the incumbent point x^k at iteration k is less flexible, since the convergence analysis depends on it. More precisely, at iteration k,

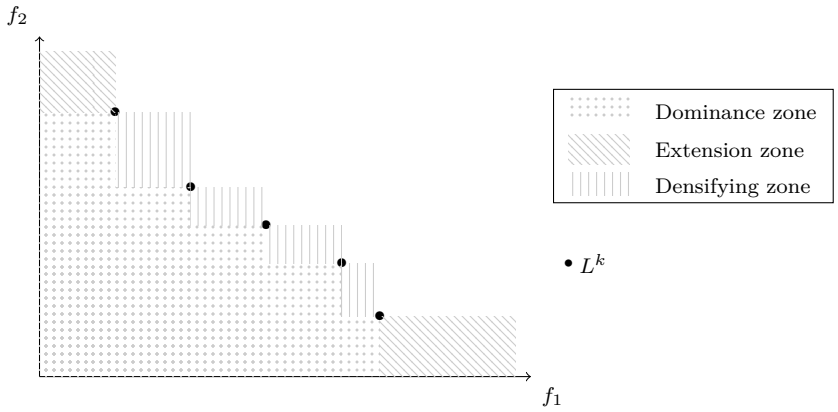


Figure 5.6 Zone of interests relatively to a set L^k for a biobjective minimization problem in the objective space.

element $(\mathbf{x}^k,\ \Delta^k)$ of the list L^k must verify

$$(\mathbf{x}^k, \Delta^k) \in \left\{(\mathbf{x}, \Delta) \in L^k : \tau^{w^+}\Delta^k_{\max} \leq \Delta \leq \Delta^k_{\max}\right\}$$

where $\tau \in (0,1) \cap \mathbb{Q}$ is the frame size adjustement parameter, $w^+ \in \mathbb{N}$ a fixed integer parameter (chosen by the user) and $\Delta^k_{\max}$ the maximum frame size parameter at iteration k defined by

$$\Delta^k_{\max} = \max_{j=1,2,\ldots,|L^k|} \Delta^j.$$

Having $w^+ = 0$ means that the current incumbent x^k at iteration k is chosen among the ones which have maximum frame size parameters. When w^+ is set to a sufficiently large value, the selection criterion is similar to the one of the DMS algorithm [74]: all elements of the list L^k at iteration k are potential current incumbents.

As new evaluated points at iteration k are initialized with the Δ^k value, it is possible to have several elements of L^h satisfying the above condition for $h \geq k$. One can ask how to choose the current incumbent point x^k at iteration k among the ones which satisfy the frame size parameter selection criterion.

Following the recommendations of [74], a first approach should be to take the first element of the list L^k which satisfies the frame size parameter selection condition as the current incumbent and add all new non dominated points at the end of the list to diversify the search. To fill gaps into the Pareto front approximation, a second approach consists in choosing elements satisfying the frame size parameter selection criterion in the least-dense zone of the Pareto front approximation.

Audet et al. [27] consider the distance between three consecutive points in biobjective optimization; the point in the middle is taken as the current incumbent. The crowding distance [85] extends this result to more objectives. For each objective, values of the non dominated points are sorted in ascendant order. The crowding distance for a given point is the sum of the normalized distance between this point and its two adjacent neighbors according to each objective. Its computational cost is in $\mathcal{O}(m \times |L^k| \times \log(|L^k|))$. But the crowding distance does not consider the extreme points of the current approximated Pareto front. Based on these remarks and the work of [27], a new way to select a current incumbent point $\mathbf{x}^k$ with frame size parameter Δ^k is proposed, as described in Algorithm 8.

Algorithm 8 `selectCurrentIncumbent`(L^k, w^+, τ)

Let $L^{select} := \left\{(\mathbf{x}, \Delta) \in L^k : \tau^{w^+}\Delta^k_{\max} \leq \Delta \leq \Delta^k_{\max}\right\}$ with $\Delta^k_{\max} = \max\limits_{j=1,2,\ldots,|L^k|} \Delta^j$.

if $|L^{select}| = 1$ **then**

  return $(\mathbf{x}, \Delta)$ with $L^{select} = \{(\mathbf{x}, \Delta)\}$.

else if $|L^{select}| = 2$ and $|L^k| = 2$ **then**

  Let $j_0 \in \arg\max\limits_{j=1,2} \max\limits_{i=1,2,\ldots,m} f_i(\mathbf{x}^j)$.

  return $(\mathbf{x}^{j_0}, \Delta^{j_0})$.

else

  Let $j_0 \in \arg\max\limits_{j=1,2,\ldots,|L^{select}|} \max\limits_{i=1,2,\ldots,m} \gamma_i(\mathbf{x}^j)$.

  return $(\mathbf{x}^{j_0}, \Delta^{j_0})$.

end if

Figure 5.7 A procedure to select the current incumbent at iteration k taking into account the spacing between elements of the iterate list L^k in the objective space.

The `selectCurrentIncumbent` procedure firstly stores elements of the list L^k satisfying the frame size parameter selection criterion into a temporary list L^{select}. If L^k possesses two elements, the procedure selects as current incumbent the one with the higher objective value among all objectives. By exploring the region around this incumbent, one can expect to find non dominated points with lowest objective values and then close to the Pareto front. If $|L^{select}| \geq 2$ and L^k has more than two elements, the element of the list L^{select} in the least dense zone of the current Pareto set approximation L^k in the objective space is selected, according to our new distance-based indicator γ_i for $i = 1, 2, \ldots, m$.

For each objective $i = 1, 2, \ldots, m$, $L^k = \{(\mathbf{x}^1, \Delta^1), (\mathbf{x}^2, \Delta^2), \ldots, (\mathbf{x}^{|L^k|}, \Delta^{|L^k|})\}$ is ordered such that

$$f_i(\mathbf{x}^1) \leq f_i(\mathbf{x}^2) \leq \ldots \leq f_i(\mathbf{x}^{|L^k|}).$$

γ_i corresponds to the scaled distance between three consecutive points according to objective i for $i = 1, 2, \ldots, m$. It is then defined, for $j = 1, 2, \ldots, |L^k|$, by

$$\gamma_i(\mathbf{x}^j) = \begin{cases} 2\,\dfrac{f_i(\mathbf{x}^2) - f_i(\mathbf{x}^1)}{f_i(\mathbf{x}^{|L^k|}) - f_i(\mathbf{x}^1)} & \text{if } j = 1, \\ 2\,\dfrac{f_i(\mathbf{x}^{|L^k|}) - f_i(\mathbf{x}^{|L^k|-1})}{f_i(\mathbf{x}^{|L^k|}) - f_i(\mathbf{x}^1)} & \text{if } j = |L^k|, \\ \dfrac{f_i(\mathbf{x}^{j+1}) - f_i(\mathbf{x}^{j-1})}{f_i(\mathbf{x}^{|L^k|}) - f_i(\mathbf{x}^1)} & \text{otherwise.} \end{cases}$$

If $\mathbf{x}^j$ is the first or last element of the sorted list L^k, the double scaled distance between this point and its closest neighbor for objective i is considered.

The point which is chosen as the current incumbent at iteration k is the one which satisfies the frame size parameter selection criterion in the least dense zone according to

$$\gamma = \max_{j=1,2,\ldots,|L^{select}|} \max_{i=1,2,\ldots,m} \gamma_i(\mathbf{x}^j).$$

Figure 5.8 illustrates this distance-based indicator for two objectives.

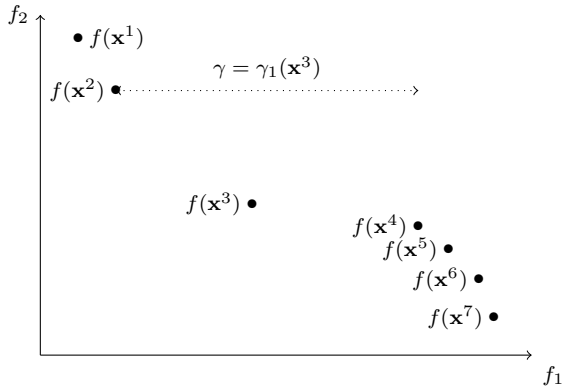


Figure 5.8 An example of the γ distance-based indicator in biobjective optimization. γ corresponds to the largest scaled distance between three consecutive points according to one objective i, for $i = 1, 2, \ldots, m$. Here, $\gamma = \gamma_1(\mathbf{x}^3)$.

Remark. In the implementation of direct search methods, the DMulti-MADS algorithm stops as soon as the mesh size parameter of the current poll center is lower than a threshold value chosen by the user or after a maximal budget of evaluations is reached. In the case where the considered budget of evaluations is important, to avoid premature stopping, the `selectCurrentIncumbent` procedure does not store in L^{select} the elements of L^k whose associated mesh size parameters are lower than the threshold value. If L^{select} is empty, an

element with maximal frame size parameter is returned and the algorithm will stop at the next iteration.

5.5 Convergence analysis of the DMulti-MADS algorithm

The previous section describes the DMulti-MADS algorithm. This section is devoted to its convergence analysis, inspired by the works of [74, 180]. Basically, the following result is shown: under mild assumptions, DMulti-MADS produces (at the limit) a set of local stationary points of the constrained multiobjective problem. As in the classical analysis of MADS [15], the objective is to show that there exists a subsequence of mesh size and frame size parameters converging to zero. However, contrary to [74], the analysis distinguishes between the Pareto set approximation and its elements. To guarantee this condition, the following assumption is required given a feasible starting point $\mathbf{x}^0$.

Assumption 5.5.1. *The level set* $L(\mathbf{x}^0) = \bigcup_{i=1}^{m} \{\mathbf{x} \in \Omega : f_i(\mathbf{x}) \leq f_i(\mathbf{x}^0)\}$ *is compact. Each component* f_i *of the objective function* f *is bounded from above and below for* $i = 1, 2, \ldots, m$.

Although this algorithm deals with bound-constrained multiobjective optimization problems, the convergence analysis can be generalized to any general feasible set $\Omega \subseteq \mathbb{R}^n$. In this section, all the results are proved with this generalization.

5.5.1 Preliminaries

This subsection is dedicated to the analysis of the convergence of the mesh and frame size parameters, based on the works of [15, 74, 244].

Theorem 7. *Let Assumption 5.5.1 hold. Then DMulti-MADS generates a sequence of Pareto set approximations iterates satisfying*

$$\liminf_{k\to\infty} \delta^k_{\max} = 0 \text{ and } \liminf_{k\to\infty} \Delta^k_{\max} = 0.$$

Proof. Suppose by contradiction that there exists a strictly positive lower bound on the mesh size parameter $\delta^k_{\max}$ for all $k \geq 0$. As in single-objective optimization [13, 15], similar arguments enable to show that all the points evaluated by DMulti-MADS lie on an integer lattice. Since the intersection of an integer lattice and a compact set, as assumed by Assumption 5.5.1, is a finite set, then only a finite number of points can be added to the Pareto set approximation. There remains to prove that the DMulti-MADS algorithm cannot cycle among these points.

When an element of the Pareto set approximation is removed, it is because it is dominated by a new point. By transitivity, it cannot be added to the Pareto set approximation again. At each successful iteration, at least one non-dominated point is added to the current Pareto set approximation. At each unsuccessful iteration, one non-dominated point may be added to the current Pareto set approximation[2]. As the number of points that can be added is finite, the number of times the maximal mesh size parameter of the approximated Pareto set can be increased or kept constant is finite. Given the update rule of the mesh size parameter, this contradicts the existence of a strictly positive lower bound on the mesh size parameter. The second part of the theorem is based on the fact that $\delta^k_{\max} = \min\left\{\Delta^k_{\max}, \left(\Delta^k_{\max}\right)^2\right\}$, which concludes the proof. □

Let highlight that when the iteration k is marked as unsuccessful, the maximal mesh size parameter of the current Pareto set approximation is not always decreased. Indeed, new non dominated points can be added to the Pareto set approximation even though the algorithm does not find new points dominating the current incumbent x^k. Furthermore, the Pareto set approximation can contain elements with the same mesh size parameter.

The following corollary puts into relief a statistical relation between the elements of the list L^k. For a given subsequence of Pareto set approximations, the mesh and frame size parameters converge in average towards zero.

Corollary 7.1. *Let Assumption 5.5.1 hold. Let* $\{L^k\}_{k\in\mathbb{N}}$ *with*

$$L^k = \left\{(\mathbf{x}^{j_k}, \Delta^{j_k}) : \mathbf{x}^{j_k} \in \Omega \text{ and } \Delta^{j_k} > 0,\ j_k = 1, 2, \ldots, |L^k|\right\}$$

be the sequence of current Pareto set approximations generated by the DMulti-MADS algorithm. Then

$$\lim_{k\to\infty} \inf \bar{\delta}^k = 0 \text{ and } \lim_{k\to\infty} \inf \bar{\Delta}^k = 0$$

with

$$\bar{\delta}^k = \frac{1}{|L^k|}\sum_{j_k=1}^{|L^k|} \delta^{j_k} \text{ and } \bar{\Delta}^k = \frac{1}{|L^k|}\sum_{j_k=1}^{|L^k|} \Delta^{j_k}.$$

Proof. Since $\delta^k_{\max} = \max_{j_k=1,2,\ldots,|L^k|} \delta^{j_k}$ for $k = 0, 1, 2, \ldots$, one has $\bar{\delta}^k \leq \delta^k_{\max}$ for all $k \in \mathbb{N}$. Theorem 7 states that there exists a subset of indexes $k \in K$ such that $\{\delta^k_{\max}\}_{k\in K}$ converges to zero. By the squeeze theorem, one gets $\lim_{k\in K} \bar{\delta}^k = \lim_{k\in K} \delta^k_{\max} = 0$. The proof with the frame size parameter is equivalent. □

[2]We replace "at most" by "may be" in this sentence.

Intuitively, this corollary claims that either the cardinality of L^k converges to infinity or all the mesh and frame size parameters converge to zero for a given set of indexes $k \in K$.

Theorem 7 analyzes the convergence of mesh size parameters relatively to the list of non dominated points L^k. One has to go deeper to analyze the behavior of mesh size and frame size parameters of specific elements of the list to prove convergence of the DMulti-MADS algorithm to stationary points. To do that, the concept of linked sequence, taken from [180] is introduced.

Definition 38. Let $\{L^k\}_{k\in\mathbb{N}}$ with $L^k = \left\{(\mathbf{x}^j, \Delta^j) : \mathbf{x}^j \in \Omega \text{ and } \Delta^j > 0, j = 1, 2, \ldots, |L^k|\right\}$ be the sequence of current Pareto set approximations generated by the DMulti-MADS algorithm. A linked sequence is defined as a sequence $\{(\mathbf{x}^{j_k}, \Delta^{j_k})\}$ such that for any $k = 1, 2, \ldots$, the pair $(\mathbf{x}^{j_k}, \Delta^{j_k}) \in L^k$ is generated at iteration $k-1$ of DMulti-MADS from the pair $(\mathbf{x}^{j_{k-1}}, \Delta^{j_{k-1}}) \in L^{k-1}$.

The relation between the pair $(\mathbf{x}^{j_k}, \Delta^{j_k}) \in L^k$ and $(\mathbf{x}^{j_{k-1}}, \Delta^{j_{k-1}}) \in L^{k-1}$ is precised below.

1. Successful iteration : the algorithm generates at least one point that dominates the current incumbent point $\mathbf{x}^{k-1}$. All non dominated points at iteration $k-1$ which are not dominated at the end of iteration $k-1$ are inserted in the Pareto set approximation L^k.

   Then one has:

   - $\forall (\mathbf{x}^{j_k}, \Delta^{j_k}) \in L^k \setminus L^{k-1}$,

     $$\mathbf{x}^{j_k} = \mathbf{x}^{k-1} + \delta^{k-1}\mathbf{D}\mathbf{z}^{k-1} \text{ for some } \mathbf{z}^{k-1} \in \mathbb{N}^l \text{ and } \Delta^{jk} \in \{\Delta^{k-1}, \tau^{-1}\Delta^{k-1}\}.$$

   - $\forall (\mathbf{x}^{j_k}, \Delta^{j_k}) \in L^k \cap L^{k-1}$,

     $$\mathbf{x}^{j_k} = \mathbf{x}^{j_{k-1}} \text{ and } \Delta^{j_k} = \Delta^{j_{k-1}}.$$

2. Unsuccessful iteration : The algorithm does not generate a point that dominates the current incumbent point $\mathbf{x}^{k-1}$. However, it is possible that the algorithm finds new non dominated points which are inserted into the Pareto set approximation.

   Then one has:

   - $\forall (\mathbf{x}^{j_k}, \Delta^{j_k}) \in L^k \setminus L^{k-1}$,

     $$\mathbf{x}^{j_k} = \mathbf{x}^{k-1} + \delta^{k-1}\mathbf{D}\mathbf{z}^{k-1} \text{ for some } \mathbf{z}^{k-1} \in \mathbb{N}^l \text{ and } \Delta^{jk} \in \{\Delta^{k-1}, \tau^{-1}\Delta^{k-1}\}.$$

- $\forall(\mathbf{x}^{j_k}, \Delta^{j_k}) \in L^k \cap L^{k-1} \setminus \{(\mathbf{x}^{k-1}, \Delta^{k-1})\}$,

$$\mathbf{x}^{j_k} = \mathbf{x}^{j_{k-1}} \text{ and } \Delta^{j_k} = \Delta^{j_{k-1}}.$$

- $\forall(\mathbf{x}^{j_k}, \Delta^{j_k}) \in \{(\mathbf{x}^{k-1}, \Delta^{k-1})\}$,

$$\mathbf{x}^{j_k} = \mathbf{x}^{j_{k-1}} \text{ and } \Delta^{j_k} = \tau\Delta^{j_{k-1}}.$$

  Note that the current incumbent at iteration $k-1$ is not always the same at iteration k.

As proved below, linked sequences generate subsequences of points whose mesh size and frame size parameters converge to 0.

Theorem 8. *Let Assumption 5.5.1 hold. Let* $\{L^k\}_{k\in\mathbb{N}}$ *with*

$$L^k = \left\{(\mathbf{x}^j, \Delta^j) : \mathbf{x}^j \in \Omega \text{ and } \Delta^j > 0,\ j = 1, 2, \ldots, |L^k|\right\}$$

be the sequence of current approximated Pareto sets generated by the DMulti-MADS algorithm. Then every linked sequence $\{(\mathbf{x}^{j_k}, \Delta^{j_k})\}$ *is such that*

$$\liminf_{k\to\infty} \delta^{j_k} = 0 \text{ and } \liminf_{k\to\infty} \Delta^{j_k} = 0.$$

Proof. $\forall k \in \mathbb{N}, 0 \leq \delta^{j_k} \leq \delta^k_{\max}$. Using Theorem 7 and the squeeze theorem, one gets

$$\liminf_{k\to\infty} \delta^{j_k} = \liminf_{k\to\infty} \delta^k_{\max} = 0.$$

As $\delta^{j_k} = \min\left\{\Delta^{j_k}, (\Delta^{j_k})^2\right\}$, it results that

$$\liminf_{k\to\infty} \Delta^{j_k} = 0.$$

□

Note that due to the update strategy, the mesh and frame size parameters for a linked sequence $\{(x^{j_k}, \Delta^{j_k})\}$ can only decrease when there exists an index $k \in \mathbb{N}$ such that $x^k = x^{j_k}$ and iteration k is marked as unsuccessful.

5.5.2 Refining subsequences and directions

The theory of classical direct search methods consists in analyzing the behavior at limit points of unsuccessful iterates. The concept of refining subsequences in the context of multiobjective optimization, previously introduced in [15], is adapted.

Definition 39. A subsequence $\{\mathbf{x}^k\}_{k\in K}$ of iterates corresponding to unsuccessful poll steps is said to be a refining subsequence if $\{\delta^k\}_{k\in K}$ converges to 0. The limit point $\hat{x}$ of a convergent refining subsequence $\{\mathbf{x}^k\}_{k\in K}$ is said to be a refining point[3].

By Assumption 5.5.1 and Theorem 8, every linked sequence produced by the DMulti-MADS algorithm contains a refining subsequence.

Theorem 9. *Let Assumption 5.5.1 hold. Let* $\{L^k\}_{k\in\mathbb{N}}$ *with*

$$L^k = \left\{(\mathbf{x}^j, \Delta^j) : \mathbf{x}^j \in \Omega \text{ and } \Delta^j > 0,\ j = 1, 2, \ldots, |L^k|\right\}$$

be the sequence of current Pareto set approximations generated by the DMulti-MADS algorithm. Then every linked sequence $\{(\mathbf{x}^{j_k}, \Delta^{j_k})\}$ *is such that* $\{\mathbf{x}^{j_k}\}_{k\in K}$ *is a refining subsequence.*

This theorem is different than the one proposed by [74] where the DMS algorithm generates at least one refining subsequence [4].

As the DMulti-MADS convergence analysis is based on the study of generalized directional derivatives along certain limits directions at refined points, the concept of a refining direction [15] is introduced.

Definition 40. Given a refining subsequence $\{\mathbf{x}^{j_k}\}_{k\in K}$ and its corresponding refining point $\hat{x}$, a direction d is said to be a refining direction if and only if there exists an infinite subset $K' \subseteq K$ such that $\mathbf{d}^k \in \mathbb{D}_\Delta^k$ with $\mathbf{x}^{j_k} + \delta^{j_k}\mathbf{d}^k \in \Omega$ and $\lim_{k\in K'} \frac{\mathbf{d}^k}{\|\mathbf{d}^k\|} = \frac{\mathbf{d}}{\|\mathbf{d}\|}$.

5.5.3 Tangent cones and generalized derivatives

The main convergence result of DMulti-MADS is that a limit point of a refining subsequence of a linked sequence generated by the algorithm is Pareto-Clarke stationary. It requires some concepts linked to stationarity in the context of nonsmooth constrained multiobjective optimization.

[3] This sentence was modified according to the published article

[4] This sentence was modified according to the published article.

Classical theory of direct-search methods in the context of constrained single-objective optimization makes use of the hypertangent cone, which is a generalization of the tangent cone at $\mathbf{x}$, i.e. the set of directions that point inside Ω. Definition and notations are taken from [15].

Definition 41. A vector $\mathbf{d} \in \mathbb{R}^n$ is said to be a Clarke tangent vector to the set $\Omega \subseteq \mathbb{R}^n$ at the point $\mathbf{x}$ in the closure of Ω if for every sequence $\{\mathbf{y}^k\}$ of elements of Ω that converges to $\mathbf{x}$ and for every sequence of positive real numbers $\{t^k\}$ converging to zero, there exists a sequence of vectors $\{\mathbf{w}^k\}$ converging to $\mathbf{d}$ such that $\mathbf{y}^k + t^k\mathbf{w}^k \in \Omega$.

The set of all Clarke tangent vectors to Ω at $\mathbf{x}$ is called the Clarke tangent cone at $\mathbf{x}$ and is denoted by $\mathcal{T}_\Omega^{Cl}(\mathbf{x})$. The interior of this cone is defined as the hypertangent cone.

Definition 42. A vector $\mathbf{d} \in \mathbb{R}^n$ is said to be a hypertangent vector to the set $\Omega \subseteq \mathbb{R}^n$ at the point $\mathbf{x} \in \Omega$ if and only if there exists a scalar $\epsilon > 0$ such that

$$\mathbf{y} + t\mathbf{w} \in \Omega, \ \forall \mathbf{y} \in \Omega \cap \mathcal{B}_\epsilon(\mathbf{x}), \ \mathbf{w} \in \mathcal{B}_\epsilon(\mathbf{d}) \text{ and } 0 < t < \epsilon$$

where $\mathcal{B}_\epsilon(\mathbf{d})$ is the open ball centered at $\mathbf{x}$ of radius ϵ.

The set of all hypertangent directions vectors to Ω at $\mathbf{x}$ is called the hypertangent cone to Ω at $\mathbf{x}$, and is denoted by $\mathcal{T}_\Omega^{H}(\mathbf{x})$.

Note that the Clarke tangent cone can be considered as the closure of the hypertangent cone. The convergence analysis requires the assumption that the objective function f is locally Lipschitz continuous in Ω, i.e., each of its components f_i, for $i = 1, 2, \ldots, m$, is locally Lipschitz continuous in Ω. Assuming this assumption is satisfied, the Clarke-Jahn generalized derivatives [59] of each function f_i along directions $\mathbf{d}$ in the hypertangent cone to Ω at $\mathbf{x}$ exist and are defined by

$$f_i^o(\mathbf{x}; \mathbf{d}) = \limsup_{\substack{\mathbf{y} \to \mathbf{x}, \ \mathbf{y} \in \Omega \\ t \searrow 0, \ \mathbf{y} + t\mathbf{d} \in \Omega}} \frac{f_i(\mathbf{y} + t\mathbf{d}) - f_i(\mathbf{y})}{t}, \ i = 1, 2, \ldots, m.$$

Audet et al [15] show that the directions $\mathbf{v}$ in the Clarke tangent cone can be expressed by taking the limit, i.e.,

$$f_i^o(\mathbf{x}; \mathbf{v}) = \lim_{\substack{\mathbf{d} \in \mathcal{T}_\Omega^H(\mathbf{x}) \\ \mathbf{d} \to \mathbf{v}}} f_i^o(\mathbf{x}; \mathbf{d}), \ i = 1, 2, \ldots, m.$$

Stationarity conditions for the DMulti-MADS algorithm can now be defined.

Definition 43. Let f be Lispchitz continuous near a point $\hat{\mathbf{x}} \in \Omega$. $\hat{\mathbf{x}}$ is said to be a Pareto-Clarke critical point of f in Ω for all directions $\mathbf{d} \in \mathcal{T}_\Omega^{Cl}(\hat{\mathbf{x}})$ if there exists $i = i(\mathbf{d}) \in \{1, 2, \ldots, m\}$ such that $f_i^o(\hat{\mathbf{x}}; \mathbf{d}) \geq 0$.

If the objective function f is differentiable, this definition can be reformulated using the gradient of each component of the objective function f.

Definition 44. Let f be differentiable at a point $\hat{\mathbf{x}} \in \Omega$. $\hat{x}$ is said to be a Pareto-Clarke KKT critical point of f in Ω for all directions $\mathbf{d} \in \mathcal{T}_\Omega^{Cl}(\hat{x})$ if there exists $i = i(\mathbf{d}) \in \{1, 2, \ldots, m\}$ such that $\nabla f_i(\hat{\mathbf{x}})^\top \mathbf{d} \geq 0$.

5.5.4 Convergence results

The main convergence results of the DMulti-MADS algorithm can now be given. It states that every limit point of a refining subsequence generated by DMulti-MADS is Pareto-Clarke optimal under the condition that the set of refining directions is dense in the unit sphere. Thus, every limit point of a refining subsequence of a linked sequence generated by DMulti-MADS is Pareto-Clarke optimal. The proof follows the classical theory of direct-search methods and in particular of the DMS algorithm [74].

Theorem 10. *Let $\{\mathbf{x}^{j_k}\}_{k \in K}$ be a refining subsequence converging to $\hat{\mathbf{x}} \in \Omega$ and a refining direction $\mathbf{d} \in \mathcal{T}_\Omega^H(\hat{\mathbf{x}})$. Assume that f is Lipschitz continuous near $\hat{\mathbf{x}}$. Then there exists $i = i(\mathbf{d}) \in \{1, 2, \ldots, m\}$ such that $f_{i(d)}^o(\hat{\mathbf{x}}; \mathbf{d}) \geq 0$.*

Proof. Let $\{\mathbf{x}^{j_k}\}_{k \in K}$ be a refining subsequence converging to a refined point $\hat{\mathbf{x}} \in \Omega$ and $\mathbf{d} \in T_\Omega^H(\hat{\mathbf{x}})$ a refining direction for $\hat{\mathbf{x}}$. By definition of a refining direction, there exists an infinite subsequence K' of the set of indices K of unsuccessful iterations, with poll directions $\mathbf{d}^k \in \mathbb{D}_\Delta^k$ such that $\mathbf{x}^{j_k} + \delta^k \mathbf{d}^k \in \Omega$ and $\lim_{k \in K'} \frac{\mathbf{d}^k}{\|\mathbf{d}^k\|} = \frac{\mathbf{d}}{\|\mathbf{d}\|}$ for all $k \in K'$. Let ν be the Lipschitz constant of f near $\hat{\mathbf{x}}$.

For $i \in \{1, 2, \ldots, m\}$, one has

$$
\begin{aligned}
f_i^o\left(\hat{\mathbf{x}}; \frac{\mathbf{d}}{\|\mathbf{d}\|}\right) &= f_i^o\left(\hat{\mathbf{x}}; \frac{\mathbf{d}}{\|\mathbf{d}\|}\right) + \limsup_{k \in K'} \frac{\nu \delta^k \|\mathbf{d}^k\| \left\| \frac{\mathbf{d}^k}{\|\mathbf{d}^k\|} - \frac{\mathbf{d}}{\|\mathbf{d}\|} \right\|}{\delta^k \|\mathbf{d}^k\|} \\
&\geq f_i^o\left(\hat{\mathbf{x}}; \frac{\mathbf{d}}{\|\mathbf{d}\|}\right) + \limsup_{k \in K'} \frac{\left| f_i(\mathbf{x}^{j_k} + \delta^k \mathbf{d}^k) - f_i(\mathbf{x}^{j_k} + \delta^k \|\mathbf{d}^k\| \frac{\mathbf{d}}{\|\mathbf{d}\|}) \right|}{\delta^k \|\mathbf{d}^k\|} \\
&\geq \limsup_{k \in K'} \frac{f_i(\mathbf{x}^{j_k} + \delta^k \|\mathbf{d}^k\| \frac{\mathbf{d}}{\|\mathbf{d}\|}) - f_i(\mathbf{x}^{j_k})}{\delta^k \|\mathbf{d}^k\|} \\
&+ \limsup_{k \in K'} \frac{\left| f_i(\mathbf{x}^{j_k} + \delta^k \mathbf{d}^k) - f_i(\mathbf{x}^{j_k} + \delta^k \|\mathbf{d}^k\| \frac{\mathbf{d}}{\|\mathbf{d}\|}) \right|}{\delta^k \|\mathbf{d}^k\|} \\
&\geq \limsup_{k \in K'} \frac{f_i(\mathbf{x}^{j_k} + \delta^k \mathbf{d}^k) - f_i(\mathbf{x}^{j_k} + \delta^k \|\mathbf{d}^k\| \frac{\mathbf{d}}{\|\mathbf{d}\|}) + f_i(\mathbf{x}^{j_k} + \delta^k \|\mathbf{d}^k\| \frac{\mathbf{d}}{\|\mathbf{d}\|}) - f_i(\mathbf{x}^{j_k})}{\delta^k \|\mathbf{d}^k\|} \\
&= \limsup_{k \in K'} \frac{f_i(\mathbf{x}^{j_k} + \delta^k \mathbf{d}^k) - f_i(\mathbf{x}^{j_k})}{\delta^k \|\mathbf{d}^k\|}.
\end{aligned}
$$

As $\{\mathbf{x}^{j_k}\}_{k \in K}$ is a refining subsequence, each $k \in K' \subseteq K$ corresponds to an unsuccessful iteration, and $\mathbf{x}^{j_k} + \delta^k \mathbf{d}^k \in \Omega$ for $\mathbf{d}^k \in \mathbb{D}_\Delta^k$ does not dominate $\mathbf{x}^{j_k}$. One can find for $k \in K'$ a component of the objective function of index $i(k)$ such that $f_{i(k)}(\mathbf{x}^{j_k} + \delta^k \mathbf{d}^k) - f_{i(k)}(\mathbf{x}^k) \geq 0$. As the objective function has a finite number of components, one can consider a subset of iteration indexes $K'' \subset K'$ such that there exists at least one index $i = i(d)$ such that:

$$
f_i^o\left(\hat{\mathbf{x}}; \frac{\mathbf{d}}{\|\mathbf{d}\|}\right) \geq \limsup_{k \in K''} \frac{f_i(\mathbf{x}^{j_k} + \delta^k \mathbf{d}^k) - f_i(\mathbf{x}^{j_k})}{\delta^k \|\mathbf{d}^k\|} \geq 0.
$$

□

If f is strictly differentiable at a refining point, one can state the following corollary.

Corollary 10.1. *Let $\{\mathbf{x}^{j_k}\}_{k \in K}$ be a refining subsequence converging to $\hat{\mathbf{x}} \in \Omega$ and a refining direction $\mathbf{d} \in T_\Omega^H(\hat{\mathbf{x}})$. Assume that f is strictly differentiable at $\hat{\mathbf{x}}$. Then there exists an $i \in \{1, 2, \ldots, m\}$ such that $\nabla f_i(\hat{\mathbf{x}})^\top \mathbf{d} \geq 0$.*

Proof. It comes from the fact that when f is strictly differentiable at a point $\mathbf{x} \in \Omega$, $f_i^o(\mathbf{x}; \mathbf{d}) = \nabla f_i(\mathbf{x})^\top \mathbf{d}$ for $\mathbf{d} \in \mathbb{R}^n$, $i = 1, 2, \ldots, m$ (see [59]). □

Assuming that the set of refining directions is dense in the hypertangent cone at a refining point $\hat{\mathbf{x}}$[5], one can state the following theorem, which complies with the DMS algorithm convergence analysis.

[5]The following works [2, 15] describe various strategies to generate such sets of refining directions.

Theorem 11. *Let $\{\mathbf{x}^{j_k}\}_{k\in K}$ be a refining subsequence converging to $\hat{\mathbf{x}} \in \Omega$. Assume that f is Lipschitz continuous near $\hat{\mathbf{x}}$ and $\mathcal{T}_\Omega^H(\hat{\mathbf{x}}) \neq \emptyset$. If the set of refining directions for $\hat{\mathbf{x}}$ is dense in $\mathcal{T}_\Omega^{Cl}(\hat{\mathbf{x}})$, then $\hat{\mathbf{x}}$ is a Pareto-Clarke critical point. In addition, if f is strictly differentiable at $\hat{\mathbf{x}}$, then $\hat{\mathbf{x}}$ is a Pareto-Clarke KKT critical point.*

Proof. As proved in [15], given a direction v in the Clarke tangent cone, one has

$$f_i^o(\hat{\mathbf{x}}; \mathbf{v}) = \lim_{\substack{\mathbf{d}\in\mathcal{T}_\Omega^H(\hat{\mathbf{x}}) \\ \mathbf{d}\to\mathbf{v}}} f_i^o(\hat{\mathbf{x}}; \mathbf{d}), i = 1, 2 \ldots, m.$$

As the set of refining directions for $\hat{\mathbf{x}}$ is dense in $\mathcal{T}_\Omega^{Cl}(\hat{x})$, there exists a sequence of refining directions $\{\mathbf{d}^r\}_{r\in R} \in \mathcal{T}_\Omega^H(\hat{\mathbf{x}})$ for $\hat{\mathbf{x}}$ such that $\lim_{r\in R} \mathbf{d}^r = \mathbf{v}$. Since the number of components of the objective function is finite, considering a subset $R' \subseteq R$ of indexes, one gets $\mathbf{v} = \lim_{r\in R'} \mathbf{d}^r$ with $f_{i(v)}^o(\hat{\mathbf{x}}; \mathbf{d}^r) \geq 0$ for all $r \in R'$ by Theorem 10. Passing at the limit concludes the proof. The second statement of the theorem can be deduced easily. □

5.6 Computational experiments

This section is devoted to the numerical experiments of DMulti-MADS on bound-constrained multiobjective problems taken from [74]. The first part introduces the considered test problems and solvers. The second part presents an extension of the classical data profiles [192] for multiobjective blackbox optimization, based on the hypervolume indicator [279]. Several variants of DMulti-MADS are then compared using this tool in a third part. In the last part, the performance of DMulti-MADS is analysed versus other state-of-the-art solvers. The DMulti-MADS code is freely available at `https://github.com/bbopt/DMultiMadsEB`.

5.6.1 Bound-constrained problems and algorithms tested

All algorithms are tested on the benchmark set of multiobjective optimization problems taken from [74]. It is composed of 100 problems, with a number of variables $n \in [1, 30]$ and a number of objective functions $m \in \{2, 3, 4\}$. It has 69 problems with $m = 2$, 29 problems with $m = 3$, and 2 problems with $m = 4$. A modeling of these problems in `AMPL` can be found at `www.mat.uc.pt/dms/`. Their implementations coded in `Matlab` and `C++` can be found at `https://github.com/bbopt/DMultiMadsEB`. In the numerical experiments, the following solvers are tested:

- BiMADS (Bi-objective Mesh Adaptive Direct Search) [27] tested only for $m = 2$ objectives – `www.gerad.ca/nomad/`.

- DMS (Direct MultiSearch) [74], version `0.3.` – `www.mat.uc.pt/dms/`
- MOIF (MultiObjective Implicit Filtering) [126], version `0.1` – `www.iasi.cnr.it/~liuzzi/DFL/`.
- NSGA-II (Non Dominated Sorting Algorithm II) [85]; implemented in the `Pymoo` Library [42], version `0.3.2` – `pymoo.org/`.

BiMADS, DMS and MOIF are deterministic algorithms, whereas NSGA-II is a stochastic solver. All numerical results can be found at `https://github.com/bbopt/DMultiMadsEB`.

5.6.2 Data profiles for multiobjective blackbox optimization

In single-objective optimization, data profiles [192] enable the user to assess the performance of a method on a set of problems for a given budget of function evaluations. Assume one wants to solve $\min_{\mathbf{x}\in\Omega} f(\mathbf{x})$ where f is a single-objective function and Ω the set of constraints. Let $\mathcal{P}$ be the set of problems and $\mathcal{A}$ the set of considered algorithms. A data profile associated to a solver $a \in \mathcal{A}$ is a cumulative distribution function which returns the percentage of problems in $\mathcal{P}$ solved by $a \in \mathcal{A}$ for a given budget of group of function evaluations $k \in \mathbb{N}$, i.e.

$$d_a(k) = \frac{1}{|\mathcal{P}|} \left|\{p \in \mathcal{P} : N_{a,p} \leq k(n_p + 1)\}\right| \tag{5.1}$$

where $N_{a,p}$ is the number of functions evaluations required by solver $a \in \mathcal{A}$ to solve the problem $p \in \mathcal{P}$ and n_p the dimension of the problem $p \in \mathcal{P}$. By convention, if a problem has not been solved given a maximum budget of function evaluations, then $N_{a,p} = +\infty$. The $n_p + 1$ term on the right part of the inequality in Equation (5.1) is added based on the assumption that a problem with higher dimension requires more function evaluations to be solved than a problem with lower dimension. $n_p + 1$ is equally the number of points needed to construct a simplex gradient in $\mathbb{R}^{n_p}$.

The definition of a convergence test to claim a problem has been solved is a critical phase in the construction of a data profile. For single-objective optimization, let $\mathbf{x}^b$ be the best feasible point found by all algorithms on a given problem, and $\mathbf{x}^e$ be the best feasible point found by a given algorithm on this problem after e evaluations. Then the problem is said to be solved by this algorithm with accuracy $\varepsilon_\tau > 0$ if

$$f(\mathbf{x}^0) - f(\mathbf{x}^e) \geq (1 - \varepsilon_\tau)(f(\mathbf{x}^0) - f(\mathbf{x}^b)),$$

where $\mathbf{x}^0$ is the feasible initial starting point.

Several works describe the construction of data profiles for multiobjective blackbox optimization, based on the use of quality indicators (see [11, 175] for surveys on quality indicators). Since the works of [74], which to the best of our knowledge, introduced data and performance profiles for multiobjective blackbox optimization, many researchers have adopted this framework to assess the performance of their methods [126, 180, 213]. However, these frameworks rely on spread and cardinality metrics, which are not Pareto compliant [11] with the dominance order for multiobjective optimization. The use of Pareto compliant quality indicators such as the hypervolume indicator is addressed in [180] for the construction of performance profiles [192] for multiobjective blackbox optimization. Nonetheless, performance profiles possess the following drawbacks: they are sensitive to the number of considered solvers [74, 130] and are more difficult to interpret than the data profiles [50].

In this work, a new extension of data profiles for multiobjective optimization is proposed, which relies on the hypervolume indicator. Note that the use of the hypervolume indicator in data profiles is not new, as it is done in [50]. Some similarities between this work and [50] are present. The main differences are highlighted:

- This work is more detailed in the description of the integration of the hypervolume indicator into the convergence criterion, specifically the scaling of the objective vectors and the positioning of the reference point.
- Variability of stochastic algorithms is included in this work based on the research of [228, 229].

The hypervolume indicator [279] represents the volume of the space in the objective space dominated by a Pareto front approximation Y_N and delimited above by an objective vector $\mathbf{r} \in \mathbb{R}^m$ such that $\forall \mathbf{y} \in Y_N, \mathbf{y} < \mathbf{r}$. An illustration of the hypervolume indicator is shown in Figure 5.9.

The hypervolume indicator enjoys many properties: it is Pareto compliant with the dominance ordering, its interpretation is simple and it can serve as a metric for convergence, cardinality, spread and extension of a Pareto front approximation [11, 175]. On the contrary, its computation is exponential in the number of objectives [11, 175]. Practically, there exist several libraries (see for example [121]) which can compute the hypervolume indicator value in less than some milliseconds on modern machines for a small number of objectives ($m \in \{1, 2, 3, 4\}$). Indeed, one has to keep in mind that the construction time of the data profiles is not important in these experiments.

To build the convergence test for the multiobjective optimization case, one needs to consider a Pareto front reference Y^p for the problem $p \in \mathcal{P}$. From this Pareto front reference, let

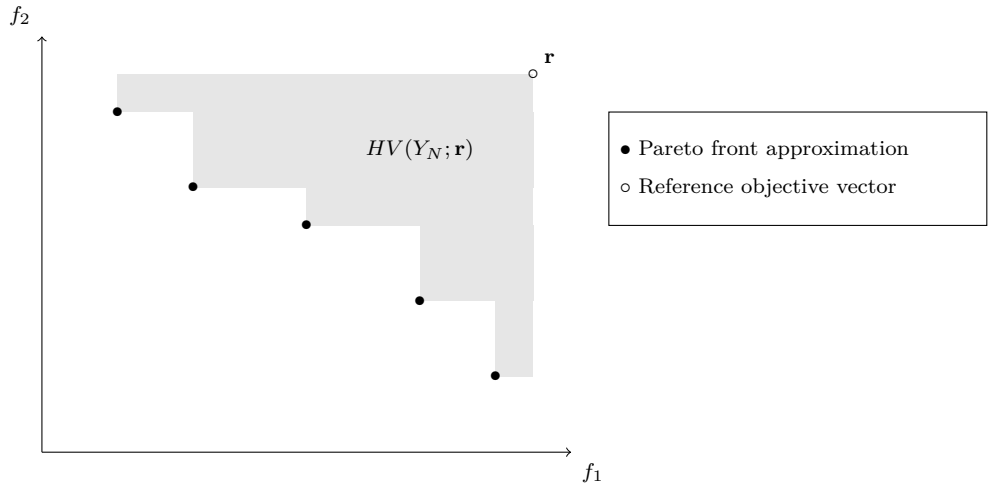


Figure 5.9 Illustration of the hypervolume indicator (HV) for a biobjective minimization problem, delimited above by the reference objective vector $r \in \mathbb{R}^2$. The higher, the better.

extract the approximated ideal objective vector

$$\tilde{\mathbf{y}}^{I,p} = \left(\min_{\mathbf{y} \in Y^p} \mathbf{y}_1, \min_{\mathbf{y} \in Y^p} \mathbf{y}_2, \ldots, \min_{\mathbf{y} \in Y^p} \mathbf{y}_{m_p} \right),$$

and the approximated Nadir objective vector

$$\tilde{\mathbf{y}}^{N,p} = \left(\max_{\mathbf{y} \in Y^p} \mathbf{y}_1, \max_{\mathbf{y} \in Y^p} \mathbf{y}_2, \ldots, \max_{\mathbf{y} \in Y^p} \mathbf{y}_{m_p} \right),$$

where m_p is the number of objectives considered in problem $p \in \mathcal{P}$.

Let Y^e be the Pareto front approximation found after e evaluations by a given deterministic algorithm. To avoid privileging an objective function against another, a transformation T is applied to the Pareto front reference, the Pareto front approximation and the approximated Nadir objective vector. This transformation is defined by: $\forall \mathbf{y} \in Y^e \cup Y^p \cup \{\tilde{\mathbf{y}}^{N,p}\}$

$$T(y) = \begin{cases} (\mathbf{y} - \tilde{\mathbf{y}}^{I,p}) \oslash (\tilde{\mathbf{y}}^{N,p} - \tilde{\mathbf{y}}^{I,p}) & \text{if } \tilde{\mathbf{y}}^{I,p} \neq \tilde{\mathbf{y}}^{N,p} \\ \mathbf{y} - \tilde{\mathbf{y}}^{I,p} & \text{otherwise} \end{cases}$$

where $\oslash$ is the element-wise divisor operator. Thus all objective vectors are scaled and translated such that $T(\tilde{\mathbf{y}}^{I,p}) = 0_{\mathbb{R}^{m_p}}$ and $T(\tilde{\mathbf{y}}^{N,p}) = 1_{\mathbb{R}^{m_p}}$ if $\tilde{\mathbf{y}}^{N,p}$ exists. Note that this translation does not modify the dominance ordering, i.e. $\mathbf{y}^1 \leq \mathbf{y}^2$ implies $T(\mathbf{y}^1) \leq T(\mathbf{y}^2)$ with $\mathbf{y}^1$, $\mathbf{y}^2$ two objective vectors. Finally, the multiobjective problem $p \in \mathcal{P}$ is said to be

solved by this algorithm with accuracy $\varepsilon_\tau > 0$ if

$$\frac{HV\left(T(Y^e); T(\tilde{\mathbf{y}}^{N,p})\right)}{HV\left(T(Y^p); T(\tilde{\mathbf{y}}^{N,p})\right)} \geq 1 - \varepsilon_\tau$$

where $HV(Y_N, r)$ is the hypervolume indicator value of the Pareto front approximation delimited from above by the reference objective vector $\mathbf{r}$. By convention, the hypervolume indicator does not consider elements of Y_N which do not dominate $\mathbf{r}$. If all elements of Y_N do not dominate $\mathbf{r}$, then $HV(Y_N; \mathbf{r}) = 0$.

Given a problem $p \in \mathcal{P}$, the Pareto front reference Y^p is constructed using the best feasible non dominated points found by all considered solvers on this problem for a maximum budget of evaluations. More precisely, the Pareto front reference is computed by removing the dominated points found in the union of the Pareto front approximations generated by the set of solvers on Problem $p \in \mathcal{P}$ once the budget of functions evaluations is exhausted.

Stochastic algorithms are commonly used to tackle blackbox multiobjective optimization problems. To include them into the data profiles framework, one can consider that different instances of a given problem obtained by different random seeds constitute different problems. This augmented set of problems can be used to construct classical data profiles as explained above. The authors in [50] adopt this approach. However, this approach does not enable to visualize the variability of stochastic algorithms on a given set of problems $\mathcal{P}$. This work proposes another approach, inspired by [228, 229].

Consider a stochastic algorithm $a \in \mathcal{A}$ and a set of problems $\mathcal{P}$. Assume the algorithm has generated after e evaluations different Pareto front approximations $Y^e_{a_1,p}, Y^e_{a_2,p}, \ldots, Y^e_{a_q,p}$ corresponding to $q = |\mathcal{I}_a|$ different instances for all the problems $p \in \mathcal{P}$. For each instance, compute the respective hypervolume values $h^e_{a_1,p}, h^e_{a_2,p}, \ldots, h^e_{a_q,p}$ as described previously. A specific instance $a_j \in \mathcal{I}_a$ of the stochastic algorithm $a \in \mathcal{A}$ is said to solve the problem $p \in \mathcal{P}$ with accuracy $\varepsilon_\tau > 0$ if:

$$\frac{h^e_{a,p}}{HV(T(Y^p), T(\tilde{\mathbf{y}}^{N,p}))} \geq 1 - \varepsilon_\tau$$

where Y^p is the Pareto front reference of Problem p and $\tilde{\mathbf{y}}^{N,p}$ its associated approximated Nadir objective vector.

With this convergence test, one can associate a unique data profile d_{a_j} for each of the instances $a_j \in \mathcal{I}_a$ of the stochastic algorithm $a \in \mathcal{A}$ (designed as an *operational characteristic* in [228,

229]). The *average data profile* of the stochastic algorithm $a \in \mathcal{A}$ is then defined as

$$\bar{d}_a : k \in \mathbb{N} \mapsto \frac{1}{|\mathcal{I}_a|} \sum_{j=1}^{|\mathcal{I}_a|} d_{a_j}(k).$$

Similarly, the *lower bound* and *upper bound data profiles* of the stochastic algorithm $a \in \mathcal{A}$ are respectively defined as

$$d_a^l : k \in \mathbb{N} \mapsto \min_{1 \leq j \leq |\mathcal{I}_a|} d_{a_j}(k)$$

and

$$d_a^u : k \in \mathbb{N} \mapsto \max_{1 \leq j \leq |\mathcal{I}_a|} d_{a_j}(k).$$

These two data profiles delimit the variations of the performance of a given stochastic algorithm for a given set of instances.

5.6.3 Comparing different variants of DMulti-MADS

The aim of the tests presented in this section is to understand the impact of the different algorithmic options of Dmulti-MADS on its computational efficiency. The experiments focus on three parameters:

- The *success iteration condition*: for DMulti-MADS, an iteration is said to be successful if the algorithm generates a new point which dominates the current poll center, named the strict success strategy. In order to compare with the DMS success strategy, a DMS strategy version of the DMulti-MADS algorithm is implemented. Specifically, an iteration of the DMS strategy version is marked as a success if a new non dominated point is found. The selection of the poll center and the update step of the new points remain the same; in case of a success, the poll center mesh size and frame size parameters remain constant. Note that with these conditions, it is possible to prove the same convergence results as for the DMulti-MADS algorithm.

- The *choice of the current incumbent*. Two selection strategies are considered. The first one picks the first element of the iterate list with the maximum mesh size and frame size parameters. The second one selects the poll center according to Algorithm 8, which includes the spread of the current non dominated points in its selection criterion.

- The *opportunistic strategy*. If the opportunistic strategy is activated, the iteration is stopped as soon as the algorithm finds a new point which triggers the success condition. Otherwise, the iteration continues until the end of the poll step.

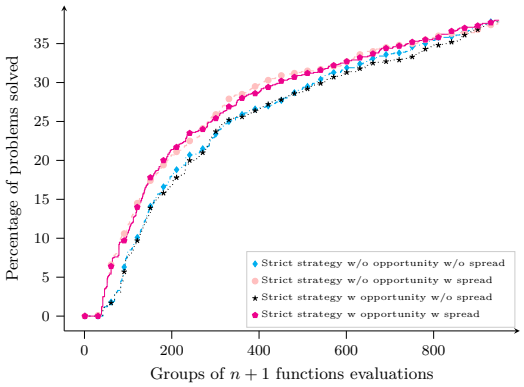
These 8 variants are implemented in `Julia` and can be found at `https://github.com/bbopt/DMultiMadsEB`. For all variants, a speculative search strategy is implemented on the model of [15], as follows: considering the incumbent point $\mathbf{x}^k$ at iteration $k > 0$ generated during iteration $l < k$ with incumbent poll center $\mathbf{x}^l$, one can build the target direction [22] $\mathbf{w}^k = \mathbf{x}^k - \mathbf{x}^l$. If no failure iteration was ever observed at $\mathbf{x}^k$, the search point $\mathbf{s}^k = \mathbf{x}^k + \mathbf{w}^k$ is firstly evaluated before executing the poll step. Moreover, our implementation of the DMulti-MADS exploits the target direction $\mathbf{w}^k$ to reduce the number of polling directions generated by OrthoMADS to $n + 1$ directions without models as described in [22]. Preliminary tests show that this approach is more efficient than the classical OrthoMADS strategy [2] with $2n$ polling directions. This implementation is also the standard one in the `NOMAD` software when all models are deactivated for single objective optimization.

The mesh is implemented using the granular mesh strategy devised in [18]. All variants stop as soon as one component of the mesh size vector is below 10^{-9} or reach a maximum number of $30,000$ evaluations. For each problem, the variants start from the same set of initial points using the linesearch starting strategy exposed in [74]. For each problem and each variant, 10 replications are run by changing the random seed which controls the generation of the polling directions.

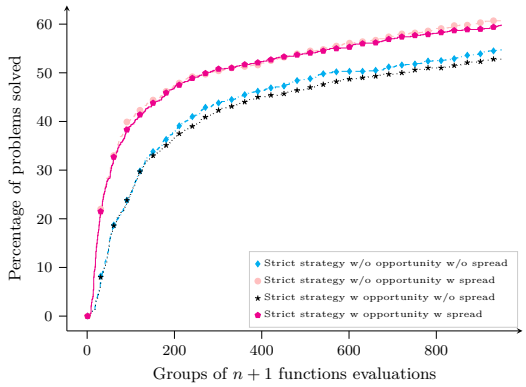
All variants use the fixed integer parameter $w^+ = 3$. For more details about this choice, the reader is invited to consult Appendix 5.8.

Figure 5.10 shows the data profiles for the set of variants implementing the strict success strategy with tolerance accuracies $\varepsilon_\tau \in \{10^{-2}, 5 \times 10^{-2}, 10^{-1}\}$. From these graphs, one can note that both strict strategy variants coupled with the spreading strategy outperform the remaining variants without the spreading strategy. From this figure, one can equally observe that the strict success strategies without opportunistic polling slightly perform better than their counterpart variants with opportunistic polling. Similar observations can be done for the set of DMulti-MADS variants with DMS strategy, as shown in Figure 5.11.

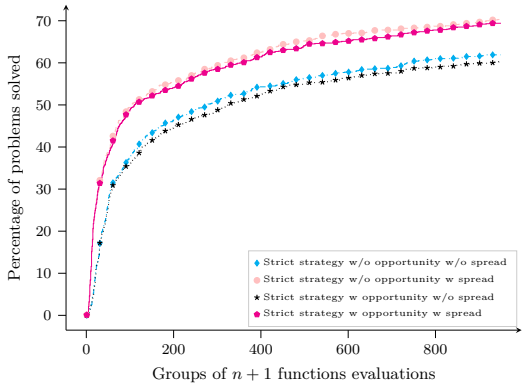
To select the most efficient strategy variants among all DMulti-MADS variants, Figure 5.12 shows the data profiles for the four best variants with tolerance accuracies $\varepsilon_\tau \in \{10^{-2}, 5 \times 10^{-2}, 10^{-1}\}$. From these graphs, one can observe that for a lower budget of evaluations (i.e. inferior to 200 $(n + 1)$ evaluations), strict success strategies variants with spread solve slightly more problems than DMS strategies variants with spread. However, for a larger budget of evaluations, using the DMS strategy performs better than using the strict strategy with and without opportunity. For the DMS strategy and strict strategy variants and for any budget on this set of problems, evaluating points opportunistically does not bring considerable advantages. For the remaining tests below, only the strict success and DMS success strategies



(a) $\varepsilon_\tau = 10^{-2}$.



(b) $\varepsilon_\tau = 5 \times 10^{-2}$.



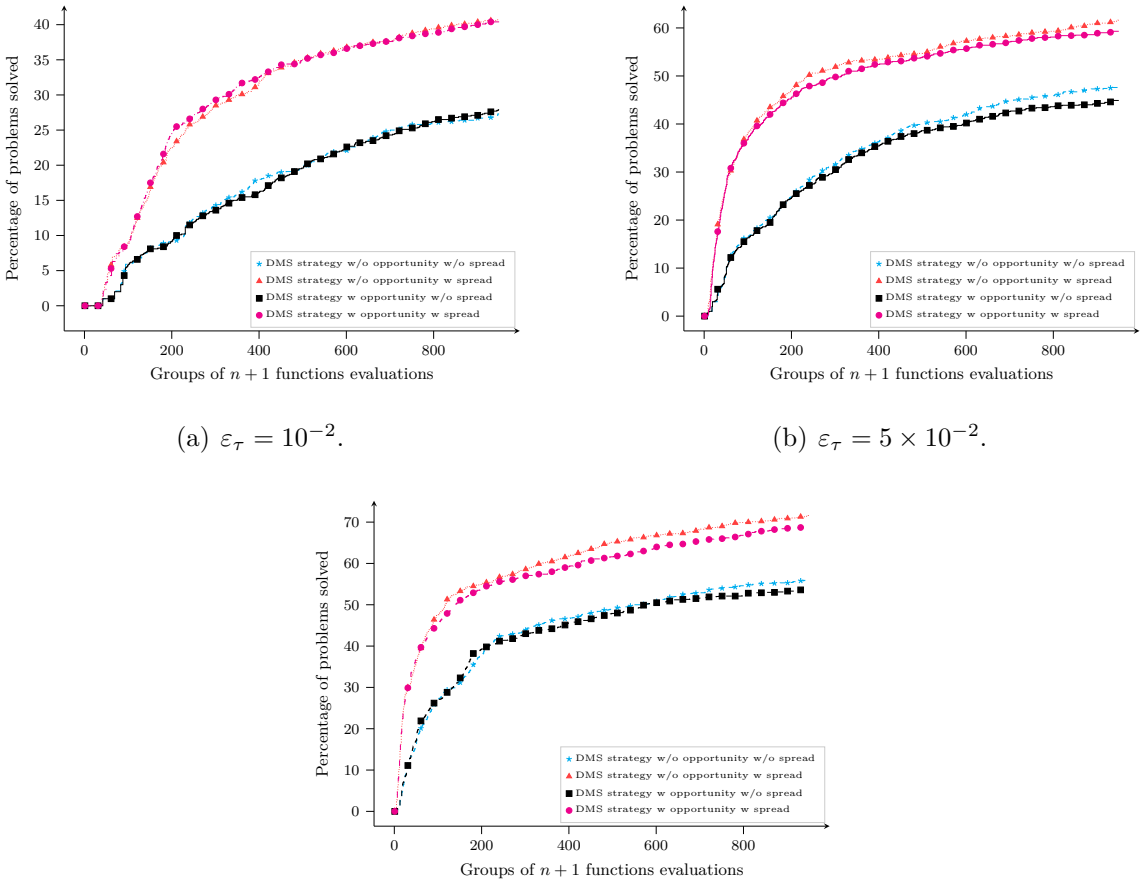
(c) $\varepsilon_\tau = 10^{-1}$.

Figure 5.10 Data profiles obtained on 10 replications from 100 multiobjective optimization problems from [74] for DMulti-MADS strict success strategy variants with tolerance $\varepsilon_\tau \in \{10^{-2}, 5 \times 10^{-2}, 10^{-1}\}$.

variants without opportunistic evaluation are kept.

5.6.4 Comparing DMulti-MADS with other algorithms

This section presents the comparison of the two best DMulti-MADS variants coded in `Julia` with other multiobjective derivative-free solvers BiMADS, DMS, MOIF and NSGA-II. The DMS and MOIF solvers have been used with their default settings as described in [74, 126]. Two variants of BiMADS, based on `NOMAD` version 3.9.1, are considered. The first uses the default settings of the MADS algorithm as implemented in `NOMAD` with state-of-the-art search step heuristics (see [22, 29, 66] for more details). The second one deactivates the search step heuristics such that the settings are equivalent to the DMulti-MADS implementation for a



(a) $\varepsilon_\tau = 10^{-2}$.

(b) $\varepsilon_\tau = 5 \times 10^{-2}$.

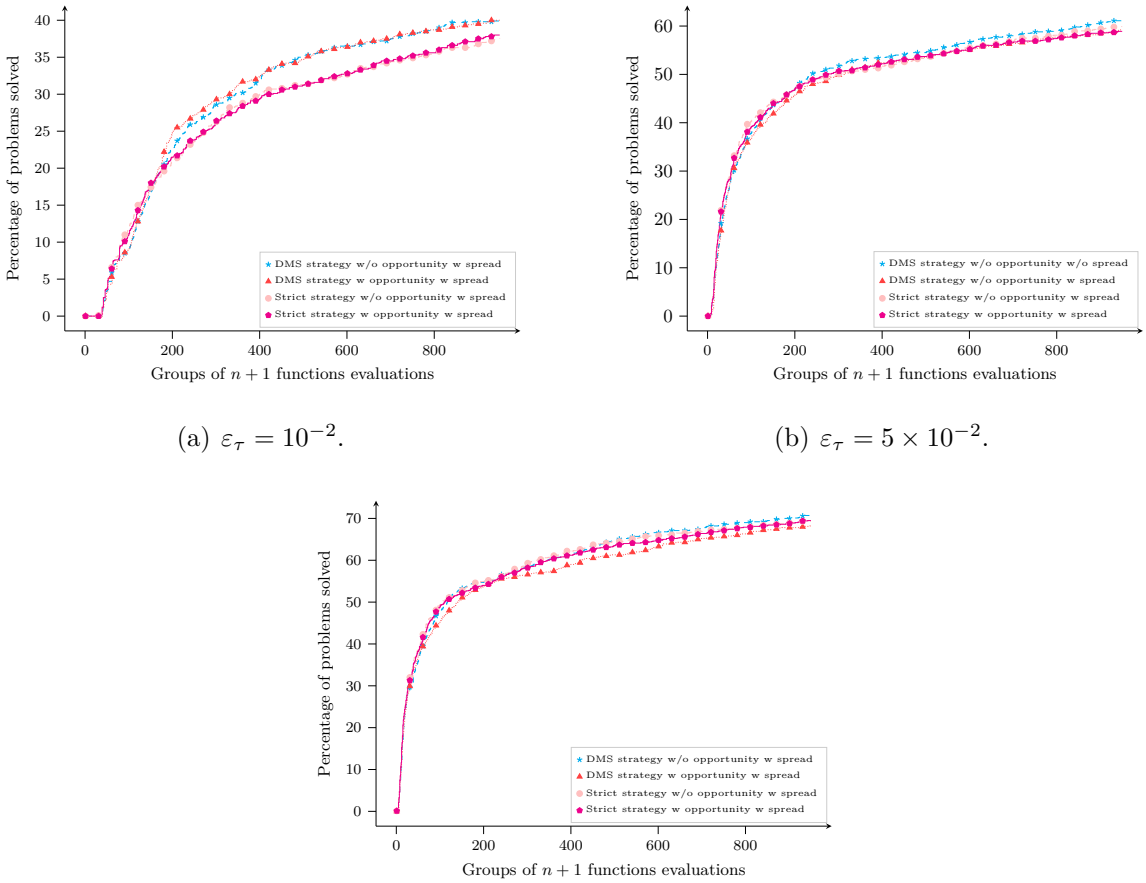
(c) $\varepsilon_\tau = 10^{-1}$.

Figure 5.11 Data profiles obtained on 10 replications from 100 multiobjective optimization problems from [74] for DMulti-MADS variants with DMS strategy with tolerance $\varepsilon_\tau \in \{10^{-2}, 5 \times 10^{-2}, 10^{-1}\}$.

fairer comparison. Specifically, the number of poll directions is set to $n+1$, with a speculative search step strategy enabled and an opportunistic polling strategy. For both variants, all single-objective runs terminate when the mesh or step size parameter is below a threshold value (see [27] for specific details). All these deterministic solvers have a maximum budget of evaluations equal to $30,000$ and start from the same set of initial points as described before.

For NSGA-II, the population size is fixed to 100 points, with a total number of generations equal to 300, which is equivalent to a budget of $30,000$ blackbox evaluations. 50 instances of this stochastic solver are considered, corresponding to 50 different random seeds.

Figure 5.13 presents the data profiles obtained by the different solvers for the whole set of multiobjective problems. As BiMADS is a biobjective method, it is then not presented. For



(a) $\varepsilon_\tau = 10^{-2}$.

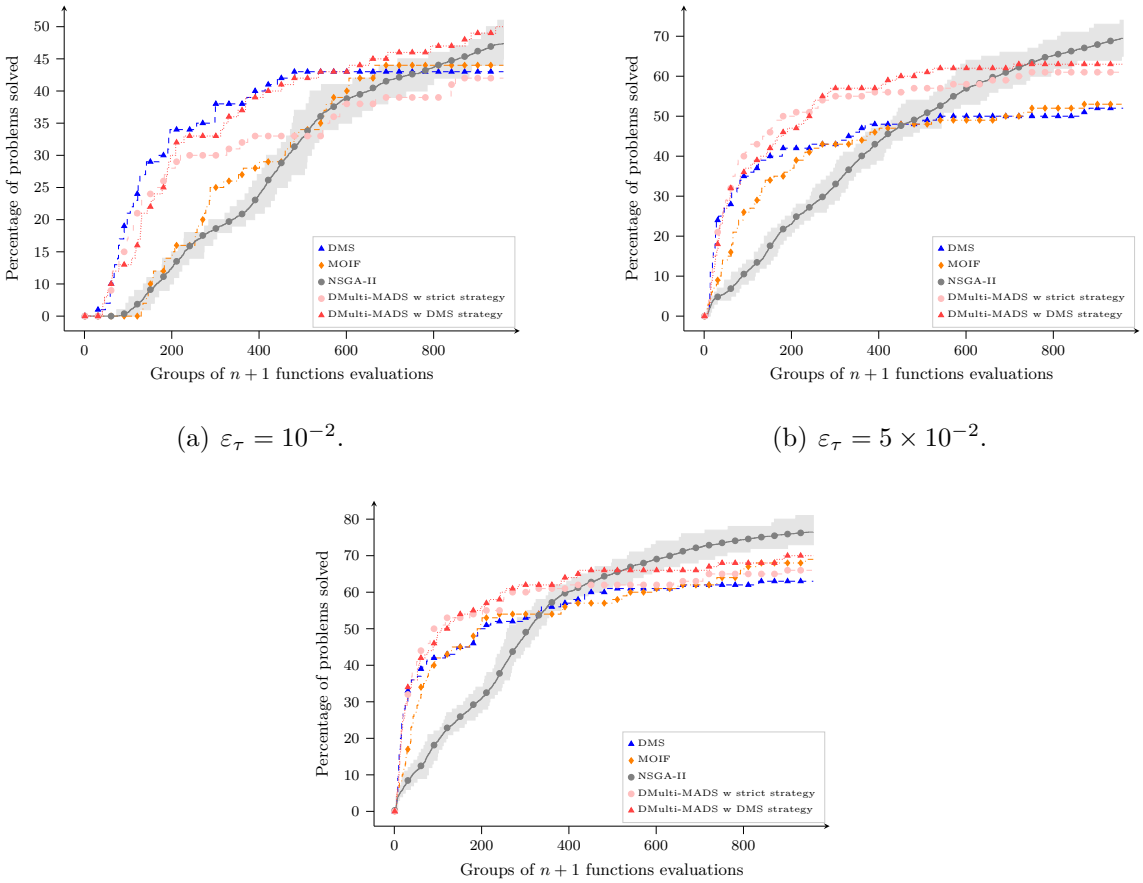
(b) $\varepsilon_\tau = 5 \times 10^{-2}$.

(c) $\varepsilon_\tau = 10^{-1}$.

Figure 5.12 Data profiles obtained on 10 replications from 100 multiobjective optimization problems from [74] for the four best DMulti-MADS variants with tolerance $\varepsilon_\tau \in \{10^{-2}, 5 \times 10^{-2}, 10^{-1}\}$.

the lowest tolerance $\varepsilon_\tau = 10^{-2}$, DMS is better than the other methods for small to medium budgets of evaluations. The DMS success strategy variant of DMulti-MADS outperforms DMS when the allowed budget is high and dominates MOIF and NSGA-II in average 5.13(a). From Figures 5.13(b) and 5.13(c), one can observe that the two DMulti-MADS variants outperform all the other deterministic solvers; the performance of NSGA-II is better for high budgets of evaluations (i.e. for example in a situation when blackbox functions are cheap to evaluate). Choosing the DMS success strategy increases the global performance of DMulti-MADS for medium to high budgets of evaluations.

From the data profiles obtained on the set of problems for $m = 2$ in Figure 5.14, where `NOMAD` (BiMADS) is added, similar results can be observed. The DMS solver is better than the others solvers with a small to moderate budget of blackbox evaluations but gets outperformed



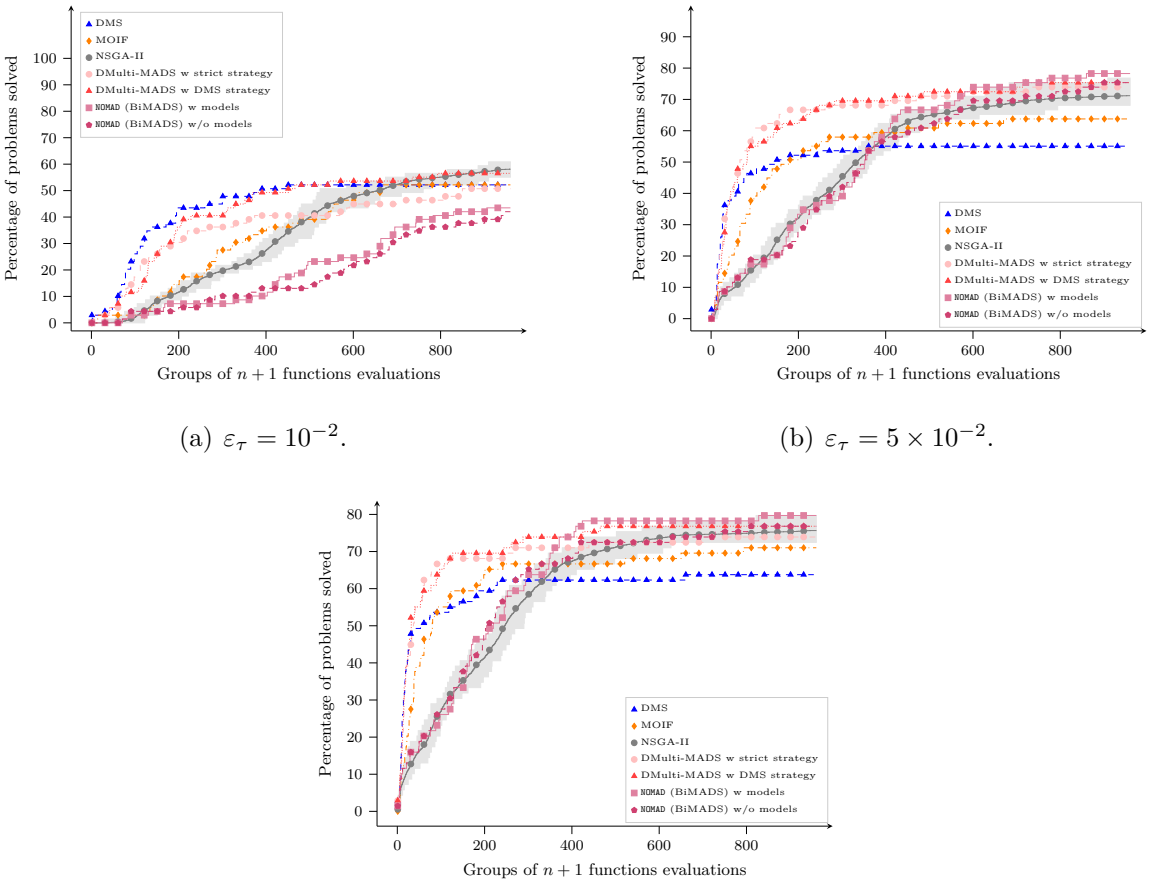
(a) $\varepsilon_\tau = 10^{-2}$.

(b) $\varepsilon_\tau = 5 \times 10^{-2}$.

(c) $\varepsilon_\tau = 10^{-1}$.

Figure 5.13 Data profiles using DMS, DMulti-MADS, MOIF and NSGA-II obtained on 100 multiobjective optimization problems (69 with $m = 2$, 29 with $m = 3$ and 2 with $m = 4$) from [74] with 50 different runs of NSGA-II with tolerance $\varepsilon_\tau \in \{10^{-2}, 5 \times 10^{-2}, 10^{-1}\}$.

by the DMulti-MADS DMS success strategy variant when one chooses a high budget of evaluations, for the lowest tolerance. For higher tolerances, Figures 5.14(b) and 5.14(c) illustrate the fact that the two DMulti-MADS variants are the dominating algorithms for a low to moderate budget of evaluations. For a high budget of evaluations and higher tolerances, `NOMAD` (BiMADS) outperforms DMulti-MADS. However, it exploits surrogate models which considerably improve its performance. When they are deactivated, DMulti-MADS is better for all considered budgets and all considered tolerances.



(a) $\varepsilon_\tau = 10^{-2}$.

(b) $\varepsilon_\tau = 5 \times 10^{-2}$.

(c) $\varepsilon_\tau = 10^{-1}$.

Figure 5.14 Data profiles using NOMAD (BiMADS), DMS, DMulti-MADS, MOIF and NSGA-II obtained on 69 biobjective optimization problems from [74] with 50 different runs of NSGA-II with tolerance $\varepsilon_\tau \in \{10^{-2}, 5 \times 10^{-2}, 10^{-1}\}$.

5.7 Conclusion

This work proposes a new extension of the MADS algorithm to multiobjective optimization, inspired by the works of [27, 74]. Contrary to the BIMADS and MultiMADS methods, the DMulti-MADS algorithm does not solve a succession of single-objective parameterized formulations. It directly updates a current list of non dominated points which gets closer to the Pareto front. This enables a better management of a given budget of evaluations to explore the feasible objective set. As in single-objective optimization, each iteration is built around a search and poll step. Theoretically, it is proved under mild assumptions that DMulti-MADS generates sequences of points whose stationary points are local Pareto

optimal[6]. This convergence result is an alternative to the proof presented in [74] which guarantees that the DMS algorithm is able to converge to at least one local Pareto optimal point[7]. However, the flexibility to choose the poll center, as it is the case for DMS, is lost. Computational results show that DMulti-MADS is competitive compared to other state-of-the-art blackbox multiobjective optimization techniques.

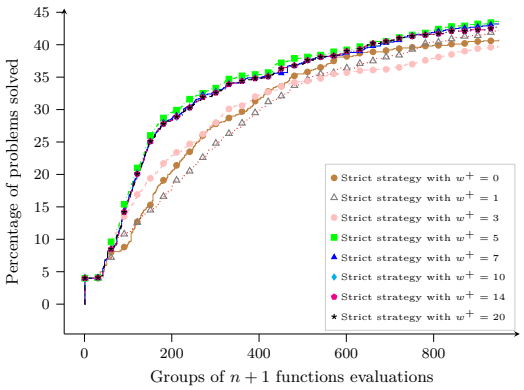
The selection mechanism of the poll center is a central part of the strong convergence properties of DMulti-MADS. Future research directions could adapt the convergence analysis of DMulti-MADS to other multiobjective optimization derivative-free methods with a posteriori preferences of articulations [126, 180, 223]. Indeed, all these methods look for improvements of a list of non dominated points possessing their own optimization parameter (trust-region radius, line step, and stepsize). Maximum optimization parameter selection could be tested on these methods.

In addition, many extensions could be implemented to improve performance of DMulti-MADS: search strategies assisted by surrogate models [53, 66] or global search strategies [73], parallelism, taking into account general inequality constraints [180] and so on. An integration of this algorithm in `NOMAD` is also planned.

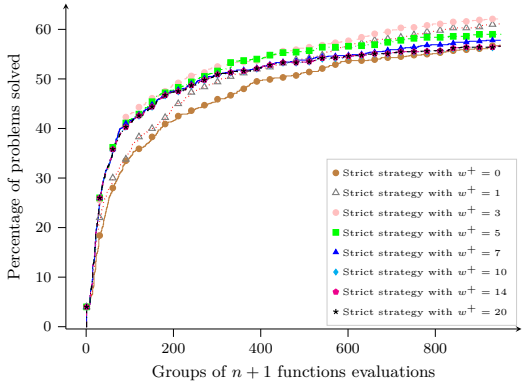
[6]This sentence was modified according to the published article.
[7]This sentence was modified according to the published article.

5.8 A study of the influence of the integer parameter w^+ on the performance of the DMulti-MADS algorithm

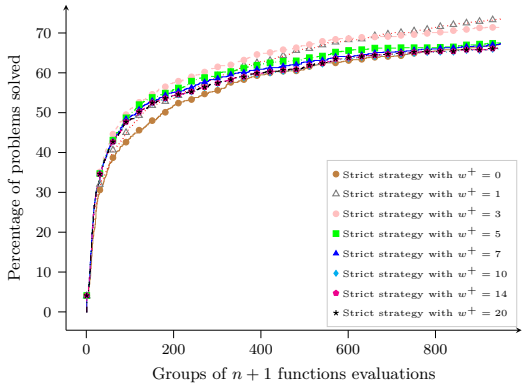
This appendix presents some experiments which led to the choice of the value of the integer parameter w^+. For readability, only the DMulti-MADS variants without opportunistic polling and the spread strategy are considered. For the set of w^+ integer values presented here, similar observations can be done as the ones presented in Subsection 5.6.3 concerning the influence of opportunistic polling and the spread strategy on the concrete performance of the DMulti-MADS algorithm. Thus, the variants compared here are the better ones for each considered w^+ value. They use the same settings as the ones described at the beginning of Subsection 5.6.3.



(a) $\varepsilon_\tau = 10^{-2}$.



(b) $\varepsilon_\tau = 5 \times 10^{-2}$.

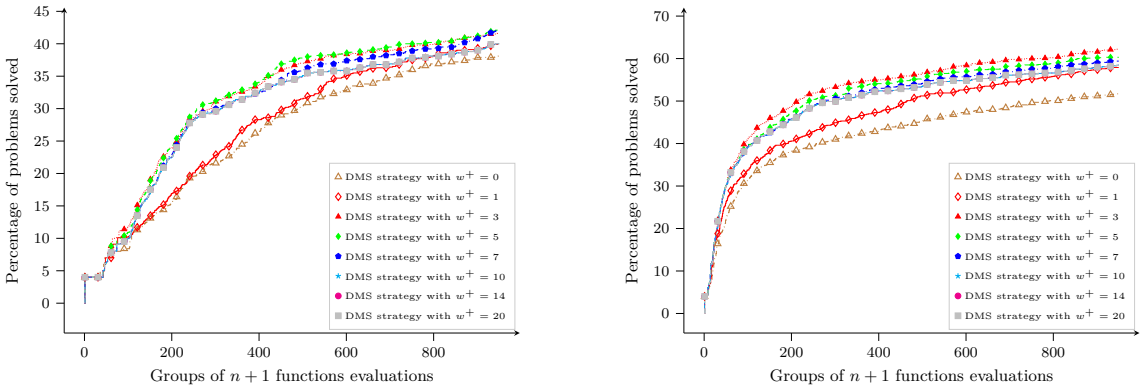


(c) $\varepsilon_\tau = 10^{-1}$.

Figure 5.15 Data profiles obtained on 10 replications from 100 multiobjective optimization problems from [74] for DMulti-MADS variants with strict success strategy without opportunistic polling and with spread strategy for tolerance $\varepsilon_\tau \in \{10^{-2}, 5 \times 10^{-2}, 10^{-1}\}$.

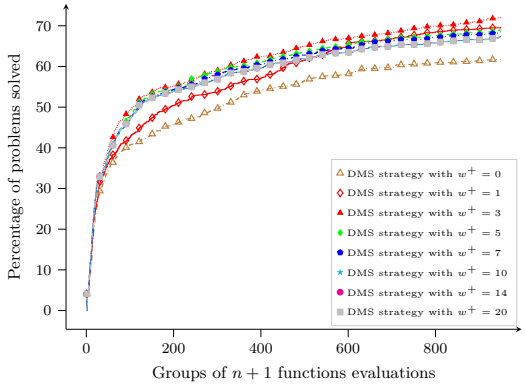
Figure 5.15 presents data profiles of the DMulti-MADS strict success strategy variants for several integer values of w^+. Fixing w^+ to 0 means that only points with maximum frame size parameter in the current Pareto front approximation can be selected as current poll centers. When w^+ is high (for example $w^+ \in \{10, 14, 20\}$), the poll selection of the algorithm is similar to the one of the DMS algorithm: all points of the current incumbent list can be chosen as poll centers. From Figure 5.15, one can note that allowing only points with maximum frame size parameters to be selected as poll centers grandly decreases the performance of DMulti-MADS for all considered tolerances. However, removing all restrictions on the choice of the current incumbent as long as it belongs to the current incumbent list is not the most performant variant (i.e. for example with $w^+ \in \{10, 14, 20\}$). Indeed, for the lowest tolerance 5.15(a), the data profiles reveal than strict strategy variants with high value of $w^+ \in \{10, 14, 20\}$ solve slightly less problems compared to the choice of $w^+ = 5$. For higher tolerances, a value $w^+ \in \{3, 5\}$ is preferable, as shown in Figures 5.15(b) and 5.15(c).

One can make similar observations when comparing DMulti-MADS variants with DMS success strategy for different w^+ values, as shown on Figure 5.16. For the DMS success strategy, a w^+ integer value comprised between 3 and 5 implies a better performance.



(a) $\varepsilon_\tau = 10^{-2}$.

(b) $\varepsilon_\tau = 5 \times 10^{-2}$.



(c) $\varepsilon_\tau = 10^{-1}$.

Figure 5.16 Data profiles obtained on 10 replications from 100 multiobjective optimization problems from [74] for DMulti-MADS variants with DMS strategy with tolerance $\varepsilon_\tau \in \{10^{-2}, 5 \times 10^{-2}, 10^{-1}\}$.

5.9 Comparing DMulti-MADS with other algorithms: performance profiles

This appendix reports comparison results between the two best DMulti-MADS variants and the other multiobjective solvers BiMADS, DMS, MOIF and NSGA-II in term of the purity metric, spread metrics Γ and Δ metrics as described and used in [74]. All settings for running the solvers are the same as the ones described in Subsection 5.6.4. Specifically the maximal allowed number of evaluations for each problem and each solver is fixed to $30,000$. Algorithms are compared in pairs (see [74, 130] for an explanation).

When looking at Figure 5.17, one can observe that the two variants of DMulti-Mads are more efficient in terms of purity than DMS and MOIF. On the contrary, they are less efficient than BiMADS (with and without models) and NSGA-II (worst and best versions) in terms of

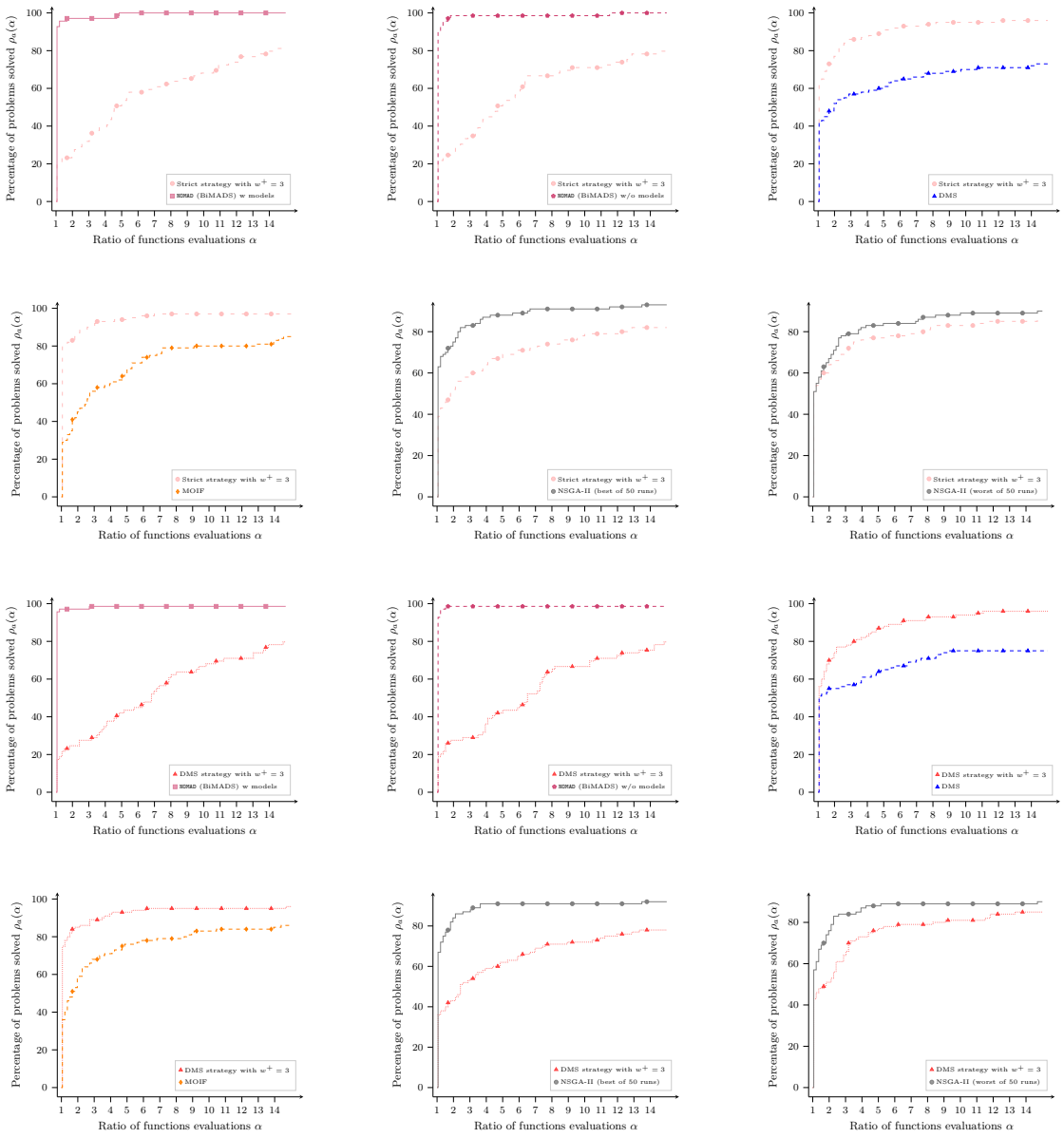


Figure 5.17 Purity performance profiles using NOMAD (BiMADS), DMS, DMulti-MADS, MOIF and NSGA-II obtained on 100 multiobjective optimization problems (69 with $m = 2$, 29 with $m = 3$ and 2 with $m = 4$) from [74] with 50 different runs of NSGA-II.

purity metric. This can be explained by the fact that BiMADS generates more points in the Pareto front reference, due to its scalarization approach when DMulti-MADS generates points that are close to the Pareto front reference, but not part of it. Concerning NSGA-II, a closer look at the runs shows that all deterministic solvers can stop before the exhaustion of the whole budget of evaluations (because the solver reaches a threshold), which can prevent them from exploring potential interesting areas in the objective space. NSGA-II always

exploits its full budget of evaluations, which allows to generate more points in the Pareto front reference, and consequently to have a better purity metric.

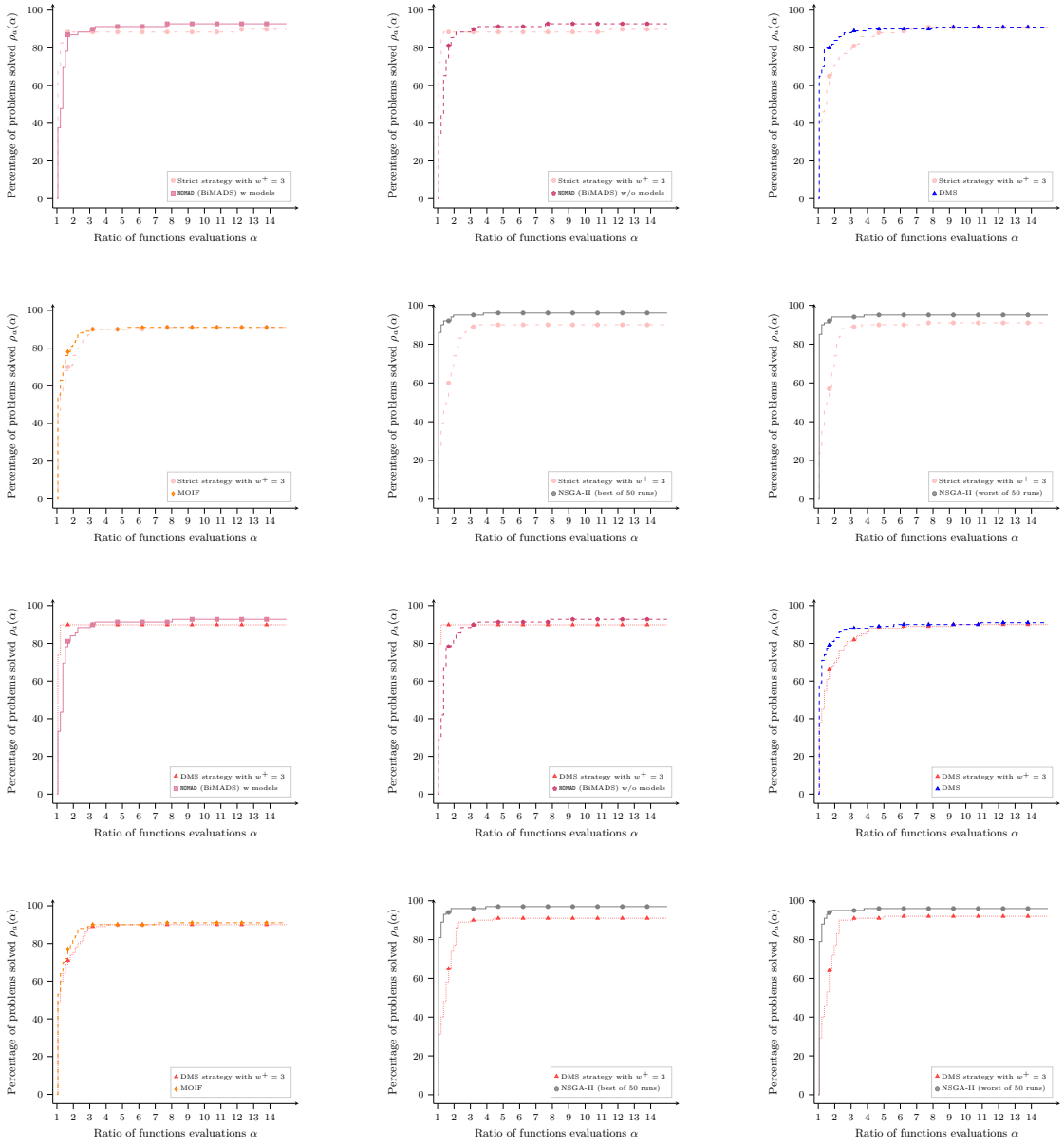


Figure 5.18 Δ spread performance profiles using NOMAD (BiMADS), DMS, DMulti-MADS, MOIF and NSGA-II obtained on 100 multiobjective optimization problems (69 with $m = 2$, 29 with $m = 3$ and 2 with $m = 4$) from [74] with 50 different runs of NSGA-II.

In terms of Δ spread metric results, reported in Figure 5.18, one can see that the two DMulti-MADS variants are slightly less performant than the other algorithms. For DMS and MOIF, the use of coordinate directions seems to play a role in the distribution of their generated

points for this set of problems. However, the use of dense directions enables DMulti-MADS to find new non-dominated points contrary to DMS and MOIF as shown in Figure 5.17.

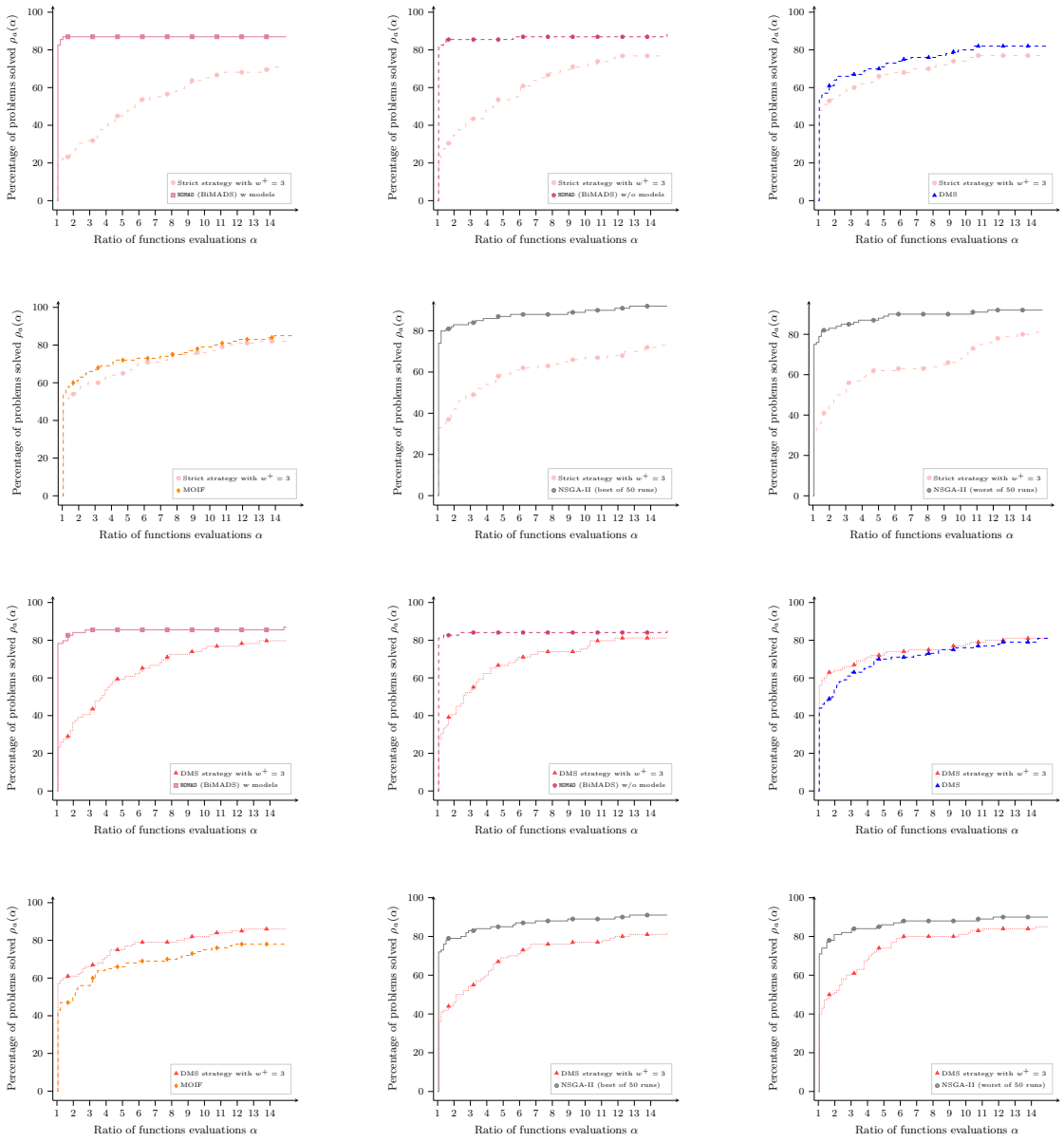


Figure 5.19 Γ spread performance profiles using NOMAD (BiMADS), DMS, DMulti-MADS with DMS strategy, MOIF and NSGA-II obtained on 100 multiobjective optimization problems (69 with $m = 2$, 29 with $m = 3$ and 2 with $m = 4$) from [74] with 50 different runs of NSGA-II.

In terms of Γ spread metric, both variants of DMulti-MADS generate less dense Pareto front approximations than BiMADS and NSGA-II, as shown in Figure 5.19. The DMulti-MADS variant with DMS strategy performs better when compared to MOIF and DMS than

the DMulti-MADS variant with strict success strategy. Thus, for an important budget of evaluations, the DMulti-MADS variant with DMS strategy generates denser Pareto front approximations.

Acknowledgments. The authors would like to thank Professor Ana Luísa Custódio (Universidade Nova de Lisboa) for interesting discussions on derivative-free multiobjective optimization.

CHAPTER 6 ARTICLE 3: HANDLING OF CONSTRAINTS IN MULTIOBJECTIVE BLACKBOX OPTIMIZATION

La probabilité de réussir la mise sur orbite d'une fusée est d'une chance sur un million. Dépêchons-nous de rater 999.999 lancements !

Jacques Rouxel, *Les Shadoks.*

Authors Jean Bigeon, Sébastien Le Digabel and Ludovic Salomon

Journal Submitted in *European Journal of Operational Research*

Abstract This work proposes the integration of two new constraint-handling approaches into the blackbox constrained multiobjective optimization algorithm DMulti-MADS, an extension of the Mesh Adaptive Direct Search (MADS) algorithm for single-objective constrained optimization. The constraints are aggregated into a single constraint violation function which is used either in a two-phase approach, where the search for a feasible point is prioritized if not available before improving the current solution set, or in a progressive barrier approach, where any trial point whose constraint violation function values are above a threshold are rejected. This threshold is progressively decreased along the iterations. As in the single-objective case, it is proved that these two variants generate feasible and/or infeasible sequences which converge either in the feasible case to a set of local Pareto optimal points or in the infeasible case to Clarke stationary points according to the constraint violation function. Computational experiments show that these two approaches are competitive with other state-of-the-art algorithms.

Keywords Multiobjective optimization, derivative-free optimization, blackbox optimization, constrained optimization, Clarke analysis.

6.1 Introduction

This work considers the following constrained multiobjective optimization problem

$$MOP : \min_{\mathbf{x}\in\Omega} f(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x}))^\top$$

where $\Omega = \{\mathbf{x} \in \mathcal{X} : c_j(\mathbf{x}) \leq 0,\ j \in \mathcal{J}\} \subset \mathbb{R}^n$ is the *feasible decision set*, and $\mathcal{X}$ a subset of $\mathbb{R}^n$. $\mathbb{R}^n$ and $\mathbb{R}^m$ are respectively designed as the *decision space* and the *objective space*. The functions $f_i : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ for $i = 1, 2, \ldots, m$ and $c_j : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ for $j \in \mathcal{J}$ are the outputs of a program seen as a blackbox. In this context, no gradient is available nor cannot be approximated and one cannot make any assumption on the structure of the problem (differentiability, continuity, convexity) in absence of analytical expressions for the objective and the constraint functions. Many engineering applications, which involve several, costly and conflicting objectives, over a given set of constraints, fit into this framework (see for example [6, 92, 107, 232]). For more general information on derivative-free methods, the reader is referred to [20, 67, 164].

The description of the feasible decision set Ω enables the modeller to distinguish different types of constraints [168]. The set $\mathcal{X}$ is the set of *unrelaxable constraints*, which cannot be violated along the optimization process (e.g., strict bound constraints). The constraints $c_j(x) \leq 0,\ j \in \mathcal{J}$, constitute the set of *relaxable and quantifiable constraints*, that can be violated during the optimization, i.e., the blackbox will execute and the constraints outputs can be aggregated as a measure of violation of the constraints. Finally, *hidden constraints* constitute the set of points in the decision space for which the blackbox does not return any value, typically when the blackbox fails to execute. Allowing the f_i and c_j functions to take infinity values refers to this last type of constraints.

Furthermore, in a multiobjective optimization context, due to the conflict between different objectives, a solution is not always optimal for all criteria. The goal is then to provide the set of best trade-off solutions to the decision maker [45, 65, 190].

In single-objective optimization, many algorithms have been proposed to solve blackbox constrained optimization problems: direct search methods via the use of a filter [14, 16], a merit function [133] or an augmented Lagrangian [161], a derivative-free linesearch algorithm coupled with a penalty function [177], or quadratic model-based approaches (see for example [12, 33, 94]). The reader can refer to [164, Section 7] for a more thorough review.

Evolutionary algorithms [45] are popular methods to tackle constrained multiobjective optimization blackbox problems. Firstly investigated in the context of bound-constrained or unconstrained blackbox optimization, researchers have adapted some of them to take into account inequality constraints (see [266, 273] for more details). However, these methods are mostly stochastic heuristics. They practically require an important number of evaluations to perform. For example, the authors in [266] suggest a budget comprised between 2×10^5 and 5×10^5 function evaluations in their experiments, which can be impracticable when evaluations are too costly. Surrogate models remove this limitation by substituting true blackboxes

by less expansive surrogates, such as radial basis functions (see [215]), or Gaussian processes (see [111]).

Recently, researchers have proposed extensions of convergence-based deterministic single-objective methods to multiobjective constrained optimization. Among the first ones, BiMADS [27] and MultiMADS [28] are scalarization-based algorithms. These two algorithms reformulate the multiobjective optimization problem into a succession of single-objective subproblems. Each of them is solved by the single-objective constrained blackbox MADS algorithm [15, 16]. Practically, it can be difficult to correctly allocate the total budget of evaluations between all the subproblems, potentially resulting in a loss of evaluations required to improve the diversity and density of the current solution set.

The Direct MultiSearch (DMS) [74], its variants [53, 73, 90], and Derivative-Free MultiObjective (DFMO) [180] algorithms consider a different approach. They all keep a list of current feasible best non-dominated solutions that they improve along the iterations. DMS extends single-objective direct search algorithms to constrained multiobjective optimization. It rejects non-feasible points via the use of an extreme barrier function approach, i.e., non feasible points are assigned infinity values. This approach does not exploit knowledge of constraint violations values, which could potentially help to improve the solution set. Furthermore, DMS imposes the use of a feasible starting point, which is not practically available (too costly) in a real engineering context. DFMO extends a derivative-free linesearch algorithm to constrained multiobjective optimization. By aggregating constraints with the objective functions via the use of a penalty function, DFMO reduces the initial constrained multiobjective optimization problem to a simple bound constrained multiobjective optimization problem, easier to solve. However, its convergence assumptions are a bit restrictive in a blackbox optimization context, i.e., constraints functions and objective functions must be Lipschitz continuous. On the contrary, DMS requires that objective functions should satisfy locally Lipschitz continuity. Besides, penalty function approaches can be sensitive to the scale of constraints (not always available in a blackbox context) and their penalty parameters values.

Based on these remarks, this work proposes two other ways to handle blackbox constraints, based on the DMulti-MADS algorithm [40]. DMulti-MADS is an extension of the MADS algorithm to multiobjective optimization, strongly inspired by the DMS and BiMADS algorithms. It possesses convergence properties similar to DMS. At the same time, experiments have shown its competitiveness according to state-of-the-art solvers on synthetic bound-constrained problems [40]. Similarly to DMS, the first version of DMulti-MADS, described in [40] requires a feasible starting point. The two extensions described below remove this limitation. The first one is an extension of the two-phase MADS algorithm described in [17] to

constrained multiobjective optimization, named DMulti-MADS-TEB. The second version is an extension of the MADS algorithm with progressive barrier [16] to constrained multiobjective optimization, designed as DMulti-MADS-PB. Contrary to a penalty function approach, this last method

- is less sensitive to the scale of the outputs of the blackbox as it does not aggregate the constraints with the objective function;
- allows to explore around several incumbent points and not just one;
- proves convergence results assuming local Lipschitz continuity of the problem functions[1].

This work is organized as follows. Section 6.2 provides a summary of multiobjective optimization concepts. Section 6.3 introduces the core elements of the DMulti-MADS algorithm. Section 6.4 describes the DMulti-MADS-TEB and DMulti-MADS-PB variants to handle blackbox constraints. Main convergence results are detailed in Section 6.5. Finally, experiments are conducted in Section 6.6 on synthetic benchmarks and three real engineering applications in comparison with other state-of-the-art solvers.

6.2 Pareto dominance and optimal solutions in multiobjective optimization

This section summarizes some notation and concepts of multiobjective optimization. In order to characterize optimal solutions, one needs the concept of *Pareto dominance* [190].

Definition 45. Given two feasible decision vectors $\mathbf{x}^1$ and $\mathbf{x}^2$ in Ω,

- $\mathbf{x}^1 \preceq \mathbf{x}^2$ ($\mathbf{x}^1$ *weakly dominates* $\mathbf{x}^2$) if and only if $f_i(\mathbf{x}^1) \leq f_i(\mathbf{x}^2)$ for $i = 1, 2, \ldots, m$.
- $\mathbf{x}^1 \prec \mathbf{x}^2$ ($\mathbf{x}^1$ *dominates* $\mathbf{x}^2$) if and only if $f_i(\mathbf{x}^1) \leq f_i(\mathbf{x}^2)$ for $i = 1, 2, \ldots, m$ and there exists at least an index $i_0 \in \{1, 2, \ldots, m\}$ such that $f_{i_0}(\mathbf{x}^1) < f_{i_0}(\mathbf{x}^2)$.
- $\mathbf{x}^1 \sim \mathbf{x}^2$ ($\mathbf{x}^1$ and $\mathbf{x}^2$ are *incomparable*) if and only if $\mathbf{x}^1$ does not dominate $\mathbf{x}^2$ and $\mathbf{x}^2$ does not dominate $\mathbf{x}^1$.

With this definition, one is able to characterize locally optimal solutions and global optimal solutions in a multiobjective context.

[1] This sentence was modified according to the submitted article.

Definition 46. A feasible decision vector $\mathbf{x}^\star \in \Omega$ is said to be (globally) *Pareto optimal* if it does not exist any other decision vector $\mathbf{x} \in \Omega$ which dominates $\mathbf{x}^\star$.

Definition 47. A feasible decision vector $\mathbf{x}^\star \in \Omega$ is said to be *locally Pareto optimal* if it exists a neighbourhood $\mathcal{N}(\mathbf{x}^\star)$ of $\mathbf{x}^\star$ such that there does not exist any other decision vector $\mathbf{x} \in \mathcal{N}(\mathbf{x}^\star) \cap \Omega$ which dominates $\mathbf{x}^\star$.

The set of all Pareto optimal solutions in Ω is called the *Pareto set* denoted by $\mathcal{X}_P$ and its image by the objective function is designed as the *Pareto front* denoted by $\mathcal{Y}_P \subseteq \mathbb{R}^m$. Any set of local Pareto optimal solutions is called a *local Pareto set*. Ideally, one would wish to find the entire Pareto set and consequently the entire Pareto front. But the Pareto set may be composed of an infinite number of solutions. In practice, an algorithm tries to find a representative set of nondominated points, denoted as a *Pareto set approximation* [277] (its mapping by the objective function f is designed as a *Pareto front approximation*). In the best case, a Pareto set approximation should be a subset of the Pareto set or a locally Pareto set, but this condition is not always satisfied.

Several objective vectors, i.e., points in the objective space, play an important role in multiobjective optimization as bounds on the Pareto front. The *ideal objective vector* [190] $\mathbf{y}^I \in \mathbb{R}^m$ bounds the Pareto front from below and is defined as

$$\mathbf{y}^I = \left(\min_{\mathbf{x}\in\Omega} f_1(\mathbf{x}), \min_{\mathbf{x}\in\Omega} f_2(\mathbf{x}), \ldots, \min_{\mathbf{x}\in\Omega} f_m(\mathbf{x}) \right)^\top.$$

From each component of the ideal objective vector, one can obtain information on the *extreme points of the Pareto set*, i.e., the elements of the Pareto set and solutions of each single-objective problem $\min_{\mathbf{x}\in\Omega} f_i(\mathbf{x})$ for $i = 1, 2, \ldots, m$. The *nadir objective vector* [190] $\mathbf{y}^{\mathrm{N}} \in \mathbb{R}^m$ provides an upper bound on the Pareto front. It is defined as

$$\mathbf{y}^{\mathrm{N}} = \left(\max_{\mathbf{x}\in\mathcal{X}_P} f_1(\mathbf{x}), \max_{\mathbf{x}\in\mathcal{X}_P} f_2(\mathbf{x}), \ldots, \max_{\mathbf{x}\in\mathcal{X}_P} f_m(\mathbf{x}) \right)^\top.$$

6.3 The DMulti-MADS algorithm

DMulti-MADS [40] is a direct search iterative method designed to solve constrained multiobjective blackbox optimization problems. It is an extension of the MADS [15] algorithm to multiobjective optimization, strongly inspired by the DMS [74] and BiMADS algorithms [27]. The notations and following definitions are taken from [16, 20].

Definition 48. At iteration k, the *set of feasible incumbent solutions* is defined as

$$F^k = \left\{ \arg\min_{\mathbf{x} \in V^k} \{f(\mathbf{x}) : \mathbf{x} \in \Omega\} \right\}$$

where $V^k \subset \mathcal{X}$ is the set of trial points which have been evaluated by the start of iteration k.

Thus, all points in V^k satisfy the set of unrelaxable constraints $\mathcal{X}$. $V^0 \neq \emptyset \subset \mathcal{X}$ is then the set of starting points provided by the user. DMulti-MADS keeps an *iterate list of best feasible incumbents* found until iteration k, denoted as L_F^k and defined as

$$L_F^k = \left\{ (\mathbf{x}^l, \Delta^l) : \mathbf{x}^l \in F^k \text{ and } \Delta^l > 0,\ l = 1, 2, \ldots, |L_F^k| \right\}$$

where Δ^l is the *frame size parameter* associated to the lth non-dominated point x^l of the list L_F^k. As L_F^k keeps only feasible non-dominated points, it is possible that $|F^k| \neq |L_F^k|$.

At the beginning of each iteration k, DMulti-MADS selects an element $(\mathbf{x}^k, \Delta^k)$ of the list L_F^k as the current *feasible frame center*, and generates a finite number of new candidates. To ensure the convergence properties, all generated candidates during iteration k must belong to the mesh M^k defined by

$$M^k = \bigcup_{\mathbf{x} \in V^k} \{\mathbf{x} + \delta^k \mathbf{D}\mathbf{z} : \mathbf{z} \in \mathbb{N}^{n_D}\} \subset \mathbb{R}^n$$

where $\delta^k > 0$ is the *mesh size parameter*; $\mathbf{D} = \mathbf{G}\mathbf{Z} \in \mathbb{R}^{n \times n_D}$ is a matrix whose columns form a positive spanning set for $\mathbb{R}^n$ (see [20, Chapter 6] or [67, Chapter 2]) for some non-singular matrix $\mathbf{G} \in \mathbb{R}^{n \times n}$ and some integer matrix $\mathbf{Z} \in \mathbb{Z}^{n \times n_D}$. Note that $\mathbf{G}, \mathbf{Z}$ and $\mathbf{D}$ do not depend on the iteration indexes. Generally, $\mathbf{G}$ and $\mathbf{Z}$ are chosen such as $\mathbf{G} = \mathbf{I}_n$ and $\mathbf{Z} = [\mathbf{I}_n \ -\mathbf{I}_n] = \mathbf{D}$, with $\mathbf{I}_n$ the identity matrix of dimensions $n \times n$. Furthermore, the current incumbent selection must satisfy at least the following condition:

$$(\mathbf{x}^k, \Delta^k) \in \left\{ (\mathbf{x}, \Delta) \in L_F^k : \tau^{w^+} \Delta_{\max}^k \leq \Delta \leq \Delta_{\max}^k \right\}$$

where $\tau \in \mathbb{Q} \cap (0; 1)$ is the *frame size adjustment parameter*, $w^+ \in \mathbb{N}$ a fixed integer and $\Delta_{\max}^k$ the *maximum frame size parameter* at iteration k defined as

$$\Delta_{\max}^k = \max_{(x,\Delta) \in L_F^k} \Delta.$$

The mesh size parameter δ^l and frame size parameter Δ^l associated to the lth non-dominated point x^l of L_F^k are linked to each other such that $0 < \delta^l \leq \Delta^l$. When a subsequence of one

of them goes to 0, so does the other. Typically, the following relation $\delta^l = \min\left\{\Delta^l, (\Delta^l)^2\right\}$ meets these requirements.

Each iteration is decomposed into two steps: the *search* and the *poll*. The search is an optional and flexible step which enables the user to design any strategy as long as the proposed trial points belong to the mesh M^k and their number is finite. A common strategy is the use of surrogate models, proposed for example in [53]. The finite set of points used in the search is denoted by S^k.

The poll is more rigorously defined, as the convergence analysis depends on it. The trial points involved in this step, named the poll set and denoted by P^k, must satisfy some specific requirements. More precisely, the construction of P^k involves the use of the current incumbent x^k and its associated frame size Δ^k and mesh size δ^k parameters to obtain a positive spanning set $\mathbb{D}_\Delta^k$. Each column of $\mathbb{D}_\Delta^k$ must be a nonnegative integer combination of the directions in D; the distance from the current incumbent $\mathbf{x}^k$ to a poll point must be bounded by a multiple of the frame size parameter Δ^k. Note that the relation between δ^k and Δ^k given above meets these requirements. Formally, P^k is described as

$$P^k = \{\mathbf{x}^k + \delta^k \mathbf{d} : \mathbf{d} \in \mathbb{D}_\Delta^k\} \subset M^k.$$

All new candidates generated during the search and the poll are assigned a frame size parameter value larger or equal to the frame size parameter of the current feasible frame center.

If a new generated candidate dominates the current feasible incumbent, the iteration is marked as a *success*. Otherwise, it is a *failure* and the frame size parameter (and so the mesh size parameter) of the current feasible frame center is decreased. The iteration can be opportunistic, meaning that as soon as it is successful, the remaining candidates (if they exist) are not evaluated. In all cases, the iterate list L_F^k is filtered to keep only best non-dominated feasible points found until the end of this iteration.

More details can be found in [40] (see Chapter 6).

6.4 Handling of constraints with DMulti-MADS

This section details several strategies to handle constraints with DMulti-MADS. The set of quantifiable and relaxable constraints is given by $\Omega = \{\mathbf{x} \in \mathcal{X} : c_j(\mathbf{x}) \leq 0,\ j \in \mathcal{J}\}$. A relaxable constraint can be violated during the optimization and still returns meaningful outputs for the blackbox. A quantifiable constraint provides a measure of violation of feasibility. All other types of constraints (unrelaxable, hidden, non quantifiable), if present are considered

to be in $\mathcal{X}$[2].

6.4.1 The constraint violation function

Exploiting constraints to guide the algorithm towards optimal solutions requires a way to quantify constraint violations. The strategies described below rely on the *constraint violation function* $h : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ used in [16] and defined by

$$h(\mathbf{x}) = \begin{cases} \sum_{j \in \mathcal{J}} \left(\max\{c_j(\mathbf{x}), 0\}\right)^2 & \text{if } \mathbf{x} \in \mathcal{X}, \\ +\infty & \text{otherwise.} \end{cases}$$

With this definition, $\mathbf{x}$ belongs to Ω if and only if $h(\mathbf{x}) = 0$, and $0 < h(\mathbf{x}) < +\infty$ when $\mathbf{x}$ is infeasible but belongs to $\mathcal{X} \setminus \Omega$. The use of a squared function instead of common ℓ^1 norm enables some conservation of first-order smoothness properties.

6.4.2 The extreme barrier (EB)

Similarly to DMS [74], the original version of the DMulti-MADS algorithm [40] treats constraints via the use of an extreme barrier approach. It replaces the objective function f by

$$f_\Omega(\mathbf{x}) = \begin{cases} (+\infty, +\infty, \ldots, +\infty)^\top & \text{if } \mathbf{x} \notin \Omega, \\ f(\mathbf{x}) & \text{otherwise.} \end{cases}$$

In other terms, all infeasible points are assigned an infinite value. This approach requires a feasible starting point, which is not always available in an engineering context. To allow the use of an infeasible starting point, this work proposes a *Two-phase Extreme Barrier (TEB)* approach, in the continuation of [17]. When starting from an infeasible point, the new strategy, called DMulti-MADS-TEB, performs a single-objective minimization of the h constraint violation function using the MADS algorithm. As soon as a feasible point is found, DMulti-MADS-TEB moves to the second phase, which is the minimization of 6.1 from the feasible point found in the first phase.

Although this approach is simple, its performance has never been investigated in the context of deterministic multiobjective derivative-free optimization. It also shares some convergence properties with MADS and DMulti-MADS, summarized in Section 6.5. Note that this strategy can be applied to any multiobjective blackbox algorithm.

[2]This sentence was modified according to the submitted article.

6.4.3 The progressive barrier (PB)

This subsection introduces the DMulti-MADS-PB extension of the single-objective MADS-PB algorithm [16] for multiobjective derivative-free optimization.

Feasible and infeasible incumbents

Similarly to the MADS-PB algorithm [16], DMulti-MADS-PB constructs two sets of incumbent solutions from V^k. F^k still denotes the set of feasible incumbent solutions. To define the set of infeasible incumbent solutions, one needs to extend the notion of dominance for infeasible solutions, as it is required in the design of filter algorithms [114, 115].

Definition 49 (Dominance relation for constrained multiobjective optimization)**.** In the context of constrained multiobjective optimization, $\mathbf{x}^1 \in \mathcal{X}$ is said to dominate $\mathbf{x}^2 \in \mathcal{X}$ if

- Both points are feasible and $\mathbf{x}^1 \in \Omega$ dominates $\mathbf{x}^2 \in \Omega$, denoted as $\mathbf{x}^1 \prec_f x^2$.
- Both points are infeasible and $f_i(\mathbf{x}^1) \leq f_i(\mathbf{x}^2)$ for $i = 1, 2, \ldots, m$ and $h(\mathbf{x}^1) \leq h(\mathbf{x}^2)$ with at least one strict inequality, denoted as $\mathbf{x}^1 \prec_h \mathbf{x}^2$.

This extension of the dominance relation is different from the definition proposed in [213]. Indeed, in this work, feasible and infeasible points are never compared, and the dominance relation takes into account both objective function values and the constraint violation function values. Another extension of dominance to constrained optimization appears in [118], but as in the previous case, it allows the comparison of feasible and infeasible points. Note that if $m = 1$, the dominance relation reduces into the dominance relation of MADS-PB [16].

With this dominance relation, one can define the set of infeasible nondominated points.

Definition 50. At iteration k, the *set of infeasible nondominated points* is defined as

$$U^k = \left\{ \mathbf{x} \in V^k \setminus \Omega : \text{ there is no } \mathbf{y} \text{ such that } \mathbf{y} \prec_h \mathbf{x} \right\}.$$

As for the MADS-PB algorithm, DMulti-MADS-PB relies on a nonnegative barrier threshold $h^k_{\max}$, set at each iteration k, to construct the set of infeasible incumbent solutions.

Definition 51. At iteration k, the *set of infeasible incumbent solutions* is

$$I^k = \left\{ \arg\min_{\mathbf{x} \in U^k} \left\{ f(\mathbf{x}) : 0 < h(\mathbf{x}) \leq h^k_{\max} \right\} \right\}.$$

All evaluated points having a value of h above $h^k_{\max}$ are automatically rejected by the algorithm. Furthermore, the barrier threshold is nonincreasing with the iteration number k. Its value at each iteration is detailed in Section 6.4.3.

Figure 6.1 illustrates these definitions. Note that I^k is not a singleton. The images of fourteen trial points generated at the beginning of iteration k, i.e., V^k, in the "augmented" objective space (a triobjective space with two objectives f_1, f_2 and the constraint violation function h) for a biobjective minimization optimization problem are represented. The set of feasible incumbent solutions, indicated by black bullets, contains four elements. Two other feasible generated points are equally visible, but each of them is dominated by a feasible incumbent solution. These six generated trial points belong to the biobjective space. The set of infeasible non-dominated points contains six elements, identified by black lozenges and diamonds. Among them, only three qualify to be infeasible incumbent solutions. Indeed, one element among the others is above the threshold value $h^k_{\max}$. The two other ones are dominated by at least one solution of I^k in terms of f objective values. Two elements of U^k dominate the two last remaining trial points, marked by $\times$ symbols. Notice that all elements among I^k and F^k could have been generated before iteration $k-1$, by definition of V^k.

From the sets F^k and I^k, DMulti-MADS constructs two lists of incumbent solutions, the iterate list of best feasible incumbents found until iteration k,

$$L^k_F = \left\{(\mathbf{x}^l, \Delta^l) : \mathbf{x}^l \in F^k \text{ and } \Delta^l > 0,\ l = 1, 2, \ldots, |L^k_F|\right\}$$

and the *iterate list of best infeasible incumbents* found until iteration k

$$L^k_I = \left\{(\mathbf{x}^l, \Delta^l) : \mathbf{x}^l \in I^k \text{ and } \Delta^l > 0,\ l = 1, 2, \ldots, |L^k_I|\right\}.$$

Each element of both lists possesses its own associated frame size parameter Δ^l.

An iteration of the DMulti-MADS-PB algorithm

As for the single-objective optimization MADS-PB algorithm [16], the search and the poll which constitute the two steps of an iteration for DMulti-MADS-PB are organized around two iterate incumbents at iteration k: a feasible one $(\mathbf{x}^k_F, \Delta^k_F) \in L^k_F$ and an infeasible one $(\mathbf{x}^k_I, \Delta^k_I) \in L^k_I$. However, as the frame size parameters associated to the feasible incumbent $\mathbf{x}^k_F$ and infeasible incumbent $\mathbf{x}^k_I$ can be distinct, it is necessary to adapt the definition of the

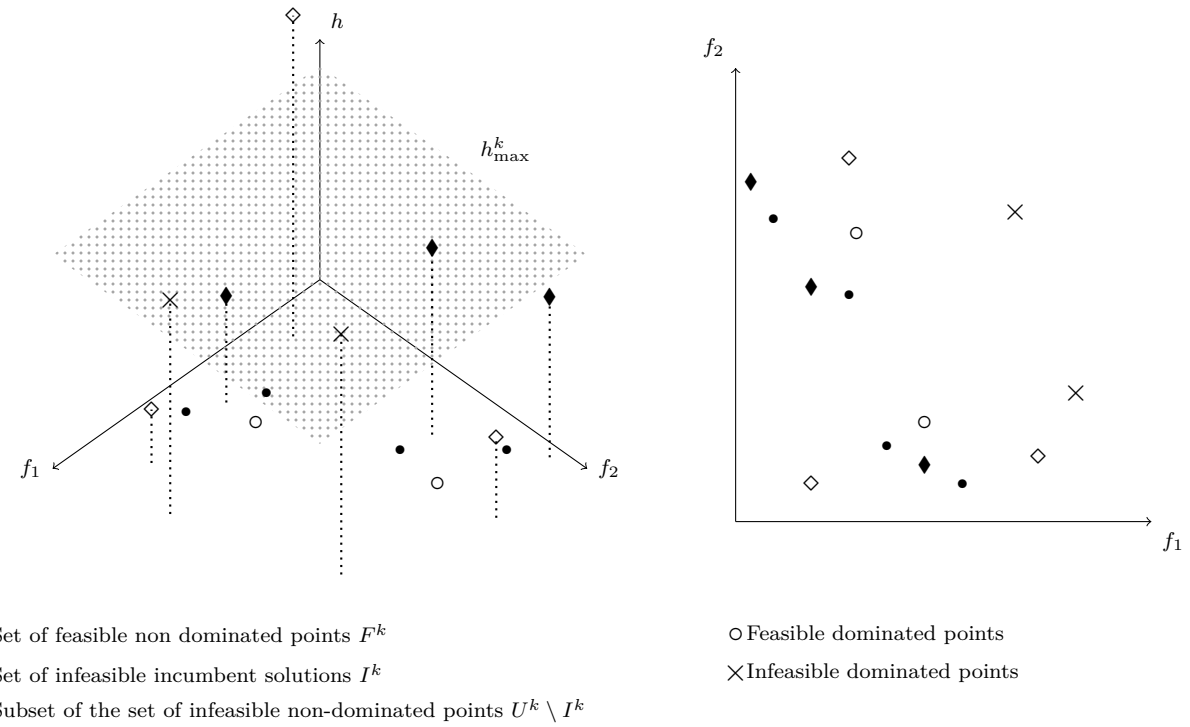


Figure 6.1 An example of feasible and infeasible incumbent solutions at iteration k for a biobjective minimization problem in the "augmented" objective space (a triobjective space with the two objectives f_1, f_2 and the constraint violation function h). On the left, a 3D view; on the right, the projection on the biobjective space.

mesh M^k. At iteration k, M^k is defined as

$$M^k = \begin{cases} \bigcup\limits_{\mathbf{x}\in V^k} \{\mathbf{x} + \delta_F^k \mathbf{D}\mathbf{z} : \mathbf{z} \in \mathbb{N}^{n_D}\} & \text{if } L_F^k \neq \emptyset; \\ \bigcup\limits_{\mathbf{x}\in V^k} \{\mathbf{x} + \delta_I^k \mathbf{D}\mathbf{z} : \mathbf{z} \in \mathbb{N}^{n_D}\} & \text{otherwise,} \end{cases}$$

where $\delta_F^k > 0$ and $\delta_I^k > 0$ are respectively the mesh size parameters associated to the feasible and infeasible incumbents x_F^k and x_I^k defined as $\delta_F^k = \min\left\{\Delta_F^k, (\Delta_F^k)^2\right\}$ and $\delta_I^k = \min\left\{\Delta_I^k, (\Delta_I^k)^2\right\}$. In other terms, the configuration of the mesh M^k at iteration k is primarily based on the selection of the feasible frame center if this last one exists.

It is then possible to adapt the definition of the poll set P^k. At iteration k, P^k is defined as

$$P^k = \begin{cases} P^k(\mathbf{x}_F^k, \Delta_F^k) & \text{for some } (\mathbf{x}_F^k, \Delta_F^k) \in L_F^k \text{ if } L_I^k = \emptyset, \\ P^k(\mathbf{x}_I^k, \Delta_I^k) & \text{for some } (\mathbf{x}_I^k, \Delta_I^k) \in L_I^k \text{ if } L_F^k = \emptyset, \\ P^k(\mathbf{x}_F^k, \Delta_F^k) \cup P^k(\mathbf{x}_I^k, \Delta_F^k) & \text{for some } (\mathbf{x}_F^k, \Delta_F^k) \in L_F^k \text{ and } (\mathbf{x}_I^k, \Delta_I^k) \in L_I^k, \text{ otherwise,} \end{cases}$$

where $P^k(\mathbf{x}, \Delta^k) = \{\mathbf{x} + \delta^k \mathbf{d} : \mathbf{d} \in \mathbb{D}_\Delta^k\} \subset M^k$ represents the poll set centered at x at iteration k with $\delta^k = \min\left\{\Delta^k, (\Delta^k)^2\right\}$.

Figure 6.2 illustrates a construction of the poll set P^k when both the feasible frame center $\mathbf{x}_F^k$ and the infeasible frame center $\mathbf{x}_I^k$ exist. Here, $\Omega \subset \mathbb{R}^2$. All poll candidates belong to one of the frames generated by $\mathbf{x}_F^k$ or $\mathbf{x}_I^k$ of size $\Delta_F^k > 0$ (this is not mandatory as long as the definition of the poll holds). The set P^k is the union of the sets $P^k(\mathbf{x}_F^k, \Delta_F^k) = \{\mathbf{p}^1, \mathbf{p}^2, \mathbf{p}^3, \mathbf{p}^4\}$ and $P^k(\mathbf{x}_I^k, \Delta_I^k) = \{\mathbf{p}^5, \mathbf{p}^6\}$. Section 6.4.3 gives more implementation details on the construction of the poll set.

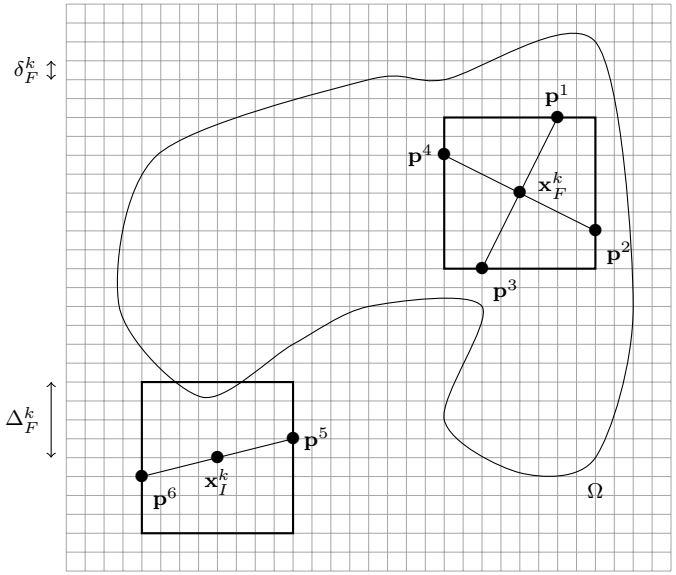


Figure 6.2 Example of a poll set $\mathbf{p}^k = \{\mathbf{p}^1, \mathbf{p}^2, \mathbf{p}^3, \mathbf{p}^4, \mathbf{p}^5, \mathbf{p}^6\}$ for $\Omega \subset \mathbb{R}^2$ when both $\mathbf{x}_F^k$ and $\mathbf{x}_I^k$ exist (inspired by [16]).

It remains to address the choice of the feasible and infeasible frame centers at iteration k. In the case of the MADS-PB algorithm, the set of feasible and infeasible incumbent solutions are often singletons (or composed of points which have the same objective function value and the same h-constrained value). Their selection is then unambiguous.

When L_F^k possesses at least one element, the choice of the current feasible frame center must satisfy the same condition as described in Section 6.3. Practically, to get a good Pareto front approximation, it is also recommended to take into consideration the gap between the different non-dominated solutions found until iteration k, as it is done in [40].

If L_F^k is empty, the infeasible frame center must satisfy

$$(\mathbf{x}_I^k, \Delta_I^k) \in \left\{(\mathbf{x}, \Delta) \in L_I^k : \Delta_{h_{\min}}^k \leq \Delta\right\}$$

where $\Delta^k_{h_{\min}}$ is defined as

$$(\mathbf{x}^k_{h_{\min}}, \Delta^k_{h_{\min}}) \in \arg\min_{(\mathbf{x},\Delta)\in L^k_I} h(\mathbf{x}).$$

The idea behind this selection criterion is to prioritize exploration along the least infeasible point with the best objective values hoping to find a "good" feasible point. At the same time, this selection criterion allows to explore some potentially interesting regions of the objective space. Intuitively, the infeasible current best incumbents associated with a frame size parameter value superior to the least current infeasible point are the ones which have not yet been explored or are promising, due to the update procedure, detailed in Section 6.4.3. Several infeasible incumbents can satisfy this criterion. Then, following the selection procedure described in [40], this work proposes Algorithm 9 to take into account the density of the set of best infeasible incumbents in the objective space.

There remains the case where L^k_F and L^k_I are both non-empty. A first approach would be to independently select the feasible and infeasible frame centers, based for example on a spacing criterion to densify the set of best feasible and best infeasible current solutions. However, this strategy does not exploit the "dominance" order which exists between both sets. More precisely, one could hope that exploring around a carefully chosen infeasible incumbent leads to the generation of a new feasible point which significantly improves the set of current feasible solutions. The proposed approach is inspired by the works of [174].

At iteration k, considering the non-empty iterate list of feasible incumbents L^k_F, this work introduces the function $\psi_{L^k_F} : \mathcal{X} \to \mathbb{R}$ given as

$$\psi_{L^k_F}(\mathbf{x}) = \Phi_{L^k_F}(f(\mathbf{x}))$$
$$= \begin{cases} \min\limits_{(\mathbf{x}^F,\Delta)\in L^k_F} \sum\limits_{i=1}^{m} \left[f_i(\mathbf{x}^F) - \min\left\{ f_i(\mathbf{x}), f_i(\mathbf{x}^F) \right\} \right] & \text{if there is no } (\mathbf{x}^F, \Delta) \in L^k_F \text{ such that} \\ & f_i(\mathbf{x}^F) \leq f_i(\mathbf{x}) \text{ for } i = 1, 2, \ldots, m; \\ -\min\limits_{(\mathbf{x}^F,\Delta)\in L^k_F} \sum\limits_{i=1}^{m} \left[f_i(\mathbf{x}) - \min\left\{ f_i(\mathbf{x}), f_i(x^F) \right\} \right] & \text{otherwise.} \end{cases}$$

The level sets of $\Phi_{L^k_F}$ are depicted in Figure 6.4. Note that all potential feasible decision vectors which are not dominated by a current feasible incumbent solution of L^k_F are given a positive $\psi_{L^k_F}$ value. All dominated feasible decision vectors correspond to a negative $\psi_{L^k_F}$ value.

The current infeasible frame center is then chosen as the element of L^k_I which maximizes the $\psi_{L^k_F}$ function, i.e.,

$$(\mathbf{x}^k_I, \Delta^k_I) \in \arg\max_{(\mathbf{x},\Delta)\in L^k_I} \psi_{L^k_F}(\mathbf{x}).$$

Algorithm 9 `selectCurrentInfeasibleIncumbent`(L_I^k)

Let $L_I^{select} := \left\{(\mathbf{x}, \Delta) \in L_I^k : \Delta_{h_{\min}}^k \leq \Delta\right\}$ with $\Delta_{h_{\min}}^k = \arg \min_{(\mathbf{x},\Delta)\in L_I^k} h(\mathbf{x})$.
if $|L_I^{select}| = 1$ **then**
  return $(\mathbf{x}, \Delta)$ with $L_I^{select} = \{(\mathbf{x}, \Delta)\}$.
else if $|L_I^{select}| = 2$ and $|L_I^k| = 2$ **then**
  Let $l_0 \in \arg\max_{l=1,2} \max_{i=1,2,\ldots,m} f_i(\mathbf{x}^l)$.
  return $(\mathbf{x}^{l_0}, \Delta^{l_0})$.
else
  for $i = 1, 2, \ldots, m$ **do**
    Order $L_I^k = \left\{(\mathbf{x}^1, \Delta^1), (\mathbf{x}^2, \Delta^2), \ldots, (\mathbf{x}^{|L_I^k|}, \Delta^{|L_I^k|})\right\}$ such that
    $f_i(\mathbf{x}^1) \leq f_i(\mathbf{x}^2) \leq \ldots \leq f_i(\mathbf{x}^{|L^k|})$.
    for $l = 1, 2, \ldots, |L_I^k|$ **do**
      Compute $\gamma_i(\mathbf{x}^l)$ defined as

$$\gamma_i(\mathbf{x}^l) = \begin{cases} 2\dfrac{f_i(\mathbf{x}^2) - f_i(\mathbf{x}^1)}{f_i(\mathbf{x}^{|L_I^k|}) - f_i(\mathbf{x}^1)} & \text{if } l = 1, \\ 2\dfrac{f_i(\mathbf{x}^{|L_I^k|}) - f_i(\mathbf{x}^{|L_I^k|-1})}{f_i(\mathbf{x}^{|L_I^k|}) - f_i(\mathbf{x}^1)} & \text{if } l = |L_I^k|, \\ \dfrac{f_i(\mathbf{x}^{l+1}) - f_i(\mathbf{x}^{l-1})}{f_i(\mathbf{x}^{|L_I^k|}) - f_i(\mathbf{x}^1)} & \text{otherwise.} \end{cases}$$

    end for
  end for
  Let $l_0 \in \arg\max_{l=1,2,\ldots,|L_I^{select}|} \max_{i=1,2,\ldots,m} \gamma_i(\mathbf{x}^l)$.
  return $(\mathbf{x}^{l_0}, \Delta^{l_0})$.
end if

Figure 6.3 A procedure to select the current incumbent at iteration k taking into account the spacing between elements of the iterate list of best infeasible incumbents L_I^k in the objective space, inspired by [40].

Intuitively, exploring around an infeasible frame center with a large positive value can lead to the generation of a feasible point which significantly improves the current Pareto front approximation. If the selected infeasible frame center possesses a negative value, one could expect it to be potentially "close" to the non-dominated zone relative to the current Pareto front approximation. An exploration around it can still improve the current feasible set[3].

[3]This sentence was modified according to the submitted article

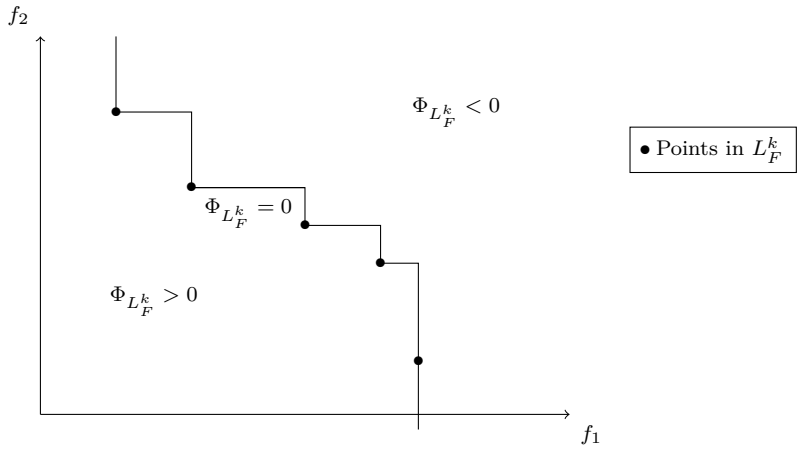


Figure 6.4 Level sets in the objective space of $\Phi_{L_F^k}$ for a biobjective minimization problem.

Update of the mesh parameter at the end of an iteration

At the end of the search and the poll at iteration k, DMulti-MADS-PB has evaluated a finite number of candidates on the mesh M^k. The cache V^{k+1} is then the union of the cache V^k at the beginning of iteration k and all the candidates evaluated during iteration k. As for MADS-PB [16], the values of f and h stored in V^{k+1} for DMulti-MADS-PB determine the way the threshold value $h_{\max}^{k+1}$ (see (6.1)) and the mesh and frame size parameters of the elements of iterate lists of feasible and infeasible incumbents L_F^{k+1} and L_I^{k+1} are updated.

Similarly to MADS-PB [16], this work uses the concept of *dominating*, *improving* and *unsuccessful* iteration. A dominating iteration occurs when DMulti-MADS-PB generates a trial point which dominates a current frame incumbent. An improving iteration is not dominating but improves the feasibility of the infeasible frame center. Otherwise, the iteration is unsuccessful. More precisely,

- Iteration k is said to be dominating whenever a trial point $\mathbf{x}^t \in V^{k+1}$ dominates one frame incumbent, i.e.,

  $$h(\mathbf{x}^t) = 0 \text{ and } \mathbf{x}^t \prec_f \mathbf{x}_F^k \text{ or } h(\mathbf{x}^t) > 0 \text{ and } \mathbf{x}^t \prec_h \mathbf{x}_I^k$$

  is found.

- Iteration k is said to be improving if it is not dominating, but generates a trial point $\mathbf{x}^t \in V^{k+1}$ which satisfies

  $$0 < h(\mathbf{x}^t) < h(\mathbf{x}_I^k) \text{ and there exists } i_0 \in \{1, 2, \ldots, m\} \text{ such that } f_{i_0}(\mathbf{x}_I^k) < f_{i_0}(\mathbf{x}^t).$$

- Iterations which are neither dominating nor improving are labelled as unsuccessful. It happens when every trial point $\mathbf{x}^t \in V^{k+1}$ is such that

$$h(\mathbf{x}^t) = 0 \text{ and } \mathbf{x}^t \not\prec_f \mathbf{x}_F^k, \text{ or } h(\mathbf{x}^t) = h(\mathbf{x}_I^k) \text{ and } \mathbf{x}^t \not\prec_h \mathbf{x}_I^k \text{ or } h(\mathbf{x}^t) > h(\mathbf{x}_I^k).$$

These three cases are described in Figure 6.5.

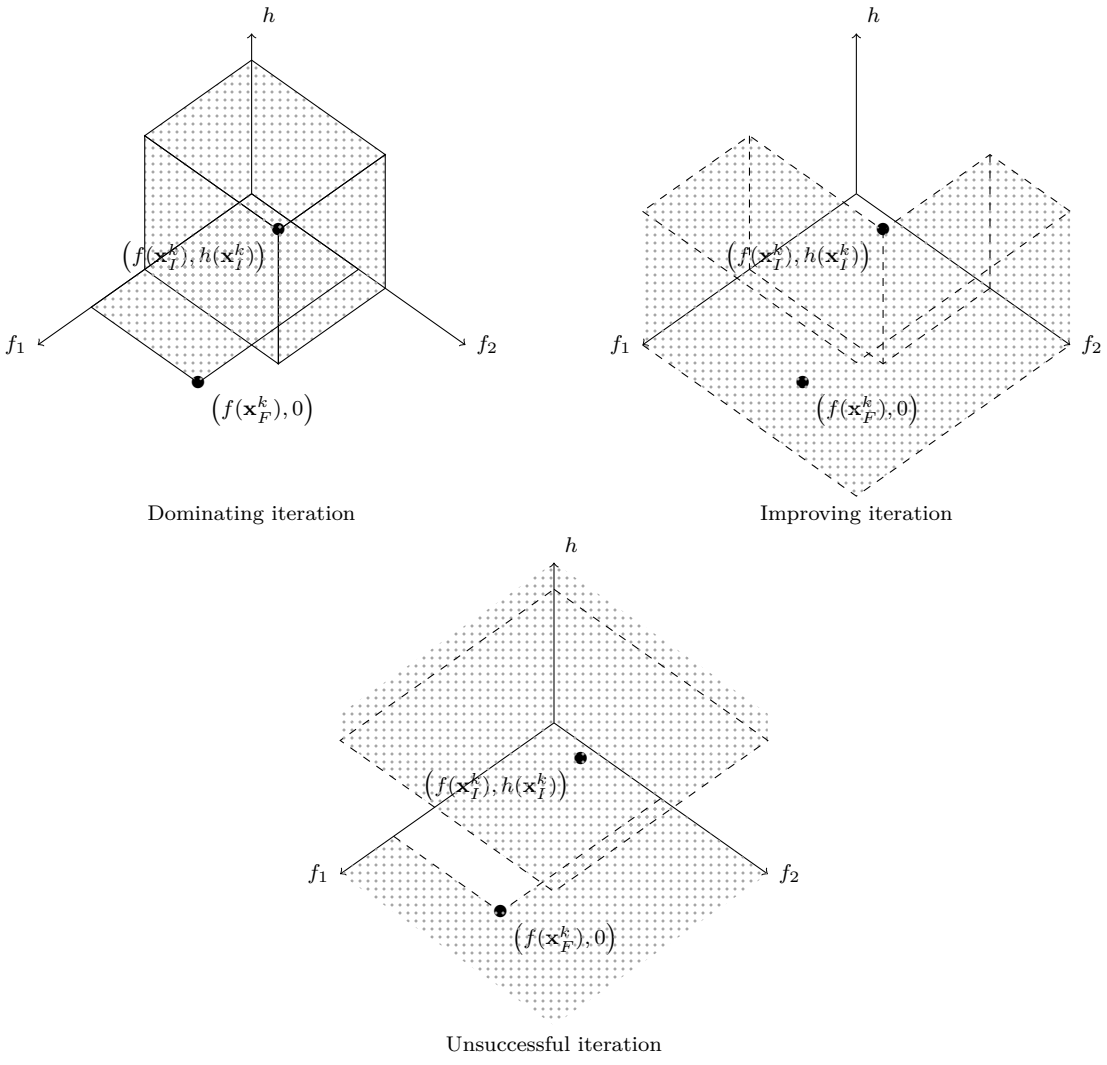


Figure 6.5 Iterations cases for DMulti-MADS-PB.

All points generated during iteration k are given a frame size parameter $\Delta \geq \Delta^k$ where Δ^k is the frame size parameter associated to M^k. More precisely, for any trial element $(\mathbf{x}^t, \Delta)$

generated during iteration k,

$$(\mathbf{x}^t, \Delta) = \begin{cases} (\mathbf{x}^t, \tau^{-1}\Delta^k) & \text{if } h(\mathbf{x}^t) = 0 \text{ and there exists at least } \mathbf{x} \in F^k \text{ such that } \mathbf{x}^t \prec_f \mathbf{x}, \text{ or} \\ (\mathbf{x}^t, \tau^{-1}\Delta^k) & \text{if } h(\mathbf{x}^t) = 0 \text{ and there exists at least an index } i_0 \in \{1, 2, \ldots, m\} \text{ such} \\ & \text{that } f_{i_0}(\mathbf{x}^t) < \min_{\mathbf{x} \in F^k} f_{i_0}(\mathbf{x}), \text{ or} \\ (\mathbf{x}^t, \tau^{-1}\Delta^k) & \text{if } h(\mathbf{x}^t) > 0 \text{ and there exists at least } \mathbf{x} \in I^k \text{ such that } \mathbf{x}^t \prec_h \mathbf{x}, \text{ or} \\ (\mathbf{x}^t, \tau^{-1}\Delta^k) & \text{if } 0 < h(\mathbf{x}^t) \leq \max_{\mathbf{x} \in I^k} h(\mathbf{x}) \text{ and for } i = 1, 2, \ldots, m, \\ & f_i(\mathbf{x}^t) \leq \min_{\mathbf{x} \in I^k} f_i(\mathbf{x}) \text{ with at least one index } i_0 \in \{1, 2, \ldots, m\} \\ & \text{such that } f_{i_0}(\mathbf{x}^t) < \min_{\mathbf{x} \in I^k} f_{i_0}(\mathbf{x}), \\ (\mathbf{x}^t, \Delta^k) & \text{otherwise;} \end{cases}$$

where $\tau \in (0, 1) \cap \mathbb{Q}$ is the frame size adjustment parameter chosen by the user. Thus, all candidates which dominate one of the points in L_F^k or L_I^k or improve the extent of the objectives values covered by at least one of the iterate list have their associated frame size parameter increased. When L_F^k is empty and no feasible point has been generated at iteration k, these candidates are likely to be potential frame center candidates at iteration k+1. If L_F^k is not empty, the update of the frame size parameter associated to a new feasible generated point is similar to the one proposed in the original DMulti-MADS-EB algorithm [40]. If the iteration is labelled as unsuccessful, no generated point at the end of iteration k dominates at least one of the frame center incumbents. In this case, DMulti-MADS-PB replaces $(\mathbf{x}_{center}^k, \Delta_{center}^k)$ by $(\mathbf{x}_{center}^k, \tau\Delta^k)$ with $\tau \in (0; 1) \cap \mathbb{Q}$ and $\mathbf{x}_{center}^k \in \{\{\mathbf{x}_F^k\}, \{\mathbf{x}_I^k\}, \{\mathbf{x}_F^k, \mathbf{x}_I^k\}\}$ relatively to the emptiness of the iterate lists L_F^k or L_I^k. If the iteration is improving, the frame size parameters associated to the existing frame center incumbents keep the same value as in iteration k.

Figure 6.6 illustrates the frame update rules for a biobjective minimization problem in the "augmented" objective space (a triobjective space with two objectives f_1, f_2 and the constraint violation function h). All candidates whose image is outside combined gray areas are affected a frame size parameter $\Delta := \Delta^k$.

The threshold barrier is then updated according to the following rules:

$$h_{\max}^{k+1} := \begin{cases} \max\limits_{\mathbf{x}^t \in V^{k+1}} \left\{ h(\mathbf{x}^t) : h(\mathbf{x}^t) < h(\mathbf{x}_I^k) \right\} & \text{if iteration } k \text{ is improving,} \\ h(\mathbf{x}_I^k) & \text{if } h(\mathbf{x}_I^k) = \max\limits_{\mathbf{x} \in I^k} h(\mathbf{x}), \\ \max\limits_{\mathbf{x}^t \in V^{k+1}} \left\{ h(\mathbf{x}^t) : h(\mathbf{x}_I^k) \leq h(\mathbf{x}^t) < \max\limits_{\mathbf{x} \in I^k} h(\mathbf{x}) \right\} & \text{otherwise.} \end{cases} \tag{6.1}$$

The threshold update rule guarantees in the case where an iteration is considered as not

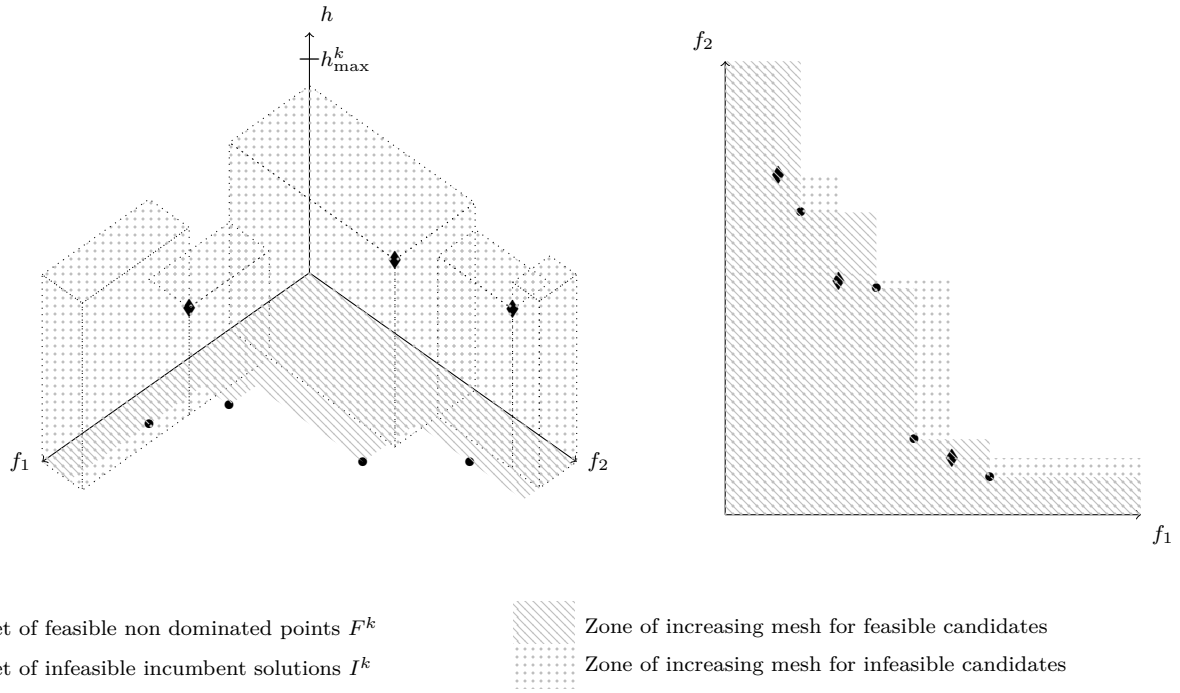


Figure 6.6 An example of an increasing zone for frame size parameters at iteration k for a biobjective minimization problem. On the left, a 3D view of the "augmented" objective space (a triobjective space with the two objectives f_1, f_2 and the constraint violation function h); on the right, projection on the biobjective space.

improving that the set I^k will change if the infeasible frame incumbent does not possess the maximum violation function value h among the elements of I^k at iteration k. Another consequence (similar to MADS-PB [16]) is that $h^k_{\max}$ is nonincreasing with iteration k and that if $I^k \neq \emptyset$, $I^q \neq \emptyset$ for all iteration indexes $q \geq k$.

Note that even if an iteration is marked as unsuccessful, the algorithm can still generate new feasible non-dominated points or infeasible non-dominated points below the value $\max_{\mathbf{x} \in I^k} h(\mathbf{x})$, which may be used as frame incumbents in some next iteration.

Remark. It is also possible to set the update rules of the threshold $h^{k+1}_{\max}$ according to the $h(\mathbf{x}^k_I)$ barrier value. Nonetheless, in some preliminary experiments, it has been observed that this approach prevents the algorithm to explore some parts of the objective space, potentially interesting to greatly improve the current feasible solution set.

Finally, the iterate lists L^k_F and L^k_I are filtered to add new non-dominated points generated during iteration k and remove potential resulting dominated elements.

Algorithm 10 summarizes the different steps of the DMulti-MADS-PB algorithm.

Remark. When $m = 1$, the classification of the different type of iterations used in the DMulti-

Algorithm 10 The DMulti-MADS-PB algorithm for constrained optimization

Initialization : Given a finite set of points $V^0 \subset \mathcal{X}$, choose $\Delta^0 > 0$, $\mathbf{D} = \mathbf{GZ}$ a positive spanning set matrix, $\tau \in (0;1) \cap \mathbb{Q}$ the frame size adjustment parameter, and $w^+ \in \mathbb{N}$ a fixed integer parameter. Define the frame trigger parameter $\rho > 0$ (optional). Initialize the lists $L_F^0 = \left\{(\mathbf{x}_F^l, \Delta^0),\ l = 1, 2, \ldots, |L_F^0|\right\}$ and $L_I^0 = \left\{(\mathbf{x}_I^l, \Delta^0),\ l = 1, 2, \ldots, |L_I^0|\right\}$ for some $(\mathbf{x}_F^l, \mathbf{x}_I^l) \in V^0$.

for $k = 0, 1, 2, \ldots$ **do**

Selection of the current infeasible frame centers. Select feasible and/or infeasible elements of respective iterate lists L_F^k and L_I^k as described in [40] and Algorithm 9. Define the current frame size parameter Δ^k according to the associated frame size parameters of the feasible incumbent element $(\mathbf{x}_F^k, \Delta_F^k)$ and/or infeasible current incumbent element $(\mathbf{x}_I^k, \Delta_I^k)$. Set $\delta^k = \min\left\{\Delta^k, \left(\Delta^k\right)^2\right\}$. Initialize $L^{add} := \emptyset$.

Search (optional) : Evaluate f and h at a finite set of points $S^k \subset \mathcal{X}$ on the mesh $M^k = \bigcup_{\mathbf{x} \in V^k} \{\mathbf{x} + \delta^k \mathbf{Dz} : \mathbf{z} \in \mathbb{N}^{n_D}\}$. Set $L^{add} := \{(\mathbf{x}, \Delta^k) : \mathbf{x} \in S^k\}$.

If an improving or dominating success criterion is satisfied, the search may terminate. In this case, skip the poll and go to the parameter update step.

Poll : Select a positive spanning set $\mathbb{D}_\Delta^k$. Evaluate f and h on the poll set $P^k \subset M^k$ as defined in Subsection 6.4.3. Set $L^{add} := L^{add} \cup \left\{(\mathbf{x}, \Delta^k) : \mathbf{x} \in P^k\right\}$. If an improving or dominating criterion is satisfied, the poll may terminate opportunistically.

Parameter update : Define V^{k+1} as the union of V^k and all new candidates evaluated in $\mathcal{X}$ during the search and the poll. Classify the iteration as dominating, improving or unsuccessful. Update $h_{\max}^{k+1}$ according to Section 6.4.3. Remove points above the threshold from L_I^k. Update the iterate lists L_F^{k+1} and/or L_I^{k+1} by adding new non-dominated points from L^{add} with their updated associated frame center $\Delta \in \{\Delta^k, \tau^{-1}\Delta^k\}$, as explained in Section 6.4.3. Remove new dominated points from L_F^k and/or L_I^k.

If the iteration is unsuccessful, replace (if they exist) the frame center elements $(\mathbf{x}_F^k, \Delta_F^k)$ and $(\mathbf{x}_I^k, \Delta_I^k)$ respectively by $(\mathbf{x}_F^k, \Delta^{k+1})$, $(\mathbf{x}_I^k, \Delta^{k+1})$ with $\Delta^{k+1} := \tau\Delta^k$.

end for

Figure 6.7 A summary of the DMulti-MADS-PB algorithm, inspired by [40].

MADS-PB context is equivalent to the one used for the MADS-PB algorithm [16]. There also exists many configurations of iteration classifications criteria such that the generalization of the MADS-PB algorithm for multiobjective optimization and the convergence properties still hold. For example, one can declare an iteration as dominating when a trial point changes the set I^k. Practically, not all of them have the same performance. The definitions used below correspond to the most efficient variant observed on some preliminary experiments.

A frame center selection rule for the DMulti-MADS-PB algorithm

The constrained single-objective MADS-PB algorithm uses a classification of its two frame centers to practically improve the performance of the poll step. More precisely, the two frame centers are ordered, based on their objective values into *primary* and *secondary* poll centers. MADS-PB concentrates more efforts (based on the number of poll directions) on the primary poll center than the secondary poll center [16]. Inspired by this strategy, this subsection proposes an extension of the so-called *frame center selection rule* to constrained multiobjective optimization.

As in the single-objective case, DMulti-MADS-PB executes the poll around at least one frame center. When $L_F^k = \emptyset$ or $L_I^k = \emptyset$, there is only one frame center, designed as the *primary frame center*. A complete set of poll points can be evaluated based on a positive spanning set $\mathbb{D}_\Delta^k$ composed of at least of $n+1$ directions (more details for the construction of $\mathbb{D}_\Delta^k$ can be found in [2, 22]).

When L_F^k and L_I^k are both non-empty, polling is done around a feasible and an infeasible frame centers. DMulti-MADS-PB orders these two frame centers into a *primary* frame center and a *secondary* frame center. This ordering is based on an user-supplied parameter $\rho > 0$, called the *frame trigger parameter*.

Recall that if L_F^k and L_I^k are nonempty, the selection of the infeasible frame center is done based on the $\psi_{L_F^k} : \mathcal{X} \to \mathbb{R}$ function parametrized by L_F^k, defined in Section 6.4.3. The following frame center selection rule is then proposed.

Definition 52 (frame center selection rule)**.** Let $\rho > 0$ provided by the user and suppose that $L_F^k \neq \emptyset$ and $L_I^k \neq \emptyset$. Let $(\mathbf{x}_F^k, \Delta_F^k) \in L_F^k$ be the feasible current incumbent and $(\mathbf{x}_I^k, \Delta_I^k) \in L_I^k$ be the infeasible current incumbent. If $\psi_{L_F^k}(\mathbf{x}_I^k) - \rho\, \xi(L_F^k) > 0$, where $\xi(L_F^k)$ is given by

$$\xi(L_F^k) = \sum_{i=1}^{m} \mu \left(\max_{(\mathbf{x},\Delta)\in L_F^k} f_i(\mathbf{x}), \min_{(\mathbf{x},\Delta)\in L_F^k} f_i(\mathbf{x}) \right)$$

with $\mu : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^+$ defined as

$$\mu(a,b) = \begin{cases} |a-b| & \text{if } a \neq b, \\ |a| & \text{otherwise;} \end{cases}$$

then the primary poll center is chosen as $\mathbf{x}_I^k$ and the secondary poll center is chosen as $\mathbf{x}_F^k$, otherwise the primary poll center is chosen as $\mathbf{x}_F^k$ and the secondary poll center is chosen as $\mathbf{x}_I^k$.

As for the single-objective MADS-PB algorithm, DMulti-MADS-PB puts more effort on the primary frame center than on the secondary frame center. The implementation of the poll strategy in this work follows the one developed in [22]: $n+1$ directions are used for the primary frame center and 2 directions for the secondary frame center by taking the negative of the first one.

If there exists at least one element $(\mathbf{x}, \Delta) \in L_F^k$ such that $f_i(\mathbf{x}) \leq f_i(\mathbf{x}_I^k)$ for $i = 1, 2, \ldots, m$, then $\mathbf{x}_F^k$ will be chosen as the primary poll center. Figure 6.8 illustrates the zone in the biobjective space where I^k elements must be to be considered as potential primary poll centers. One could hope that putting more effort on the infeasible frame center in this case should enable it to reach a better part of the feasible decision region [16].

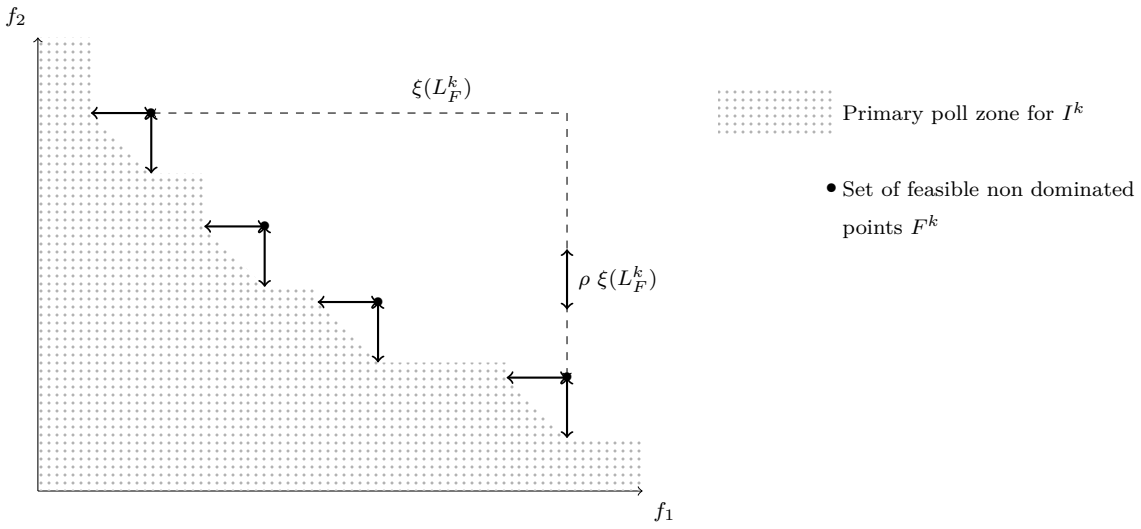


Figure 6.8 Representation of the selection of I^k frame incumbent as primary poll in the objective space for a biobjective minimization problem.

Remark. If $m = 1$, the frame center selection rule is equivalent to $f(\mathbf{x}_I^k) < f(\mathbf{x}_F^k) - \rho|f(\mathbf{x}_F^k)|$, used in [12]. In this work, this rule was privileged to the original one in [12] $f(\mathbf{x}_F^k) - f(\mathbf{x}_I^k) > \rho$, as it takes into account the scale of the objective function values. A corresponding frame center selection rule extension to multiobjective optimisation would have been $\psi_{L_F^k}(\mathbf{x}_I^k) - m\rho > 0$.

6.5 Convergence analysis

This section is devoted to the convergence analysis of the DMulti-MADS-TEB and DMulti-MADS-PB algorithms, inspired by [16,40]. This work makes use of the following assumptions, taken from [16]:

Assumption 6.5.1. *There exists some point* $\mathbf{x}^0$ *in the user-provided set* V^0 *such that* $\mathbf{x}^0 \in \mathcal{X}$ *and* $f(\mathbf{x}^0)$ *and* $h(\mathbf{x}^0)$ *are both finite.*

Assumption 6.5.2. *All iterates considered by the algorithm lie in a bounded set.*

If Assumption 6.5.1 is not satisfied, DMulti-MADS cannot start. Assumption 6.5.2 is ensured if one imposes the existence of a bounded set in $\mathbb{R}^n$ containing V^k for all $k \in \mathbb{N}$. As V^k for $k \in \mathbb{N}$ is always composed of points satisfying the unrelaxable constraints, it is sufficient to guarantee that the set of unrelaxable constraints is itself a bounded set. For example, many engineering problems possess bound variables constraints, which cannot be violated.

As in single-objective MADS-PB algorithm [16], combining assumptions 6.5.1 and 6.5.2 and the structure of the mesh M^k enables to show that $\lim_{k\to+\infty} \inf \Delta^k = \lim_{k\to+\infty} \inf \delta^k = 0$ (see for example [74, Theorem A.1]). The classical convergence analysis of direct search methods focuses on subsequences of generated frame centers for which corresponding mesh size and frame size parameters converge to zero. The following notations and definitions are adapted from [16].

Let $\mathbb{U} \subset \mathbb{N}$ the set of unsuccessful iterations indexes. The poll generates one or several trial points around at least one of the two feasible and infeasible incumbents. If $k \in \mathbb{U}$ and the poll is executed around the feasible current frame center $\mathbf{x}_F^k$, this last one is designed as a *feasible minimal frame center*. Otherwise, if $k \in \mathbb{U}$ and the poll is executed around the infeasible current frame center $\mathbf{x}_I^k$, this last one is designed as an *infeasible minimal frame center*. From the rest of this work, these subsequences of frame centers are investigated separately. Note that for the DMulti-MADS-TEB variant, studying a subsequence of infeasible minimal frame centers means that the algorithm does not manage to find a feasible point.

Definition 53. A subsequence $\{\mathbf{x}^k\}_{k\in K}$ of DMulti-MADS frame centers, for some infinite subset of indexes $K \subseteq \mathbb{U}$ is said to be a *refining subsequence* if $\{\Delta^k\}_{k\in K}$ converges to 0. The limit point $\hat{\mathbf{x}}$ of a convergent refining subsequence is called a *refining point*.

Definition 54. Given a corresponding refining subsequence $\{\mathbf{x}^k\}_{k\in K}$ and its refining point $\hat{\mathbf{x}}$, a direction $\mathbf{d}$ is said to be a *refining direction* if and only if there exists an infinite subset of indexes $K' \subseteq K$ such that $\mathbf{d}^k \in \mathbb{D}_\Delta^k$ with $\mathbf{x}^k + \delta^k \mathbf{d}^k \in \mathcal{X}$ and $\mathbf{d} = \lim_{k\in K'} \frac{\mathbf{d}^k}{\|\mathbf{d}^k\|}$.

The convergence analysis also requires some mathematical tools from nonsmooth analysis. The following definitions are taken from [16].

Definition 55. A vector $\mathbf{d} \in \mathbb{R}^n$ is said to be a Clarke tangent vector to the set $\Omega \subseteq \mathbb{R}^n$ at the point $\mathbf{x}$ in the closure of Ω if for every sequence $\{\mathbf{y}^k\}$ of elements of Ω that converge

to $\mathbf{x}$ and for every sequence of positive real numbers $\{t^k\}$ converging to zero, there exists a sequence of vectors $\{\mathbf{w}^k\}$ converging to $\mathbf{d}$ such that $\mathbf{y}^k + t^k\mathbf{w}^k \in \Omega$.

The set of all Clarke tangent vectors to Ω at $\mathbf{x}$ is the Clarke tangent cone to Ω at $\mathbf{x}$ denoted as $\mathcal{T}_\Omega^{Cl}(\mathbf{x})$. The DMulti-MADS analysis in a general constrained optimization context makes use of the hypertangent cone [219], which is the interior of the Clarke tangent cone, defined as:

$$\mathcal{T}_\Omega^{H}(\mathbf{x}) = \{\mathbf{d} \in \mathbb{R}^n : \exists\ \epsilon > 0 \text{ such that } \mathbf{y} + t\mathbf{w} \in \Omega, \text{ for all } \mathbf{y} \in \Omega \cap \mathcal{B}_\epsilon(\mathbf{x}), \mathbf{w} \in \mathcal{B}_\epsilon(\mathbf{d}), \text{ and } 0 < t < \epsilon\}$$

where $\mathcal{B}_\epsilon(\mathbf{x})$ is the open ball of radius $\epsilon > 0$ centered at $\mathbf{x}$.

The DMulti-MADS analysis also requires that the objective function f is locally Lipschitz continuous in $\mathcal{X}$, i.e., each of its components f_i for $i = 1, 2, \ldots, m$ is locally Lipschitz continuous in $\mathcal{X}$. If this condition is satisfied, the Clarke-Jahn generalized derivatives [59] of f_i at $\mathbf{x} \in \mathcal{X}$ in the direction $\mathbf{d} \in \mathbb{R}^n$ exist and are defined by

$$f_i^o(\mathbf{x}; \mathbf{d}) = \limsup_{\substack{\mathbf{y} \to \mathbf{x},\ \mathbf{y} \in \mathcal{X} \\ t \searrow 0,\ \mathbf{y} + t\mathbf{d} \in \mathcal{X}}} \frac{f(\mathbf{y} + t\mathbf{d}) - f(\mathbf{y})}{t}, \text{ for } i = 1, 2, \ldots, m.$$

This work can then introduce the main stationary conditions.

Definition 56. Let f be Lipschitz continuous near a point $\hat{\mathbf{x}} \in \Omega$. $\hat{\mathbf{x}}$ is a Pareto-Clarke critical point of f in Ω if for all directions $\mathbf{d} \in \mathcal{T}_\Omega^{Cl}(\hat{\mathbf{x}})$, there exists $i(\mathbf{d}) \in \{1, 2, \ldots, m\}$ such that $f_{i(\mathbf{d})}^o(\hat{\mathbf{x}}; \mathbf{d}) \geq 0$.

With the additional assumption that f is equally strictly differentiable at $\hat{\mathbf{x}}$ (i.e. the corresponding Clarke generalized is a singleton containing only the gradient of one objective component at $\hat{\mathbf{x}}$), the previous definition can be reformulated.

Definition 57. Let f be strictly differentiable at a point $\hat{\mathbf{x}} \in \Omega$. $\hat{\mathbf{x}}$ is a Pareto-Clarke-KKT critical point of f in Ω if for all directions $\mathbf{d} \in \mathcal{T}_\Omega^{Cl}(\hat{\mathbf{x}})$, there exists $i(\mathbf{d}) \in \{1, 2, \ldots, m\}$ such that $\nabla f_{i(\mathbf{d})}(\hat{\mathbf{x}})^\top \mathbf{d} \geq 0$.

As in the single-objective case [16], this work divides the convergence analysis into two cases: the study of subsequences of feasible minimal frame centers and the study of subsequences of infeasible minimal frame centers. For each case, the following methodology is used:

1. Prove that a subsequence of mesh size parameters and frame size parameters converges to zero.

2. Determine a particular sequence of iterate points associated to the previous subsequence of parameters, i.e. a so-called *refined subsequence*.

3. This sequence of iterate points converges to a refined point. Prove that this point satisfies some stationary properties.

6.5.1 Feasible case: results for f

As in [40], one wants to show that starting from a set of feasible points, DMulti-MADS produces at the limit locally stationary points for the constrained multiobjective optimization problem. To do that, this work proves the existence of finer refining subsequences, as it is done in [40]. The following analysis is a summary of the convergence analysis developed in [40] and covers the two variants DMulti-MADS-TEB and DMulti-MADS-PB.

Theorem 12. *Let Assumptions 6.5.1 and 6.5.2 hold and suppose DMulti-MADS generates a sequence of feasible iterates lists* $\{L_F^k\}$ *with* $L_F^k = \{(\mathbf{x}^j, \Delta^j), j = 1, 2 \ldots, |L_F^k|\}$. *Then*

$$\lim_{k \to +\infty} \inf \delta_{\max}^k = \lim_{k \to +\infty} \inf \Delta_{\max}^k = 0.$$

Proof. Combining assumptions 6.5.1, 6.5.2, and the selection criterion of the feasible frame center with the structure of the mesh has been shown to be enough to ensure $\lim_{k \to +\infty} \inf \delta_{\max}^k = \lim_{k \to +\infty} \inf \Delta_{\max}^k = 0$ (see [40, Theorem 5.1] for more details). □

This work wants to prove the convergence of specific elements of the feasible iterate list generated by DMulti-MADS to stationary points. The concept of a feasible linked list, adapted from [40, 180], is then introduced.

Definition 58. Suppose DMulti-MADS generates the sequence of feasible iterate lists $\{L_F^k\}_{k \geq k_0}$ with $L_F^k = \{(\mathbf{x}^l, \Delta^l), l = 1, 2, \ldots, |L_F^k|\}$ and $k_0 \in \mathbb{N}$ the iteration index such that $k_0 \in \arg\min \left\{k \in \mathbb{N} : F^k \neq \emptyset\right\}$. A *feasible linked sequence* is defined as a sequence $\{(\mathbf{x}^{l_k}, \Delta^{l_k})\}$ such that there exists an iteration index $\ell_0 \geq k_0$ such that for any $k = \ell_0 + 1, \ell_0 + 2, \ldots$, the pair $\left\{(\mathbf{x}^{l_k}, \Delta^{l_k})\right\} \in L_F^k$ is generated at iteration $k - 1$ of DMulti-MADS from the pair $(\mathbf{x}^{l_{k-1}}, \Delta^{l_{k-1}}) \in L_F^{k-1}$.

For the DMulti-MADS-PB variant algorithm, the following cases can occur:

1. Dominating iteration: either the algorithm generates at least one point which dominates the feasible frame center $\mathbf{x}_F^{k-1}$, or it generates some infeasible points which have triggered the dominating success condition in the infeasible case.

- $\forall(\mathbf{x}^{l_k}, \Delta^{l_k}) \in L_F^k \setminus L_F^{k-1}$,

$$\mathbf{x}^{l_k} = \mathbf{x}^{k-1} + \delta^{k-1}\mathbf{D}\mathbf{z}^{k-1} \text{ for some } \mathbf{z}^{k-1} \in \mathbb{N}^{n_D} \text{ and } \Delta^{l_k} \in \{\Delta^{k-1}, \tau^{-1}\Delta^{k-1}\}$$

  with $\mathbf{x}^{k-1} \in \{\mathbf{x}_F^{k-1}, \mathbf{x}_I^{k-1}\}$.
- $\forall(\mathbf{x}^{l_k}, \Delta^{l_k}) \in L_F^k \cap L_F^{k-1}$,

$$\mathbf{x}^{l_k} = \mathbf{x}^{l_{k-1}} \text{ and } \Delta^{l_k} = \Delta^{l_{k-1}}.$$

2. Improving iteration: the algorithm may generate some new feasible non-dominated points without dominating the feasible frame incumbent.

- $\forall(\mathbf{x}^{l_k}, \Delta^{l_k}) \in L_F^k \setminus L_F^{k-1}$,

$$\mathbf{x}^{l_k} = \mathbf{x}^{k-1} + \delta^{k-1}\mathbf{D}\mathbf{z}^{k-1} \text{ for some } \mathbf{z}^{k-1} \in \mathbb{N}^{n_D} \text{ and } \Delta^{l_k} \in \{\Delta^{k-1}, \tau^{-1}\Delta^{k-1}\}$$

  with $\mathbf{x}^{k-1} \in \{\mathbf{x}_F^{k-1}, \mathbf{x}_I^{k-1}\}$.
- $\forall(\mathbf{x}^{l_k}, \Delta^{l_k}) \in L_F^k \cap L_F^{k-1}$,

$$\mathbf{x}^{l_k} = \mathbf{x}^{l_{k-1}} \text{ and } \Delta^{l_k} = \Delta^{l_{k-1}}.$$

3. Unsuccessful iteration: the algorithm may generate some new feasible non-dominated points without dominating the feasible frame incumbent.

- $\forall(\mathbf{x}^{l_k}, \Delta^{l_k}) \in L_F^k \setminus L_F^{k-1}$,

$$\mathbf{x}^{l_k} = \mathbf{x}^{k-1} + \delta^{k-1}\mathbf{D}\mathbf{z}^{k-1} \text{ for some } \mathbf{z}^{k-1} \in \mathbb{N}^{n_D} \text{ and } \Delta^{l_k} \in \{\Delta^{k-1}, \tau^{-1}\Delta^{k-1}\}$$

  with $\mathbf{x}^{k-1} \in \{\mathbf{x}_F^{k-1}, \mathbf{x}_I^{k-1}\}$.
- $\forall(\mathbf{x}^{l_k}, \Delta^{l_k}) \in (L_F^k \cap L_F^{k-1}) \setminus \{(\mathbf{x}_F^{k-1}, \Delta_F^{k-1})\}$,

$$\mathbf{x}^{l_k} = \mathbf{x}^{l_{k-1}} \text{ and } \Delta^{l_k} = \Delta^{l_{k-1}}.$$

- $\forall(\mathbf{x}^{l_k}, \Delta^{l_k}) \in \{(\mathbf{x}_F^{k-1}, \Delta_F^{k-1})\}$,

$$\mathbf{x}^{l_k} = \mathbf{x}_F^{k-1} \text{ and } \Delta^{l_k} = \tau\Delta^{k-1}.$$

Similar relations can be drawn for the DMulti-MADS-TEB variant algorithm : note that for all $k > k_0$, no point at iteration k can be generated from an infeasible point at iteration $k-1$.

One can then prove that feasible linked sequences contain a feasible refining subsequence. The original proof can be found in [40], but for better understanding, it is restated below.

Theorem 13. *Let assumptions 6.5.1 and 6.5.2 hold and suppose DMulti-MADS generates the sequence of feasible iterate lists* $\{L_F^k\}_{k\geq k_0}$ *with* $L_F^k = \{(\mathbf{x}^l, \Delta^l), l = 1, 2, \ldots, |L_F^k|\}$ *and* $k_0 \in \mathbb{N}$ *the iteration index such that* $k_0 \in \arg\min\left\{k \in \mathbb{N} : F^k \neq \emptyset\right\}$. *Then every feasible linked sequence* $\{(\mathbf{x}^{l_k}, \Delta^{l_k})\}$ *contains a refining subsequence* $\{\mathbf{x}^{l_k}\}_{k\in K}$ *for some infinite subset of indexes* $K \subset \mathbb{U}$.

Proof. $\forall k \geq k_0$, $0 \leq \Delta^{l_k} \leq \Delta_{\max}^k$. By combining Theorem 12 and the squeeze theorem, one gets

$$\lim_{k\to+\infty} \inf \Delta^{l_k} = \lim_{k\to+\infty} \inf \Delta_{\max}^k = 0,$$

which implies by definition the existence of a refining feasible subsequence within $\{(x^{l_k}, \Delta^{l_k})\}$. □

The analysis which follows is similar to [40].

Theorem 14. *Let assumptions 6.5.1 and 6.5.2 hold and suppose DMulti-MADS generates a feasible refining subsequence* $\{\mathbf{x}_F^k\}_{k\in K}$, *with* $\mathbf{x}_F^k \in F^k$, *converging to a refining point* $\hat{\mathbf{x}}_F \in \Omega$. *Assume that* f *is Lipschitz continuous near* $\hat{\mathbf{x}}_F$. *If* $\mathbf{d} \in \mathcal{T}_\Omega^H(\hat{\mathbf{x}}_F)$ *is a refining direction for* $\hat{\mathbf{x}}_F$, *then there exists an objective index* $i(\mathbf{d}) \in \{1, 2, \ldots, m\}$ *such that* $f_{i(\mathbf{d})}^o(\hat{\mathbf{x}}_F; \mathbf{d}) \geq 0$.

Proof. Let $\{\mathbf{x}_F^k\}_{k\in K}$, with $\mathbf{x}_F^k \in F^k$, be a refining subsequence converging to a feasible refining point $\hat{\mathbf{x}}_F \in \Omega$ and $\mathbf{d} = \lim_{k\in K'} \frac{\mathbf{d}^k}{\|\mathbf{d}^k\|} \in \mathcal{T}_\Omega^H(\hat{\mathbf{x}}_F)$ a refining direction for $\hat{\mathbf{x}}_F$, where $K' \subseteq K$ is an infinite subsequence of some infinite subset of unsuccessful iteration indexes, with poll directions $\mathbf{d}^k \in \mathbb{D}_\Delta^k$ such that $\mathbf{x}_F^k + \delta^k \mathbf{d} \in \Omega$. Denote by $\nu \geq 0$ the Lipschitz constant of f near $\hat{\mathbf{x}}_F$.

Then, for $i \in \{1, 2, \ldots, m\}$, the inequality

$$\begin{aligned}
f_i(\hat{\mathbf{x}}_F; \mathbf{d}) &= f_i^o(\hat{\mathbf{x}}_F; \mathbf{d}) + \limsup_{k\in K'} \frac{\nu\ \delta^k \|\mathbf{d}^k\| \left\| \frac{\mathbf{d}^k}{\|\mathbf{d}^k\|} - \mathbf{d} \right\|}{\delta^k \|\mathbf{d}^k\|} \\
&\geq f_i^o(\hat{\mathbf{x}}_F; d) + \limsup_{k\in K'} \frac{|f_i\left(\mathbf{x}_F^k + \delta^k \mathbf{d}^k\right) - f_i\left(\mathbf{x}_F^k + \delta^k \|\mathbf{d}^k\| \mathbf{d}\right)|}{\delta^k \|\mathbf{d}^k\|} \\
&\geq \limsup_{k\in K'} \frac{f_i\left(\mathbf{x}_F^k + \delta^k \|\mathbf{d}^k\| \mathbf{d}\right) - f(\mathbf{x}_F^k)}{\delta^k \|\mathbf{d}^k\|} \\
&+ \limsup_{k\in K'} \frac{|f_i\left(\mathbf{x}_F^k + \delta^k \mathbf{d}^k\right) - f_i\left(\mathbf{x}_F^k + \delta^k \|\mathbf{d}^k\| \mathbf{d}\right)|}{\delta^k \|\mathbf{d}^k\|}
\end{aligned}$$

$$\geq \limsup_{k\in K'} \frac{f(\mathbf{x}_F^k + \delta^k\mathbf{d}^k) - f_i\left(\mathbf{x}_F^k + \delta^k\|\mathbf{d}^k\|\mathbf{d}\right) + f_i\left(\mathbf{x}_F^k + \delta^k\|\mathbf{d}^k\|\mathbf{d}\right) - f_i(\mathbf{x}_F^k)}{\delta^k\|\mathbf{d}^k\|}$$
$$= \limsup_{k\in K'} \frac{f_i(\mathbf{x}_F^k + \delta^k\mathbf{d}) - f_i(\mathbf{x}_F^k)}{\delta^k\|\mathbf{d}^k\|}$$

is satisfied.

$\{\mathbf{x}_F^k\}_{k\in K}$ being a refining subsequence, the infinite subset of indexes $K' \subseteq K$ corresponds to unsuccessful iterations. Consequently, the point $\mathbf{x}_F^k + \delta^k\mathbf{d} \in \Omega$ does not dominate $\mathbf{x}_F^k$. One can then find an infinite subsequence of indexes $K'' \subset K'$ such that there exists an index $i(\mathbf{d}) \in \{1, 2, \ldots, m\}$ satisfying

$$f_{i(\mathbf{d})}(\hat{\mathbf{x}}_F; \mathbf{d}) \geq \limsup_{k\in K''} \frac{f_{i(\mathbf{d})}(\mathbf{x}_F^k + \delta^k\mathbf{d}) - f_{i(\mathbf{d})}(\mathbf{x}_F^k)}{\delta^k\|\mathbf{d}^k\|} \geq 0.$$

□

When the set of refining directions is dense in a non-empty hypertangent cone at Ω, Pareto Clarke stationarity is ensured, similarly to the analysis conducted in [40, 74].

Theorem 15. *Let assumptions 6.5.1 and 6.5.2 and suppose DMulti-MADS generates a feasible refining subsequence $\{\mathbf{x}_F^k\}_{k\in K}$, with $\mathbf{x}_F^k \in F^k$, converging to a refining point $\hat{\mathbf{x}}_F \in \Omega$. Assume that f is Lipschitz continuous near $\hat{\mathbf{x}}_F$ and $\mathcal{T}_\Omega^H(\hat{\mathbf{x}}_F) \neq \emptyset$. If the set of refining directions for $\hat{\mathbf{x}}_F$ is dense in $\mathcal{T}_\Omega^{Cl}(\hat{\mathbf{x}}_F)$, then $\hat{\mathbf{x}}_F$ is a Pareto-Clarke critical point of (MOP).*

Proof. The authors in [15] prove than for any direction $\mathbf{v}$ in the Clarke tangent cone,

$$f_i^0(\hat{\mathbf{x}}_F; \mathbf{v}) = \lim_{\substack{\mathbf{d} \in \mathcal{T}_\Omega^H(\hat{\mathbf{x}}_F) \\ \mathbf{d} \to \mathbf{v}}} f_i^o(\hat{\mathbf{x}}_F; \mathbf{d}) \text{ for } i = 1, 2, \ldots, m.$$

By hypothesis, the set of refining directions for $\hat{\mathbf{x}}_F \in \Omega$ is dense in $T_\Omega^{Cl}(\hat{x}_F)$. Then there exists a sequence of refining directions $\{\mathbf{d}^r\}_{r\in R} \in \mathcal{T}_\Omega^H(\hat{\mathbf{x}}_F)$ such that $\lim_{r\in R} \mathbf{d}^r = \mathbf{v}$. Since the number of components of the objective function is finite, one can find a subsequence $\{\mathbf{d}^r\}_{r\in R'}$ with $R' \subseteq R$ such that $\mathbf{v} = \lim_{r\in R'} \mathbf{d}^r$ and $f_{i(\mathbf{v})}(\hat{\mathbf{x}}_F; \mathbf{v}) \geq 0$ by Theorem 14 for all indexes $r \in R'$. Passing at the limit concludes the proof. □

6.5.2 Infeasible case: results for h

In this subsection, the goal is to analyse refining subsequences of infeasible points according to the h violation function as in [16] for the single-objective constrained case. Two cases can

occur. The refining point $\hat{\mathbf{x}}_I$ of an infeasible refining subsequence satisfies $h(\hat{\mathbf{x}}_I) = 0$. In this case, it means that the feasible set is non-empty, and that $\hat{\mathbf{x}}_I$ is a global minimum for the single-objective problem $\min_{\mathbf{x}\in\mathcal{X}} h(\mathbf{x})$. Otherwise, this work proves than $\hat{\mathbf{x}}_I$ satisfies some stationarity results relatively to h. Note that the DMulti-MADS-TEB variant generates an infeasible refining subsequence if and only if it starts from an infeasible point belonging to V^0 and generates no feasible points along the iterations.

Contrary to the feasible case, this work does not characterize particular infeasible sequences of points within the sequence of infeasible frame incumbents.

Theorem 16. *Let assumptions 6.5.1 and 6.5.2 hold and suppose DMulti-MADS generates a refining subsequence $\{\mathbf{x}_I^k\}_{k\in K}$, with $\mathbf{x}_I^k \in I^k$, converging to a refining point $\hat{\mathbf{x}}_I \in \mathcal{X}$. Assume that h is Lipschitz continuous near $\hat{\mathbf{x}}_I$. If $\mathbf{d} \in \mathcal{T}_{\mathcal{X}}^H(\hat{\mathbf{x}}_I)$ is a refining direction for $\hat{\mathbf{x}}_I$, then $h^o(\hat{\mathbf{x}}_I; \mathbf{d}) \geq 0$.*

Proof. The proof is similar to that of Theorem 14, h and $\mathcal{X}$ playing respectively the roles of f_i for a fixed objective index $i \in \{1, 2, \ldots, m\}$ and Ω. □

The next theorem's proof is identical to 15.

Theorem 17. *Let assumptions 6.5.1 and 6.5.2 hold and suppose DMulti-MADS generates a refining subsequence $\{\mathbf{x}_I^k\}_{k\in K}$, with $\mathbf{x}_I^k \in I^k$, converging to a refining point $\hat{\mathbf{x}}_I \in \mathcal{X}$. Assume that h is Lipschitz continuous near $\hat{\mathbf{x}}_I$ and $\mathcal{T}_{\mathcal{X}}^H(\hat{\mathbf{x}}_I) \neq \emptyset$. If the set of refining directions is dense for $\hat{\mathbf{x}}_I$ in $\mathcal{T}_{\mathcal{X}}^{Cl}(\hat{\mathbf{x}}_I)$, then $\hat{\mathbf{x}}_I$ is a Clarke stationary point for*

$$\min_{\mathbf{x}\in\mathcal{X}} h(\mathbf{x}).$$

Proof. The authors in [15] prove than for any direction $\mathbf{v}$ in the Clarke tangent cone,

$$h^0(\hat{\mathbf{x}}_I; \mathbf{v}) = \lim_{\substack{\mathbf{d} \in \mathcal{T}_{\mathcal{X}}^H(\hat{\mathbf{x}}_I) \\ \mathbf{d} \to \mathbf{v}}} h^o(\hat{\mathbf{x}}_I; \mathbf{d}).$$

By hypothesis, the set of refining directions for $\hat{\mathbf{x}}_I \in \mathcal{X}$ is dense in $\mathcal{T}_{\mathcal{X}}^{Cl}(\hat{\mathbf{x}}_I)$. Then there exists a sequence of refining directions $\{\mathbf{d}^r\}_{r\in R} \in \mathcal{T}_{\mathcal{X}}^H(\hat{\mathbf{x}}_I)$ such that $\lim_{r\in R} \mathbf{d}^r = \mathbf{v}$. By Theorem 16, for all $r \in R$, $h^o(\hat{\mathbf{x}}_I; \mathbf{d}) \geq 0$. Passing at the limit concludes the proof. □

6.6 Computational experiments

This section is devoted to the computational experiments of DMulti-MADS on constrained multiobjective problems. The first part presents the considered solvers. The second part is dedicated to the comparison of all solvers and DMulti-MADS variants on a set of analytical problems using data profiles for multiobjective optimization [40]. The last part shows comparison of solvers on "real" engineering problems using convergence profiles.

To assess the performance of different algorithms, this work relies on the use of data profiles for multiobjective optimization [40] and convergence profiles. Both tools require the definition of a convergence test for a given computational problem, based on the hypervolume indicator [279].

The hypervolume indicator represents the volume of the objective space dominated by a Pareto front approximation Y_N and delimited from above by a reference point $\mathbf{r} \in \mathbb{R}^m$ such that for all $\mathbf{y} \in Y_N$, $\mathbf{y}_i < \mathbf{r}_i$ for $i = 1, 2, \ldots, m$. The hypervolume possesses many useful properties: Pareto compliant with the dominance ordering, it can capture many properties of a Pareto front approximation as spread, cardinality, convergence to the Pareto front, or extension [11, 175].

The convergence test requires a Pareto front approximation reference Y^p for a given problem $p \in \mathcal{P}$, where $\mathcal{P}$ is the set of considered problems, from which the approximated ideal objective vector

$$\tilde{\mathbf{y}}^{I,p} = \left(\min_{\mathbf{y} \in Y^p} \mathbf{y}_1, \min_{\mathbf{y} \in Y^p} \mathbf{y}_2, \ldots, \min_{\mathbf{y} \in Y^p} \mathbf{y}_{m_p} \right)^\top$$

and the approximated nadir objective vector

$$\tilde{\mathbf{y}}^{N,p} = \left(\max_{\mathbf{y} \in Y^p} \mathbf{y}_1, \max_{\mathbf{y} \in Y^p} \mathbf{y}_2, \ldots, \max_{\mathbf{y} \in Y^p} \mathbf{y}_{m_p} \right)^\top$$

are extracted, with m_p the number of objectives of problem $p \in \mathcal{P}$. Y^p is constructed using the set of best non dominated points found by all algorithms on problem $p \in \mathcal{P}$ for a maximal budget of evaluations.

Assuming Y^e is a Pareto front approximation generated after e evaluations by a given deterministic solver for problem p, a scaling and translating transformation is applied to this last one defined by: $\forall \mathbf{y} \in Y^e \cup Y^p \cup \{\tilde{\mathbf{y}}^{N,p}\}$,

$$T(\mathbf{y}) = \begin{cases} (\mathbf{y} - \tilde{\mathbf{y}}^{I,p}) \oslash (\tilde{\mathbf{y}}^{N,p} - \mathbf{y}) & \text{if } \tilde{\mathbf{y}}^{N,p} \neq \tilde{\mathbf{y}}^{I,p}, \\ \mathbf{y} - \tilde{\mathbf{y}}^{I,p} & \text{otherwise;} \end{cases}$$

where $\oslash$ is the element wise-divisor operator. Note that this transformation conserves the dominance order relation. The computational problem is said to be solved by the algorithm with tolerance $\varepsilon_\tau > 0$ if

$$\frac{HV\left(T(Y^e); T(\tilde{\mathbf{y}}^{N,p})\right)}{HV\left(T(Y^p); T(\tilde{\mathbf{y}}^{N,p})\right)} \geq 1 - \varepsilon_\tau$$

where $HV(Y_N; \mathbf{r})$ is the hypervolume indicator value of the volume dominated by the Pareto front approximation Y_N and delimited above by the reference point $\mathbf{r} \in \mathbb{R}^m$. All elements of Y_N which are dominated by $\mathbf{r} \in \mathbb{R}^m$ are removed during the computation of the hypervolume indicator. If no element of Y_N dominates $\mathbf{r}$, then $HV(Y_N; \mathbf{r}) = 0$. In other terms, the hypervolume indicator is not computed in this case.

Data profiles show the proportion of all computational problems solved by an algorithm in function of the number of groups of $n+1$ evaluations required to build a simplex gradient in $\mathbb{R}^n$. In these experiments, stochastic solvers are also considered. In this case, data profiles are modified to take into account their performance variability, as described in [40].

6.6.1 Tested solvers and variants of DMulti-MADS

The following constrained solvers are considered:

- the deterministic solver `NOMAD` [167] which implements the BiMADS algorithm (Bi-objective Mesh Adaptive Direct Search) [27] tested only for $m = 2$ objectives - `www.gerad.ca/nomad/`;
- the deterministic solver DFMO (Derivative-Free Multi Objective) [180] - `http://www.iasi.cnr.it/~liuzzi/DFL/`;
- the stochastic heuristic solver NSGA-II (Non Dominating Sorting Algorithm II) [84]; a constrained version is implemented in the `Pymoo` Library [42] version `0.4.2.2` - `https://pymoo.org`.

For the BiMADS algorithm, two variants based on `NOMAD 3.9.1` are considered. The first uses the default settings of the MADS algorithm, detailed in [18,22,29,66]. The second deactivates models and other heuristics such that BiMADS relies only on the MADS algorithm with $n+1$ directions, a speculative search and an opportunistic polling strategy, for a fairer comparison with DMulti-MADS. DFMO and NSGA-II are used with their default settings. NSGA-II uses an initial population with 100 elements.

In these experiments, this work considers another variant of DMulti-MADS for constrained multiobjective optimization based on the penalty approach used in [180]. More specifically,

given the constrained multiobjective problem (MOP), the authors of [180] introduce the following penalty functions

$$Z_i(\mathbf{x};\varepsilon) = f_i(\mathbf{x}) + \frac{1}{\epsilon}\sum_{j\in\mathcal{J}} \max\{0, c_i(\mathbf{x})\},\ i = 1, 2, \ldots, m$$

where $\epsilon > 0$ is an external parameter and consider the following multiobjective problem

$$(MOP_p) : \min_{\mathbf{x}\in\mathcal{X}} Z(\mathbf{x}) = (Z_1(\mathbf{x};\epsilon), Z_2(\mathbf{x};\epsilon), \ldots, Z_m(\mathbf{x};\epsilon))^\top.$$

The DMulti-MADS-Penalty variant uses the DMulti-MADS-TEB variant on the modified (MOP_p) multiobjective problem. The external parameter $\epsilon > 0$ is set to the default value proposed by [180]. Note that this approach has already been used by these authors to compare DMS (which cannot start from infeasible points) and DFMO on constrained multiobjective problems [180]. As the first strategy proposed to handle constraints with convergence results, it is natural to see if this approach performs well compared to the two new variants proposed in this paper.

For all constrained variants of DMulti-MADS, a speculative search strategy is implemented as in [40] for one or both feasible and infeasible current incumbents if they exist, combined with a polling strategy with $n+1$ directions [22]. The implementation of the mesh follows a granular mesh strategy [18]. All variants stop as soon as one component of the mesh size vector is below 10^{-9} or after running out of evaluations budget. All variants use an opportunistic strategy: as soon as a new candidate dominates at least one current incumbent, the iteration stops. All variants also apply a spread selection with parameter value $w^+ = 1$. For the DMulti-MADS-PB variant, the trigger parameter is set to $\rho = 0.1$. When DMulti-MADS-TEB switches from the first phase to the second phase, the frame and mesh size parameters of the generated feasible points are not resettled to their respective initial values Δ^0 and δ^0.

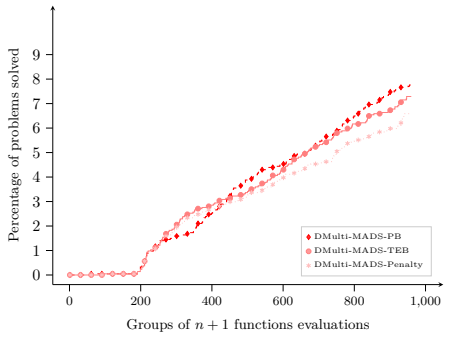
Finally, the implementation of the progressive barrier in `NOMAD 3.9.1` for the BiMADS algorithm diverges from the description given in [16] for efficiency gains. The DMulti-MADS-PB algorithm variant equally incorporates these modifications for a fairer comparison with the implementation of `NOMAD 3.9.1`. Precisely, the threshold $h^k_{\max}$ is updated according to the set $U^{k+1} \subseteq V^{k+1}$, which enables it to decrease faster. Furthermore, in the implementation, an iteration k is considered as improving if the algorithm generates a point $\mathbf{x}^t \in V^{k+1} \setminus V^k$ satisfying improving conditions. Note that the convergence properties for infeasible refining subsequences still hold. However, it may exist a point $\mathbf{x} \in \cup_{k\in\mathbb{N}} V^k$ with $0 < h(\mathbf{x}) < h(\hat{\mathbf{x}}^I)$ where $\hat{\mathbf{x}}^I \in \mathcal{X}$ is an infeasible refining point.

The code used for experiments can be found at `https://github.com/bbopt/DMultiMadsPB`.

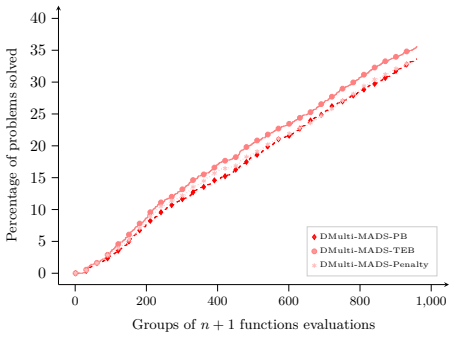
6.6.2 Comparing solvers on synthetic benchmarks

In this subsection, this work considers a set of 214 analytical multiobjective optimization problems proposed by [180], with $n \in [3, 30]$, $m \in \{2, 3, 4\}$ and $|\mathcal{J}| \in [3, 30]$. Among them, 103 problem possess $m = 2$ objectives.

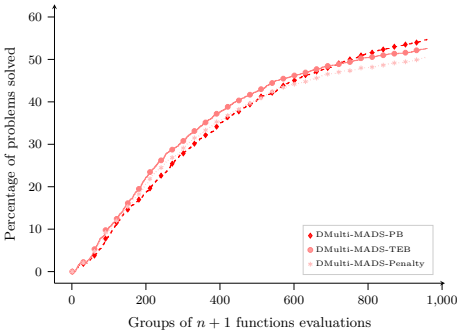
In a first part, this work compares the three variants of DMulti-MADS on this set of problems. The three of them use a maximum budget of $30,000$ evaluations. For each problem, the three variants start from the same set of initial points, using the linesearch strategy described in [74]. Each variant on each problem executes 10 replications by changing the random seed which controls the generation of polling directions.



(a) $\varepsilon_\tau = 10^{-2}$



(b) $\varepsilon_\tau = 5 \times 10^{-2}$



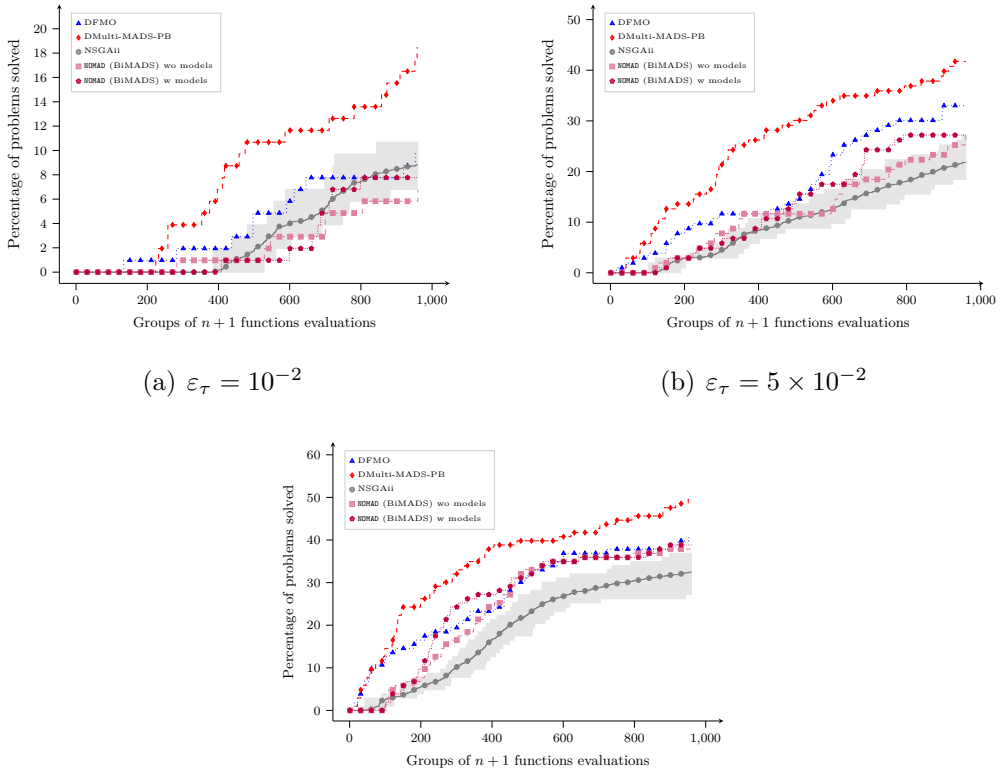
(c) $\varepsilon_\tau = 10^{-1}$

Figure 6.9 Data profiles obtained from 10 replications from 214 multiobjective analytical problems taken from [180] for DMulti-MADS-PB, DMulti-MADS-TEB and DMulti-MADS-Penalty with tolerance $\varepsilon_\tau \in \{10^{-2}, 5 \times 10^{-2}, 10^{-1}\}$.

The data profiles given in Figure 6.9 show that for the three tolerance values considered, DMulti-MADS-Penalty solves slightly less problems than the two other variants introduced in this work. One can equally observe than DMulti-MADS-PB performs better for a medium

to high budget of evaluations for the lowest tolerance $\varepsilon_\tau = 10^{-2}$. For the largest tolerance $\varepsilon_\tau = 10^{-1}$, DMulti-MADS-PB solves more problems for a high budget of evaluations. However, for medium tolerance, the performance of DMulti-MADS-PB is similar to DMulti-MADS-Penalty. A closer look at the considered problems shows than in this case, it is better to firstly look for feasible solutions than to explore the infeasible decision space. It then gives an advantage to DMulti-MADS-TEB over the two other variants. For the rest of this subsection, only DMulti-MADS-PB and DMulti-MADS-TEB are kept, as they are more performant.

For the comparison with the other algorithms, the same maximum budget of $30,000$ function evaluations is kept. Practically, for NSGA-II, the total number of population generations is fixed to 300, with a fixed population size equal to 100. For each problem, the deterministic solvers start from the same initial points using the linesearch strategy [74]. For each problem, NSGA-II is run 30 times with different seeds to capture stochastic behavior and analyze its performance variation.

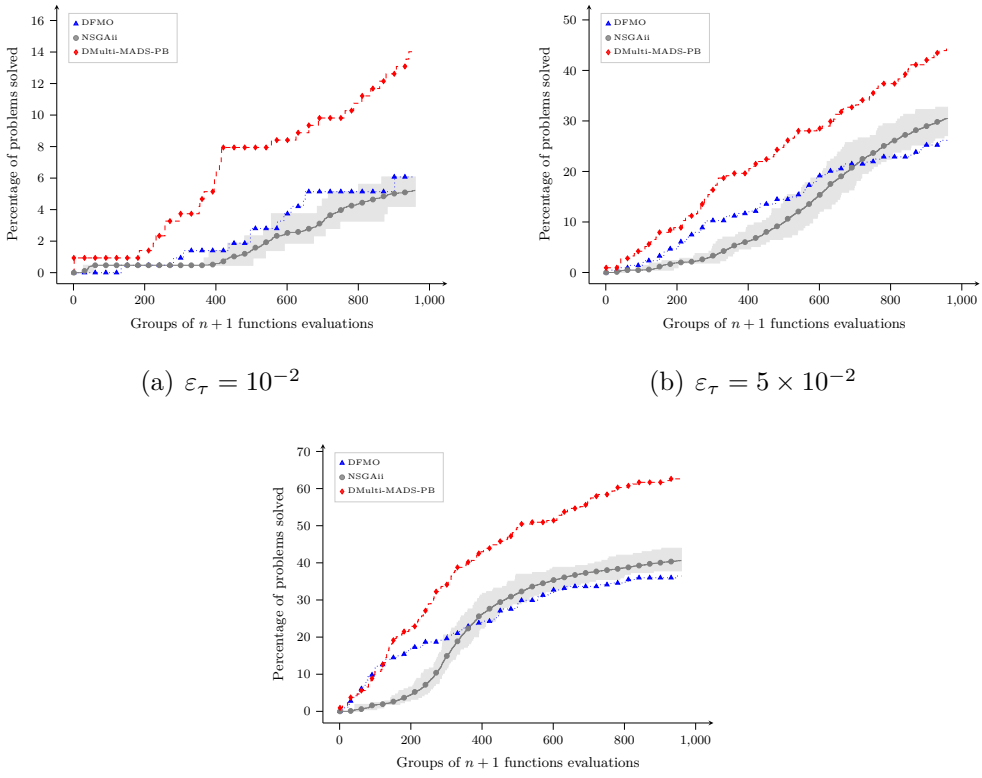


(a) $\varepsilon_\tau = 10^{-2}$

(b) $\varepsilon_\tau = 5 \times 10^{-2}$

(c) $\varepsilon_\tau = 10^{-1}$

Figure 6.10 Data profiles using NOMAD (BiMADS), DFMO, DMulti-MADS-PB and NSGA-II obtained on 103 biojective analytical problems from [180] with 30 different runs of NSGA-II with tolerance $\varepsilon_\tau \in \{10^{-2}, 5 \times 10^{-2}, 10^{-1}\}$.



(a) $\varepsilon_\tau = 10^{-2}$

(b) $\varepsilon_\tau = 5 \times 10^{-2}$

(c) $\varepsilon_\tau = 10^{-1}$

Figure 6.11 Data profiles using DFMO, DMulti-MADS-PB and NSGA-II obtained on 214 multiobjective analytical problems from [180] with 30 different runs of NSGA-II with tolerance $\varepsilon_\tau \in \{10^{-2}, 5 \times 10^{-2}, 10^{-1}\}$.

From Figures 6.10 and 6.11, one can see that DMulti-MADS-PB outperforms the other solvers on this set of analytical functions, for all tolerances considered. The same conclusions can be drawn for DMulti-MADS-TEB. From these figures, one can also observe than DFMO displays better performance on biojective problems than for the whole set (see Figure 6.10).

6.6.3 Comparing solvers on real engineering benchmarks

In this subsection, this work considers three multiobjective optimization problems: the biobjective SOLAR8 and SOLAR9 design problems and the triobjective STYRENE design problem [10,28]. These three applications are more costly to solve than the analytical benchmarks considered in the previous subsection. The use of data profiles to compare solvers on these problems is then difficult to put into practice.

To assess the performance of solvers on these problems, an adaptation of convergence profiles

(see [20, Appendix A] for a description) to multiobjective optimization is proposed. Convergence profiles for multiobjective optimization make use of the normalized hypervolume value, presented at the beginning of Section 6.6 and given by:

$$\frac{HV\left(T(Y^e); T(\tilde{\mathbf{y}}^{N,p})\right)}{HV(T(Y^p); T(\tilde{\mathbf{y}}^{N,p}))}$$

where Y^p is the Pareto front approximation reference for problem p, Y^e the Pareto front approximation generated after e evaluations by a given solver on an instance of problem p, T a scaling and translating transformation applied and $\tilde{y}^{N,p}$ the approximated nadir objective vector of Y^p.

Convergence profiles for multiobjective optimization on a given problem p visualize the evolution of the normalized hypervolume indicator for a given solver against the number of evaluations used. Consequently, a normalized hypervolume value equal to 1 means that the solver has solved the problem p. A normalized hypervolume equal to 0 means that the solver has not generated points which dominate the approximated nadir objective vector of the Pareto front approximation reference.

Comparing solvers on the SOLAR8 and SOLAR9 design problems

SOLAR8 and SOLAR9 are two biobjective optimization problems derived from a numerical simulator coded in C++ of a solar plant with a molten salt heat storage system [125]. The simulation is composed of three steps. The heliostats field captures sun rays which are transmitted to a central cavity receiver. The sun energy is given to the thermal storage which converts it to thermal energy. This last one activates the powerblock, which triggers a steam turbine, generating electrical power output. Numerical simulations intervene all along the different phases of the process, which make it impossible to provide gradients. For more details, the reader can refer to [125]. The simulator can be found at `https://github.com/bbopt/solar`.

For the two considered problems, a blackbox evaluation can take more than 10 seconds (on a machine with 8 Intel(R) Core(TM) i7-2600K CPU @ 3.40GHz 16G RAM). Experiments equally reveal the presence of hidden constraints. Tables 6.12 and 6.13 describe the objectives, constraints and starting points used for each problem.

SOLAR8 and SOLAR9 both possess integer decision variables. In the experiments, the only solver which can treat integer variables is `NOMAD` (BiMADS). Consequently, for the other solvers, all integer variables are fixed to their starting values along the optimization. For the

Constraints/Objectives	Description of constraints and objectives
$-f_1$	Maximize heliostat field performance (absorbed energy)
f_2	Minimize cost of field, tower and receiver
Heliostat design constraints	Four constraints related to the dimensions of the heliostat field
Receiver constraints	Three constraints related to the design of the receiver
Energy constraints	Two constraints which depend on the energy production
Variables	Description and type
Heliostats field	Nine variables related to the dimensions of the heliostats field Eight real and one integer
Heat transfer loop	Four variables related to the design of the heat transfer system Three real and one integer
Starting point (infeasible)	$(11.0, 11.0, 200.0, 10.0, 10.0, 2650, 89.0, 0.5, 8.0, 36, 0.30, 0.020, 0.0216)$

Figure 6.12 Objectives, constraints, variables and starting point of the SOLAR8 problem.

Constraints/Objectives	Description of constraints and objectives
f_1	Minimize production costs
$-f_2$	Maximize energy production
Heliostats design constraints	Four constraints related to the dimensions of the heliostat field
Heat storage constraints	Four constraints relative to the molten salt heat thermic/pressure storage system
Receiver design constraints	Two constraints which depend on the tube size and diameter receiver
Steam constraints	Five constraints related to steam temperature, power output and steam design.
Variables	Description and type
Heliostats field	Nine variables related to the dimensions of the heliostats field Eight real and one integer
Heat transfer loop	Nineteen variables related to the design of the heat transfer system Fourteen real and five integer
Powerblock	One variable: type of turbine; integer
Starting point (infeasible)	$(9.0, 9.0, 150.0, 6.0, 8.0, 1000, 45.0, 0.5, 5.0, 900.0,$ $9.0, 9.0, 0.30, 0.20, 560.0, 500, 0.30, 0.0165, 0.018, 0.017,$ $10.0, 0.0155, 0.016, 0.20, 3, 12000, 1, 2, 2)$

Figure 6.13 Objectives, constraints, variables and starting point of the SOLAR9 problem.

SOLAR8 problem, three variants of `NOMAD` (BiMADS) are considered: two for which integer variables are fixed and one which treat mixed integer (MI) problems. For this last variant, the algorithmic parameters are chosen by default.

Remark. For SOLAR9, `NOMAD` (BiMADS) completely outperforms the other algorithms when it can modify integer variables. After investigation, this behaviour is not related to the performance of the algorithm, but the initial choice of the integer variables. However, for the sake of reproducibility, these values are kept.

All deterministic algorithms are allocated a maximal budget of $5,000$ evaluations and start from the same infeasible point for each problem. NSGA-II does not take starting points as arguments. To compare it with the others, NSGA-II is run 10 times to capture stochastic behaviour, with a population size fixed to 100 and a total number of generations equal to 50.

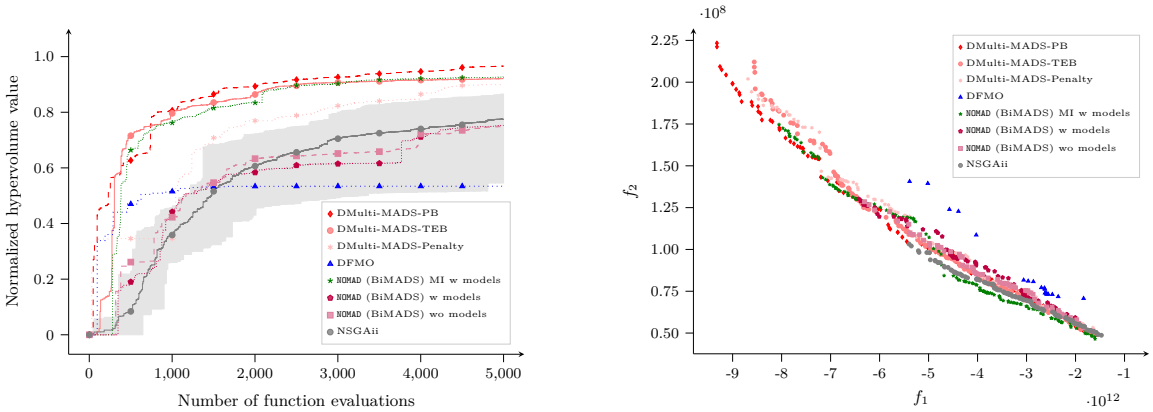


Figure 6.14 (a) On the left, convergence profiles for the SOLAR8 problem using DFMO, DMulti-MADS, NOMAD (BiMADS) and NSGA-II with 10 different runs of NSGA-II for a maximal budget of $5,000$ evaluations. (b) On the right, Pareto front approximations obtained at the end of the resolution of SOLAR8 for DFMO, DMulti-MADS, NOMAD (BiMADS) and an instance of NSGA-II in the objective space.

From Figure 6.14(a), one can see that DMulti-MADS-PB performs better than the other algorithms on SOLAR8. When looking at the Pareto front plottings (Figure 6.14(b)), one can note that DMulti-MADS-PB captures a portion of the Pareto front on the top left. DMulti-MADS-TEB is slightly better than DMulti-MADS-Penalty and compares well in terms of performance with NOMAD (BiMADS) when allowing the use of mixed integer variables. DFMO does not perform well on this problem, due to the different scales on the constraints included in the penalty objective function, which impacts its efficiency.

Figure 6.15(a) shows the convergence profiles for the SOLAR9 problem for different solvers. On this problem, NOMAD (BiMADS) are the most efficient, even if DMulti-MADS-PB catches it for the last evaluations. As shown on Figure 6.15(b), the extent of the Pareto front approximation reference is low, which favours scalarization-based approaches such as BiMADS. This problem also illustrates the default of penalty-based approaches against other methods. As the constraint functions possess different amplitudes, the penalized optimization problem differs from the original, which explains why DFMO and DMulti-MADS-Penalty fail.

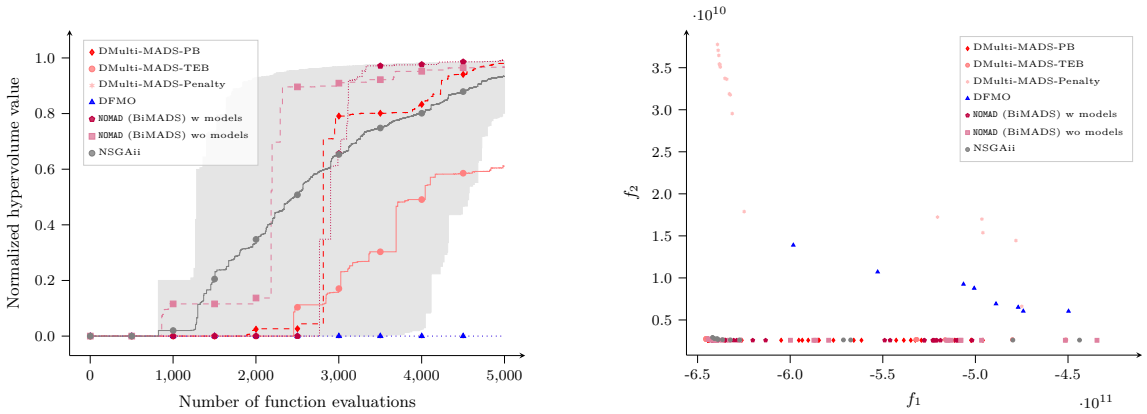


Figure 6.15 (a) On the left, convergence profiles for the SOLAR9 problem using DFMO, DMulti-MADS, NOMAD (BiMADS) and NSGA-II with 10 different runs of NSGA-II for a maximal budget of 5,000 evaluations. (b) On the right, Pareto front approximations obtained at the end of the resolution of SOLAR9 for DFMO, DMulti-MADS, NOMAD (BiMADS) and an instance of NSGA-II in the objective space.

Comparing solvers on the STYRENE design problem

STYRENE is a triobjective optimization problem related to the production of styrene, as described in [10, 28]. Styrene production process is composed of four steps: reactants preparation, catalytic reactions, a first distillation to recover styrene and a second one to recover benzene. The second distillation equally involves the recycling of unreacted ethylbenzaline, reintroduced into the styrene production as an initial reactant. The proposed triobjective optimization problem, based on a numerical implementation coded in C++, aims at maximizing the net present value associated to the process (f_1), the purity of produced styrene (f_2), and the overall ethylbenzene conversion into styrene (f_3). This application possesses eight bounded variables, and nine general inequality constraints related to the chemical process (e.g., environmental regulations), or costs (e.g., investment). More details can be found in [28].

A simulation takes around 1 second to run, starting from a feasible point (on a machine with 8 Intel(R) Core(TM) i7-2600K CPU @ 3.40GHz 16G RAM). This problem has hidden constraints. Even when starting from a feasible point, the simulation can sometimes fail to produce a finite numerical value.

A maximal budget of 20,000 evaluations is allocated for all deterministic solvers, which all start from the same feasible point. This experiment does not consider NOMAD (BiMADS), as it only treats biojective problems. NSGA-II is run 10 times, with a population size fixed to

100, and a maximal number of generations equal to 200.

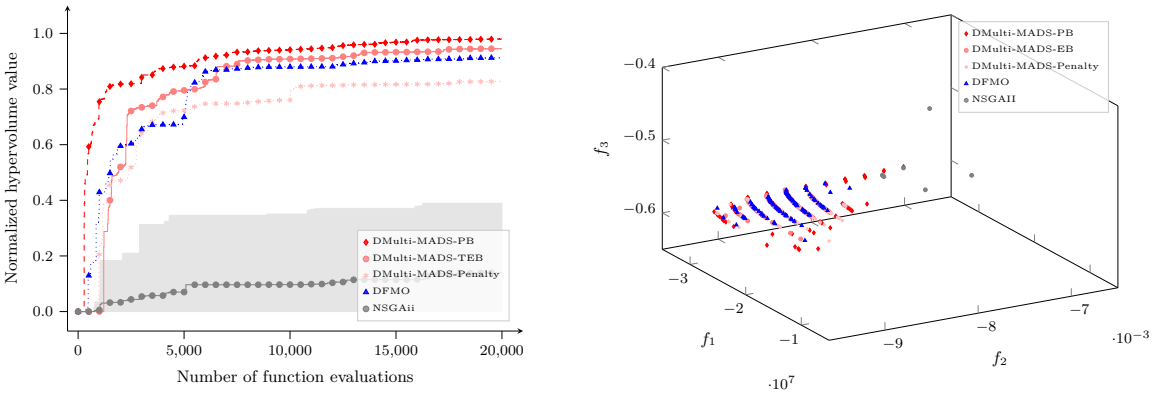


Figure 6.16 (a) On the left, convergence profiles for the STYRENE design problem using DFMO, DMulti-MADS, and NSGA-II with 10 different runs of NSGA-II for a maximal budget of 20,000 evaluations. (b) On the right, Pareto front approximations obtained at the end of the resolution of STYRENE for DFMO, DMulti-MADS, and an instance of NSGA-II in the objective space.

Figure 6.16(a) shows the convergence profiles obtained for the STYRENE design triobjective problem. This figure shows that DMulti-MADS-PB performs better than the other solvers, followed by DMulti-MADS-TEB. From Figure 6.16(b), one can observe that DMulti-MADS-PB captures more parts of the Pareto front reference than all the other methods. Finally, even when taking into account variability, NSGA-II is less efficient than all the other solvers on this problem.

6.7 Discussion

This work proposes two extensions of the DMulti-MADS algorithm [40] to handle blackbox constraints, generalizing the works conducted for the single-objective MADS algorithm [16, 17]. It is proved that these two extensions possess the same convergence properties than DMulti-MADS [40] when studying feasible sequences generated by these two extensions. Convergence analysis for the infeasible case is also derived, as in [16].

Experiments show that these two variants are competitive comparing to other state-of-the-art methods, and more robust on real engineering applications than a penalty-based approach, as proposed in [180]. These experiments also reveal that a two-phase approach performs surprisingly well on blackbox multiobjective optimization problems, contrary to single-objective ones [17]. Future work involves the integration of surrogate methods into a search strategy [53, 66], and the use of parallelism. An integration in the `NOMAD` solver is also planned.

CHAPTER 7 GENERAL DISCUSSION

7.1 Summary of Works

This section summarizes the contributions detailed in Chapter 4 (the survey on performance indicators), Chapter 5 (DMulti-MADS) and Chapter 6 (DMulti-MADS-TEB and DMulti-MADS-PB), and examines the obtained results.

The first objective was to conduct a state-of-the-art survey on performance indicators for MOO. Thus, 63 metrics have been listed and classified into four groups: cardinality, convergence, distribution and spread, and convergence and distribution. This survey analyses their main properties in relation with the theory of Pareto front and set approximations proposed by [159, 277, 280]. Finally, it presents four application cases of performance indicators: comparison of algorithms, their embedding into multiobjective methods, their use as stopping criteria for MOO, and the identification of promising distribution-based metrics. The comparison and performance analysis of different algorithms are the most important application in our context. The use of Pareto compliant metrics, i.e., which take into consideration Pareto dominance in the objective space is then critical. The hypervolume indicator can be considered the most relevant of all of them. Intuitive to understand, it captures convergence and distribution properties and does not require the knowledge of the Pareto front. Its main drawback is its complexity, which grows exponentially as the number of objectives increases. However, efficient algorithms exist for the low-dimensional case (for two to four objectives), which encompasses all our engineering applications.

In Chapter 5, the objective was to design an a posteriori articulation of preferences method extending MADS for MOO as efficient as state-of-the-art deterministic MOO methods. DMulti-MADS generalizes MADS to several objectives. It is not restricted to biobjective BBO problems, contrary to BiMADS. Inspired by DMS, it generates sequences of sets of points whose stationary points are locally Pareto optimal. Chapter 5 also introduces hypervolume-based data profiles for MOO. As in SOO, this tool examinates the robustness and efficiency of an algorithm in function of its evaluation budget (contrary to performance profiles). Using the hypervolume indicator, it offers a finer analysis than purity-based data profiles [74]. Numerical experiments show that DMulti-MADS is competitive according to state-of-the-art algorithms on a set of 100 bound-constrained analytical problems. More precisely, it is more efficient for a low to medium budget of evaluations with high tolerance, whereas NSGA-II and BiMADS display better performance for a large budget of evaluations.

Chapter 6 introduces two new extensions of DMulti-MADS to handle general inequality constraints, inspired by the works done on MADS [16, 17]: a two-phase approach, designed as DMulti-MADS-TEB, and an extension based on the progressive barrier for SOO [16], named DMulti-MADS-PB. This chapter develops new convergence analysis results for the constrained case for both methods. Their performance is put into perspective with a penalty-based approach proposed in [180]. These two algorithms bring important gains on analytical problems compared to other methods such as DFMO, NSGA-II and BiMADS. They equally perform well on three real engineering applications compared to state-of-the-art algorithms. Numerical experiments reveal than these two methods are more robust than a penalty-based approach, whose performance highly depends on the choice of penalty parameters. DMulti-MADS-PB is more difficult to implement than DMulti-MADS-TEB, but performs slightly better on real engineering problems. However, the two-phase approach is very reliable and efficient on all benchmarks, without introducing new algorithmic parameters.

7.2 Limitations

This thesis proposes to advance the state-of-the-art in deterministic multiobjective BBO and benchmarking tools to compare them. However, the following limitations can be pointed out:

- Enforcing a selection-based criterion enables DMulti-MADS to obtain stronger convergence properties than DMS. However, it introduces a new algorithmic parameter which requires to be tuned, as it impacts the performance. The latest version of DMulti-MADS uses a parameter value equal to $w^+ = 1$, but it may exist more efficient options depending on the polling strategy.

- Contrary to BiMADS, DMulti-MADS needs to loop over all feasible incumbents and infeasible undominated solutions to choose its frame centers. When the number of elements in the Pareto front approximation increases, this can result in a slowdown. If the blackbox is not expensive and one requires the computation of a Pareto front approximation in a short time, DMulti-MADS may not be a good choice, as it does not impose a limit on the solution set size.

- DMulti-MADS is specifically suited for nonsmooth MOO blackbox problems. However, some engineering applications possess smoothness properties even though derivatives are not available. DMulti-MADS does not exploit this information.

- The analytical benchmarks used in Chapters 5 and 6 represent an extensive database to compare new MOO methods. However, one needs to be careful not to rely too much

on them, but also use more realistic benchmarks (if possible). Indeed, all analytical constrained problems [180] used in Chapter 6 have smooth constraints. More problematic is that 23 of the 100 bound-constrained problems (with $n \geq 2$ variables) collected in [74] and used in Chapter 5, have as an objective a function of one variable (more often $\mathbf{x}_1$). One could also artificially extend this set of problems by considering different starting points as described in [18], instead of executing an unique initial linesearch strategy [74]. The experimental settings used in these two chapters are the result of a compromise between accuracy (for example, to capture the stochastic behaviour of NSGA-II; and to not advantage a solver according to another by not starting from the same points, when available), resource constraints (data profiles for multiobjective optimization are extremely memory consuming) and time.

- DMulti-MADS is not designed to deal with equality-constraint problems. Indeed, transforming equality constraints into two inequalities results in theoretical convergence issues and performance losses, as in SOO [23].

CHAPTER 8 CONCLUSION AND RECOMMENDATIONS

> *C'est une toile d'environ un mètre soixante sur un mètre vingt, peinte en blanc. Le fond est blanc et si on cligne des yeux, on peut apercevoir de fins liserés blancs transversaux.*
>
> Yasmina Reza, *Art*

This thesis proposes a new generalization of MADS to multiobjective BBO designed as DMulti-MADS, competitive with other state-of-the-art methods. As many research works, this thesis certainly opens more questions than it answers. However, some seem important for us to mention.

Since its conception in 2006, MADS has benefited from several extensions which considerably improve its performance. It is natural to try to include them in DMulti-MADS.

- DMulti-MADS could integrate quadratic models. For example, BoostDMS implements a quadratic search for DMS [53] which could be added to our algorithm. One could go further and use surrogates to sort polling directions, as it is done in SOO [66]. In our opinion, the two following research directions deserve to be investigated: the use of single-objective formulations [27, 28] or single-objective quadratic models according to the nature of the selected frame incumbent; the use of the ϕ_{L^k} function, introduced in Chapter 6, combined with quadratic models, equally opens interesting possibilities.

- The Nelder-Mead search, introduced in [29] is a powerful heuristic used in SOO for MADS. One could extend it to DMulti-MADS, inspiring by BiMADS single-objective reformulations.

- DMulti-MADS remains a convergence-based local method. However, one could integrate some heuristics, in the lineage of GLODS and MultiGLODS [72, 73], or Variable Neighbourhood Search [10].

- DMulti-MADS is a sequential procedure. Making it parallel represents an interesting research challenge. Some works have already been conducted for DMS [240].

- Finally, the adaptation of DMulti-MADS to take into account integer and binary variables or linear equality constraints is straightforward [18, 23].

The future integration of DMulti-MADS into `NOMAD` [24, 25] should make the development of these new extensions straightforward, especially since the new architecture of this software

has considerably improved its maintainability.

The convergence and numerical results developed in Chapter 5 and 6, inspired by [16,74,180] raise interesting questions.

- State-of-the art a posteriori articulation of preference methods for deterministic MODFO employ one of the following iteration schemes. The first involves iterating over all elements of the current Pareto front approximation before updating it. DFMO subscribes to this category, but is not efficient (see Chapter 6 and [126]). The second consists in choosing a current incumbent from the list of non-dominated solutions, performing some optimization steps, and updating the current solution set. DMS, MOIF and BOTR belong to this group. Although less effective, the first mechanism enforces convergence to a set of Pareto stationary points. Numerical experiments conducted in Chapter 5 show that imposing a selection-based strategy to guarantee "convergence to a set" does not penalize the performance of DMulti-MADS. Consequently, it may be possible to strengthen convergence results of these methods by adapting the selection mechanism of DMulti-MADS to these algorithms (adding a selection restriction using the trust-region radius, stepsize, or approximated gradient of the current incumbent) without performance loss. Some derivative-based MOO algorithms could also adopt this mechanism (e.g. [60,61,118]).

- A further topic should be to extend direct search methods for multiobjective optimization to multicriteria optimization [98] using more general ordering cones than the cone $\mathbb{R}^m_+$. Two potential applications are lexicographic BBO (see [69]) and construction of ε-efficient set approximations (see [200] for a definition).

- Finally, Chapter 5 formalizes hypervolume-based data profiles for MOO, following Audet et al. [11] recommendations. Researchers may use them to conduct an extensive comparison study of multiobjective BBO solvers. Such review would benefit the whole optimization community. One can look at [202,217] for some examples.

REFERENCES

[1] M. Abramson and C. Audet, Convergence of mesh adaptive direct search to second-order stationary points, SIAM Journal on Optimization **17** no. 2 (2006), 606–619. https://doi.org/Doi:10.1137/050638382.

[2] M. Abramson, C. Audet, J. Dennis, Jr., and S. Le Digabel, OrthoMADS: A Deterministic MADS Instance with Orthogonal Directions, SIAM Journal on Optimization **20** no. 2 (2009), 948–966. https://doi.org/10.1137/080716980.

[3] T. Akhtar and C. Shoemaker, Multi objective optimization of computationally expensive multi-modal functions with RBF surrogates and multi-rule selection, Journal of Global Optimization **64** no. 1 (2016), 17–32. https://doi.org/10.1007/s10898-015-0270-y.

[4] A. Al-Dujaili and S. Suresh, BMOBench: Black-Box Multi-Objective Optimization Benchmarking Platform, Tech. Report 1605.07009, arXiv, 2016. Available at https://arxiv.org/abs/1605.07009.

[5] A. Al-Dujaili and S. Suresh, Revisiting norm optimization for multi-objective black-box problems: a finite-time analysis, Journal of Global Optimization **73** no. 3 (2018), 659–673. https://doi.org/10.1007/s10898-018-0709-z.

[6] S. Alexandropoulos, C. Aridas, S. Kotsiantis, and M. Vrahatis, Multi-Objective Evolutionary Optimization Algorithms for Machine Learning: A Recent Survey, pp. 35–55, Springer International Publishing, Cham, 2019. Available at https://doi.org/10.1007/978-3-030-12767-1_4.

[7] R. Alimo, P. Beyhaghi, and T. Bewley, Delaunay-based derivative-free optimization via global surrogates. part III: nonconvex constraints, Journal of Global Optimization **77** no. 4 (2020), 743–776. https://doi.org/10.1007/s10898-019-00854-2.

[8] M. Asafuddoula, T. Ray, and H. Singh, Characterizing Pareto Front Approximations in Many-objective Optimization, in Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, ACM, New York, NY, USA, 2015, pp. 607–614. https://doi.org/10.1145/2739480.2754701.

[9] P. B. Assunção, O. P. Ferreira, and L. F. Prudente, Conditional gradient method for multiobjective optimization, Computational Optimization and Applications **78** no. 3 (2021), 741–768. https://doi.org/10.1007/s10589-020-00260-5.

[10] C. Audet, V. Béchard, and S. Le Digabel, Nonsmooth optimization through Mesh Adaptive Direct Search and Variable Neighborhood Search, Journal of Global Optimization **41** no. 2 (2008), 299–318. https://doi.org/10.1007/s10898-007-9234-1.

[11] C. Audet, J. Bigeon, D. Cartier, S. Le Digabel, and L. Salomon, Performance indicators in multiobjective optimization, European Journal of Operational Research **292** no. 2 (2021), 397–422, Invited Review. https://doi.org/10.1016/j.ejor.2020.11.016.

[12] C. Audet, A. Conn, S. Le Digabel, and M. Peyrega, A progressive barrier derivative-free trust-region algorithm for constrained optimization, Computational Optimization and Applications **71** no. 2 (2018), 307–329. https://doi.org/10.1007/s10589-018-0020-4.

[13] C. Audet and J. Dennis, Jr., Analysis of generalized pattern searches, SIAM Journal on Optimization **13** no. 3 (2003), 889–903. https://doi.org/Doi:10.1137/S1052623400378742.

[14] C. Audet and J. Dennis, Jr., A pattern search filter method for nonlinear programming without derivatives, SIAM Journal on Optimization **14** no. 4 (2004), 980–1010. https://doi.org/10.1137/S105262340138983X.

[15] C. Audet and J. Dennis, Jr., Mesh Adaptive Direct Search Algorithms for Constrained Optimization, SIAM Journal on Optimization **17** no. 1 (2006), 188–217. https://doi.org/10.1137/040603371.

[16] C. Audet and J. Dennis, Jr., A Progressive Barrier for Derivative-Free Nonlinear Programming, SIAM Journal on Optimization **20** no. 1 (2009), 445–472. https://doi.org/10.1137/070692662.

[17] C. Audet, J. Dennis, Jr., and S. Le Digabel, Globalization strategies for Mesh Adaptive Direct Search, Computational Optimization and Applications **46** no. 2 (2010), 193–215. https://doi.org/10.1007/s10589-009-9266-1.

[18] C. Audet, S. L. Digabel, and C. Tribes, The Mesh Adaptive Direct Search Algorithm for Granular and Discrete Variables, SIAM Journal on Optimization **29** no. 2 (2019), 1164–1189. https://doi.org/10.1137/18M1175872.

[19] C. Audet, K. Dzahini, M. Kokkolaras, and S. Le Digabel, Stochastic mesh adaptive direct search for blackbox optimization using probabilistic estimates, Computational Optimization and Applications **79** no. 1 (2021), 1–34. https://doi.org/10.1007/s10589-020-00249-0.

[20] C. Audet and W. Hare, Derivative-Free and Blackbox Optimization, Springer Series in Operations Research and Financial Engineering, Springer, Cham, Switzerland, 2017. https://doi.org/10.1007/978-3-319-68913-5.

[21] C. Audet and W. Hare, Model-Based Methods in Derivative-Free Nonsmooth Optimization, pp. 655–691, Springer International Publishing, 2020. https://doi.org/10.1007/978-3-030-34910-3_19.

[22] C. Audet, A. Ianni, S. Le Digabel, and C. Tribes, Reducing the Number of Function Evaluations in Mesh Adaptive Direct Search Algorithms, SIAM Journal on Optimization **24** no. 2 (2014), 621–642. https://doi.org/10.1137/120895056.

[23] C. Audet, S. Le Digabel, and M. Peyrega, Linear equalities in blackbox optimization, Computational Optimization and Applications **61** no. 1 (2015), 1–23. https://doi.org/10.1007/s10589-014-9708-2.

[24] C. Audet, S. Le Digabel, V. Rochon Montplaisir, and C. Tribes, NOMAD version 4: Nonlinear optimization with the MADS algorithm, Tech. Report G-2021-23, Les cahiers du GERAD, 2021. Available at http://www.optimization-online.org/DB_HTML/2021/04/8351.html.

[25] C. Audet, S. Le Digabel, and C. Tribes, NOMAD user guide, Tech. Report G-2009-37, Les cahiers du GERAD, 2009. Available at https://www.gerad.ca/nomad/Downloads/user_guide.pdf.

[26] C. Audet, S. Le Digabel, and C. Tribes, Dynamic scaling in the mesh adaptive direct search algorithm for blackbox optimization, Optimization and Engineering **17** no. 2 (2016), 333–358. https://doi.org/10.1007/s11081-015-9283-0.

[27] C. Audet, G. Savard, and W. Zghal, Multiobjective Optimization Through a Series of Single-Objective Formulations, SIAM Journal on Optimization **19** no. 1 (2008), 188–210. https://doi.org/10.1137/060677513.

[28] C. Audet, G. Savard, and W. Zghal, A mesh adaptive direct search algorithm for multiobjective optimization, European Journal of Operational Research **204** no. 3 (2010), 545–556. https://doi.org/10.1016/j.ejor.2009.11.010.

[29] C. Audet and C. Tribes, Mesh-based Nelder-Mead algorithm for inequality constrained optimization, Computational Optimization and Applications **71** no. 2 (2018), 331–352. https://doi.org/10.1007/s10589-018-0016-0.

[30] A. Auger, J. Bader, and D. Brockhoff, Theoretically Investigating Optimal μ-Distributions for the HyperVolume Indicator: First Results for Three Objectives, in Parallel Problem Solving from Nature, PPSN XI (R. Schaefer, C. Cotta, J. Kołodziej, and G. Rudolph, eds.), Springer, Berlin, Heidelberg, 2010, pp. 586–596. https://doi.org/10.1007/978-3-642-15844-5_59.

[31] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler, Theory of the HyperVolume Indicator: Optimal μ-distributions and the Choice of the Reference Point, in Proceedings of the Tenth ACM SIGEVO Workshop on Foundations of Genetic Algorithms, ACM, New York, NY, USA, 2009, pp. 87–102. https://doi.org/10.1145/1527125.1527138.

[32] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler, HyperVolume-based multiobjective optimization: Theoretical foundations and practical implications, Theoretical Computer Science **425** (2012), 75–103. https://doi.org/10.1016/j.tcs.2011.03.012.

[33] F. Augustin and Y. Marzouk, NOWPAC: A provably convergent derivative-free nonlinear optimizer with path-augmented constraints, Tech. report, arXiv, 2014. Available at https://arxiv.org/abs/1403.1931.

[34] T. Bäck, Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms, Oxford University Press, New York, NY, USA, 1996. Available at https://dl.acm.org/doi/book/10.5555/229867.

[35] A. Bagirov, N. Karmitsa, and M. Mkel, Introduction to Nonsmooth Optimization: Theory, Practice and Software, Springer Publishing Company, Incorporated, August 2014. Available at https://www.ebook.de/de/product/22426715/adil_bagirov_napsu_karmitsa_marko_m_maekelae_introduction_to_nonsmooth_optimization.html.

[36] V. Beiranvand, W. Hare, and Y. Lucet, Best practices for comparing optimization algorithms, Optimization and Engineering **18** no. 4 (2017), 815–848. https://doi.org/10.1007/s11081-017-9366-1.

[37] N. Beume, C. Fonseca, M. Lopez-Ibanez, L. Paquete, and J. Vahrenhold, On the Complexity of Computing the HyperVolume Indicator, IEEE Transactions on Evolutionary Computation **13** no. 5 (2009), 1075–1082. https://doi.org/10.1109/tevc.2009.2015575.

[38] N. Beume, B. Naujoks, and M. Emmerich, SMS-EMOA: Multiobjective selection based on dominated hypervolume, European Journal of Operational Research **181** no. 3 (2007), 1653–1669. https://doi.org/10.1016/j.ejor.2006.08.008.

[39] B. Beykal, F. Boukouvala, C. Floudas, and E. Pistikopoulos, Optimal design of energy systems using constrained grey-box multi-objective optimization, Computers & Chemical Engineering **116** (2018), 488–502. https://doi.org/https://doi.org/10.1016/j.compchemeng.2018.02.017.

[40] J. Bigeon, S. Le Digabel, and L. Salomon, DMulti-MADS: Mesh adaptive direct multisearch for bound-constrained blackbox multiobjective optimization, Computational Optimization and Applications **79** no. 2 (2021), 301–338. https://doi.org/10.1007/s10589-021-00272-9.

[41] J. Bigeon, S. Le Digabel, and L. Salomon, Handling of constraints in multiobjective blackbox optimization, Tech. Report G-2022-10, Les cahiers du GERAD, 2022. Available at http://www.optimization-online.org/DB_HTML/2022/04/8857.html.

[42] J. Blank and K. Deb, Pymoo: Multi-objective optimization in python, IEEE Access **8** (2020), 89497–89509. https://doi.org/10.1109/access.2020.2990567.

[43] F. Boukouvala, M. F. Hasan, and C. Floudas, Global optimization of general constrained grey-box models: new method and its application to constrained PDEs for pressure swing adsorption, Journal of Global Optimization **67** no. 1 (2017), 3–42. https://doi.org/10.1007/s10898-015-0376-2.

[44] E. Bradford, A. Schweidtmann, and A. Lapkin, Efficient multiobjective optimization employing Gaussian processes, spectral sampling and a genetic algorithm, Journal of Global Optimization **71** no. 2 (2018), 407–438. https://doi.org/10.1007/s10898-018-0609-2.

[45] J. Branke, K. Deb, K. Miettinen, and R. Słowiński, Multiobjective optimization: Interactive and evolutionary approaches, **5252**, Springer, 2008. https://doi.org/10.1007/978-3-540-88908-3.

[46] K. Bringmann and T. Friedrich, Approximating the Volume of unions and intersections of high-dimensional geometric objects, Computational Geometry **43** no. 6 (2010), 601–610. https://doi.org/10.1016/j.comgeo.2010.03.004.

[47] K. Bringmann and T. Friedrich, Tight Bounds for the Approximation Ratio of the HyperVolume Indicator, in Parallel Problem Solving from Nature, PPSN XI (R. Schaefer, C. Cotta, J. Kołodziej, and G. Rudolph, eds.), Springer, Berlin, Heidelberg, 2010, pp. 607–616. Available at https://link.springer.com/chapter/10.1007/978-3-642-15844-5_61.

[48] K. Bringmann and T. Friedrich, Approximation quality of the hyperVolume indicator, Artificial Intelligence **195** (2013), 265–290. https://doi.org/10.1016/j.artint.2012.09.005.

[49] D. Brockhoff, A. Auger, N. Hansen, and T. Tušar, Quantitative Performance Assessment of Multiobjective Optimizers: The Average Runtime Attainment Function, in Evolutionary Multi-Criterion Optimization (H. Trautmann, G. Rudolph, K. Klamroth, O. Schütze, M. Wiecek, Y. Jin, and C. Grimme, eds.), Springer, Cham, 2017, pp. 103–119. https://doi.org/10.1007/978-3-319-54157-0_8.

[50] D. Brockhoff, T. Tran, and N. Hansen, Benchmarking Numerical Multiobjective Optimizers Revisited, in Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, GECCO '15, 2015, pp. 639–646. https://doi.org/10.1145/2739480.2754777.

[51] D. Brockhoff, T. Wagner, and H. Trautmann, On the Properties of the R2 Indicator, in Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation, ACM, New York, NY, USA, 2012, pp. 465–472. https://doi.org/10.1145/2330163.2330230.

[52] D. Brockhoff, T. Wagner, and H. Trautmann, 2 Indicator-Based Multiobjective Search, Evolutionary Computation **23** no. 3 (2015), 369–395. https://doi.org/10.1162/EVCO_a_00135.

[53] C. Brás and A. Custódio, On the use of polynomial models in multiobjective directional direct search, Computational Optimization and Applications **77** no. 3 (2020), 897–918. https://doi.org/10.1007/s10589-020-00233-8.

[54] L. Bueno, A. Friedlander, J. Martínez, and F. Sobral, Inexact restoration method for derivative-free optimization with smooth constraints, SIAM Journal on Optimization **23** no. 2 (2013), 1189–1213. https://doi.org/10.1137/110856253.

[55] A. Bűrmen, J. Olenšek, and T. Tuma, Mesh adaptive direct search with second directional derivative-based hessian update, Computational Optimization and Applications **62** no. 3 (2015), 693–715. https://doi.org/10.1007/s10589-015-9753-5.

[56] X. Cai, H. Sun, and Z. Fan, A diversity indicator based on reference vectors for many-objective optimization, Information Sciences **430-431** (2018), 467–486. https://doi.org/10.1016/j.ins.2017.11.051.

[57] T. Chan, Klee's Measure Problem Made Easy, in IEEE 54th Annual Symposium on Foundations of Computer Science, IEEE, 2013, pp. 410–419. https://doi.org/10.1109/FOCS.2013.51.

[58] S. Cheng, Y. Shi, and Q. Qin, On the Performance Metrics of Multiobjective Optimization, in Advances in Swarm Intelligence (Y. Tan, Y. Shi, and Z. Ji, eds.), Springer, Berlin, Heidelberg, 2012, pp. 504–512. https://doi.org/10.1007/978-3-642-30976-2_61.

[59] F. Clarke, Optimization and Nonsmooth Analysis, John Wiley and Sons, New York, 1983, Reissued in 1990 by SIAM Publications, Philadelphia, as Vol. 5 in the series Classics in Applied Mathematics. Available at http://www.ec-securehost.com/SIAM/CL05.html.

[60] G. Cocchi, M. Lapucci, and P. Mansueto, Pareto front approximation through a multi-objective augmented lagrangian method, EURO Journal on Computational Optimization **9** (2021), 100008. https://doi.org/10.1016/j.ejco.2021.100008.

[61] G. Cocchi, G. Liuzzi, S. Lucidi, and M. Sciandrone, On the convergence of steepest descent methods for multiobjective optimization, Computational Optimization and Applications **77** no. 1 (2020), 1–27. https://doi.org/10.1007/s10589-020-00192-0.

[62] C. Coello and N. Cortés, Solving Multiobjective Optimization Problems Using an Artificial Immune System, Genetic Programming and Evolvable Machines **6** no. 2 (2005), 163–190. https://doi.org/10.1007/s10710-005-6164-x.

[63] Y. Collette and P. Siarry, Optimisation multiobjectif, Eyrolles, 2002. Available at http://goo.gl/s1PHG.

[64] Y. Collette and P. Siarry, Three new metrics to measure the convergence of metaheuristics towards the Pareto frontier and the aesthetic of a set of solutions in biobjective optimization, Computers and Operations Research **32** no. 4 (2005), 773–792. https://doi.org/https://dx.doi.org/10.1016/j.cor.2003.08.017.

[65] Y. Collette and P. Siarry, Multiobjective optimization: principles and case studies, Springer, 2011. https://doi.org/10.1007/978-3-662-08883-8.

[66] A. Conn and S. L. Digabel, Use of quadratic models with mesh-adaptive direct search for constrained black box optimization, Optimization Methods and Software **28** no. 1 (2013), 139–158. https://doi.org/10.1080/10556788.2011.623162.

[67] A. Conn, K. Scheinberg, and L. Vicente, Introduction to Derivative-Free Optimization, MOS-SIAM Series on Optimization, SIAM, Philadelphia, 2009. https://doi.org/10.1137/1.9780898718768.

[68] G. Cornuéjols, J. Peña, and R. Tütüncü, Optimization Methods in Finance, Cambridge University Press, 2018. https://doi.org/10.1017/9781107297340.

[69] P. Côté, C. Audet, N. Amaioua, E. Bigeon, Q. Desreumaux, A. Ihaddadene, Y. Mir, J. Rodriguez, and L. Zéphyr, Planning of the maintenance outages for a set of hydroelectric turbogenerators, Tech. Report CRM-3350, Centre de Recherches Mathématiques, 2015.

[70] A. Custódio, J. Dennis, Jr., and L. Vicente, Using simplex gradients of nonsmooth functions in direct search methods, IMA Journal of Numerical Analysis **28** no. 4 (2008), 770–784. https://doi.org/10.1093/imanum/drn045.

[71] A. Custódio, M. Emmerich, and J. Madeira, Recent Developments in Derivative-Free Multiobjective Optimization, Computational Technology Reviews **5** no. 1 (2012), 1–30. https://doi.org/10.4203/ctr.5.1.

[72] A. Custódio and J. Madeira, GLODS: Global and Local Optimization using Direct Search, Journal of Global Optimization **62** no. 1 (2015), 1–28. https://doi.org/10.1007/s10898-014-0224-9.

[73] A. Custódio and J. Madeira, MultiGLODS: global and local multiobjective optimization using direct search, Journal of Global Optimization **72** no. 2 (2018), 323–345. https://doi.org/10.1007/s10898-018-0618-1.

[74] A. Custódio, J. Madeira, A. Vaz, and L. Vicente, Direct multisearch for multiobjective optimization, SIAM Journal on Optimization **21** no. 3 (2011), 1109–1140. https://doi.org/10.1137/10079731X.

[75] A. Custódio, H. Rocha, and L. Vicente, Incorporating minimum Frobenius norm models in direct search, Computational Optimization and Applications **46** no. 2 (2010), 265–278. https://doi.org/10.1007/s10589-009-9283-0.

[76] A. Custódio and L. Vicente, Using Sampling and Simplex Derivatives in Pattern Search Methods, SIAM Journal on Optimization **18** no. 2 (2007), 537–555. https://doi.org/10.1137/050646706.

[77] A. Custódio, Y. Diouane, R. Garmanjani, and E. Riccietti, Worst-Case Complexity Bounds of Directional Direct-Search Methods for Multiobjective Optimization, Journal of Optimization Theory and Applications **188** no. 1 (2020), 73–93. https://doi.org/10.1007/s10957-020-01781-z.

[78] P. Czyżżak and A. Jaszkiewicz, Pareto simulated annealing–a metaheuristic technique for multiple-objective combinatorial optimization, Journal of Multi-Criteria Decision Analysis **7** no. 1 (1998), 34–47. https://doi.org/10.1002/(sici)1099-1360(199801)7:1<34::aid-mcda161>3.0.co;2-6.

[79] K. Dächert, K. Klamroth, R. Lacour, and D. Vanderpooten, Efficient computation of the search region in multi-objective optimization, European Journal of Operational Research **260** no. 3 (2017), 841–855. https://doi.org/10.1016/j.ejor.2016.05.029.

[80] I. Das and J. Dennis, Jr., Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems, SIAM Journal on Optimization **8** no. 3 (1998), 631–657. https://doi.org/10.1137/S1052623496307510.

[81] C. Davis, Theory of positive linear dependence, American Journal of Mathematics **76** (1954), 733–746. Available at http://www.ams.org/mathscinet-getitem?mr=16:211e.

[82] R. De Leone, M. Gaudioso, and L. Grippo, Stopping criteria for linesearch methods without derivatives, Mathematical Programming **30** (1984), 285–300. Available at http://www.ams.org/mathscinet-getitem?mr=86g:90093.

[83] K. Deb, Multi-Objective Optimization Using Evolutionary Algorithms, John Wiley and Sons, New York, NY, USA, 2001.

[84] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II, in Parallel Problem Solving from Nature PPSN VI (M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. Merelo, and H.-P. H.P. Schwefel, eds.), Springer, Berlin, Heidelberg, 2000, pp. 849–858. https://doi.org/10.1007/3-540-45356-3_83.

[85] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation **6** no. 2 (2002), 182–197. https://doi.org/10.1109/4235.996017.

[86] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, Scalable multi-objective optimization test problems, in Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600), **1**, IEEE, 2002, pp. 825–830 vol.1. https://doi.org/10.1109/CEC.2002.1007032.

[87] K. Deb and S. Tiwari, Omni-optimizer: A generic evolutionary algorithm for single and multi-objective optimization, European Journal of Operational Research **185** no. 3 (2008), 1062–1087. https://doi.org/10.1016/j.ejor.2006.06.042.

[88] K. Deb, Multi-Objective Optimization Using Evolutionary Algorithms, John Wiley & Sons, Inc., New York, NY, USA, 2001.

[89] K. Deb and K. Miettinen, Multiobjective optimization: interactive and evolutionary approaches, **5252**, Springer Science & Business Media, 2008.

[90] S. Dedoncker, W. Desmet, and F. Naets, An adaptive direct multisearch method for black-box multi-objective optimization, Optimization and Engineering (2021). https://doi.org/10.1007/s11081-021-09657-5.

[91] A. Deutz, M. Emmerich, and K. Yang, The Expected R2-Indicator Improvement for Multi-objective Bayesian Optimization, in Evolutionary Multi-Criterion Optimization (K. Deb, E. Goodman, C. Coello,

K. Klamroth, K. Miettinen, S. Mostaghim, and P. Reed, eds.), Springer, Cham, 2019, pp. 359–370. https://doi.org/10.1007/978-3-030-12598-1_29.

[92] F. Di Pierro, S. Khu, D. Savić, and L. Berardi, Efficient multi-objective optimal design of water distribution networks on a budget of simulations using hybrid algorithms, Environmental Modelling & Software **24** no. 2 (2009), 202–213. https://doi.org/https://doi.org/10.1016/j.envsoft.2008.06.008.

[93] E. Dilettoso, S. Rizzo, and N. Salerno, A Weakly Pareto Compliant Quality Indicator, Mathematical and Computational Applications **22** no. 1 (2017). https://doi.org/10.3390/mca22010025.

[94] M. Diniz-Ehrhardt, J. Martinez, and L. Pedroso, Derivative-free methods for nonlinear programming with general lower-level constraints, To appear in Journal of Computational and Applied Mathematics (2011). Available at http://www.optimization-online.org/DB_HTML/2010/06/2663.html.

[95] Y. Diouane, S. Gratton, and L. Vicente, Globally convergent evolution strategies, Mathematical Programming **152** no. 1 (2015), 467–490. https://doi.org/10.1007/s10107-014-0793-x.

[96] Y. Diouane, S. Gratton, and L. Vicente, Globally convergent evolution strategies for constrained optimization, Computational Optimization and Applications **62** no. 2 (2015), 323–346. https://doi.org/10.1007/s10589-015-9747-3.

[97] E. Dolan and J. Moré, Benchmarking optimization software with performance profiles, Mathematical Programming **91** no. 2 (2002), 201–213. https://doi.org/10.1007/s101070100263.

[98] M. Ehrgott, Multicriteria Optimization, Volume 491 of Lecture Notes in Economics and Mathematical Systems, 2nd ed., Springer, Berlin, 2005. Available at https://www.springer.com/gp/book/9783540213987.

[99] A. Eiben and J. Smith, Introduction to Evolutionary Computing, 2nd ed., **53**, Springer, July 2015. Available at https://www.ebook.de/de/product/22721904/a_e_eiben_j_e_smith_introduction_to_evolutionary_computing.html.

[100] M. Emmerich, A. Deutz, J. Kruisselbrink, and P. Shukla, Cone-Based HyperVolume Indicators: Construction, Properties, and Efficient Computation, in

Evolutionary Multi-Criterion Optimization (R. Purshouse, P. Fleming, C. Fonseca, S. Greco, and J. Shaw, eds.), Springer, Berlin, Heidelberg, 2013, pp. 111–127. https://doi.org/10.1007/978-3-642-37140-0_12.

[101] M. Emmerich, K. Yang, A. Deutz, H. Wang, and C. Fonseca, A Multicriteria Generalization of Bayesian Global Optimization, pp. 229–242, Springer, Cham, 2016. https://doi.org/10.1007/978-3-319-29975-4_12.

[102] M. Emmerich, A. Deutz, and J. Kruisselbrink, On Quality Indicators for Black-Box Level Set Approximation, pp. 157–185, Springer, Berlin, Heidelberg, 2013. https://doi.org/10.1007/978-3-642-32726-1_4.

[103] M. Emmerich, A. Deutz, and I. Yevseyeva, On Reference Point Free Weighted HyperVolume Indicators based on Desirability Functions and their Probabilistic Interpretation, Procedia Technology **16** (2014), 532–541. https://doi.org/10.1016/j.protcy.2014.10.001.

[104] J. Falcón-Cardona and C. C. Coello, Indicator-Based Multi-Objective Evolutionary Algorithms: A Comprehensive Survey, ACM Computing Surveys **53** no. 2 (2020), 1–35. https://doi.org/10.1145/3376916.

[105] J. Falcón-Cardona, C. C. Coello, and M. Emmerich, CRI-EMOA: A Pareto-Front Shape Invariant Evolutionary Multi-objective Algorithm, in Evolutionary Multi-Criterion Optimization (K. Deb, E. Goodman, C. C. Coello, K. Klamroth, K. Miettinen, S. Mostaghim, and P. Reed, eds.), Springer, Cham, 2019, pp. 307–318.

[106] J. Falcón-Cardona, M. Emmerich, and C. C. Coello, On the Construction of Pareto-Compliant Quality Indicators, in Proceedings of the Genetic and Evolutionary Computation Conference Companion, ACM, 2019, pp. 2024–2027. https://doi.org/10.1145/3319619.3326902.

[107] H. Fang, M. Rais-Rohani, Z. Liu, and M. Horstemeyer, A comparative study of metamodeling methods for multiobjective crashworthiness optimization, Computers & Structures **83** no. 25-26 (2005), 2121–2136. https://doi.org/https://doi.org/10.1016/j.compstruc.2005.02.025.

[108] A. Farhang-Mehr and S. Azarm, An Information-Theoretic Entropy Metric for Assessing Multi-Objective Optimization Solution Set Quality, Journal of Mechanical Design **125** no. 4 (2004), 655–663. https://doi.org/10.1115/1.1623186.

[109] G. Fasano, G. Liuzzi, S. Lucidi, and F. Rinaldi, A Linesearch-Based Derivative-Free Approach for Nonsmooth Constrained Optimization, SIAM Journal on Optimization **24** no. 3 (2014), 959–992. https://doi.org/10.1137/130940037.

[110] S. Faulkenberg and M. Wiecek, On the quality of discrete representations in multiple objective programming, Optimization and Engineering **11** no. 3 (2009), 423–440. https://doi.org/10.1007/s11081-009-9099-x.

[111] P. Feliot, J. Bect, and E. Vazquez, A Bayesian approach to constrained single- and multi-objective optimization, Journal of Global Optimization **67** no. 1–2 (2017), 97–133. https://doi.org/10.1007/s10898-016-0427-3.

[112] P. Feliot, J. Bect, and E. Vazquez, User Preferences in Bayesian Multi-objective Optimization: The Expected Weighted HyperVolume Improvement Criterion, in Machine Learning, Optimization, and Data Science (G. Nicosia, P. Pardalos, G. Giuffrida, R. Umeton, and V. Sciacca, eds.), Springer, Cham, 2019, pp. 533–544. https://doi.org/10.1007/978-3-030-13709-0_45.

[113] E. Fermi and N. Metropolis, Numerical solution of a minimum problem, Los Alamos Unclassified Report LA–1492, Los Alamos National Laboratory, Los Alamos, USA, 1952.

[114] R. Fletcher and S. Leyffer, Nonlinear programming without a penalty function, Mathematical Programming **Series A, 91** (2002), 239–269. https://doi.org/10.1007/s101070100244.

[115] R. Fletcher, S. Leyffer, and P. Toint, On the global convergence of a filter–SQP algorithm, SIAM Journal on Optimization **13** no. 1 (2002), 44–59. https://doi.org/10.1137/S105262340038081X.

[116] J. Fliege, L. G. Drummond, and B. Svaiter, Newton's method for multiobjective optimization, SIAM Journal on Optimization **20** no. 2 (2009), 602–626. https://doi.org/10.1137/08071692x.

[117] J. Fliege and B. Svaiter, Steepest descent methods for multicriteria optimization, Mathematical Methods of Operations Research **51** no. 3 (2000), 479–494. https://doi.org/https://doi.org/10.1007/s001860000043.

[118] J. Fliege and A. Vaz, A method for constrained multiobjective optimization based on SQP techniques, SIAM Journal on Optimization **26** no. 4 (2016), 2091–2119. https://doi.org/10.1137/15M1016424.

[119] C. Fonseca, V. da Fonseca, and L. Paquete, Exploring the Performance of Stochastic Multiobjective Optimisers with the Second-Order Attainment Function, in Evolutionary Multi-Criterion Optimization (C. Coello, A. H. Aguirre, and E. Zitzler, eds.), Springer, Berlin, Heidelberg, 2005, pp. 250–264. https://doi.org/10.1007/978-3-540-31880-4_18.

[120] C. Fonseca, A. Guerreiro, M. López-Ibáñez, and L. Paquete, On the Computation of the Empirical Attainment Function, in Evolutionary Multi-Criterion Optimization (R. Takahashi, K. Deb, E. Wanner, and S. Greco, eds.), Springer, Berlin, Heidelberg, 2011, pp. 106–120. https://doi.org/10.1007/978-3-642-19893-9_8.

[121] C. Fonseca, L. Paquete, and M. Lopez-Ibanez, An Improved Dimension-Sweep Algorithm for the Hypervolume Indicator, in 2006 IEEE International Conference on Evolutionary Computation, 2006, pp. 1157–1163. https://doi.org/10.1109/cec.2006.1688440.

[122] V. da Fonseca, C. Fonseca, and A. Hall, Inferential Performance Assessment of Stochastic Optimisers and the Attainment Function, in Evolutionary Multi-Criterion Optimization (E. Zitzler, L. Thiele, K. Deb, C.A.C., and D. Corne, eds.), Springer, Berlin, Heidelberg, 2001, pp. 213–225. https://doi.org/10.1007/3-540-44719-9_15.

[123] T. Friedrich, K. Bringmann, T. Voss, and C. Igel, The Logarithmic HyperVolume Indicator, in Proceedings of the 11th Workshop Proceedings on Foundations of Genetic Algorithms, ACM, New York, NY, USA, 2011, pp. 81–92. https://doi.org/10.1145/1967654.1967662.

[124] R. Garmanjani and L. Vicente, Smoothing and worst-case complexity for direct-search methods in nonsmooth optimization, IMA Journal of Numerical Analysis **33** (2013), 1008–1028. https://doi.org/10.1093/imanum/drs027.

[125] M. L. Garneau, Modelling of a solar thermal power plant for benchmarking blackbox optimization solvers, Master's thesis, Polytechnique Montréal, 2015, Available at https://publications.polymtl.ca/1996/. Available at https://publications.polymtl.ca/1996/.

[126] G.Cocchi, G.Liuzzi, A.Papini, and M.Sciandrone, An implicit filtering algorithm for derivative-free multiobjective optimization with box constraints, Computational

Optimization and Applications **69** no. 2 (2018), 267–296. https://doi.org/10.1007/s10589-017-9953-2.

[127] M. Gendreau, J. Potvin, and others, Handbook of Metaheuristics, **2**, Springer, 2019. Available at https://www.ebook.de/de/product/34313206/handbook_of_metaheuristics.html.

[128] D. Ghosh and D. Chakraborty, A direction based classical method to obtain complete Pareto set of multi-criteria optimization problems, Opsearch **52** no. 2 (2015), 340–366. https://doi.org/10.1007/s12597-014-0178-1.

[129] P. Gilmore and C. Kelley, An implicit filtering algorithm for optimization of functions with many local minima, SIAM Journal on Optimization **5** no. 2 (1995), 269–285. https://doi.org/10.1137/0805015.

[130] N. Gould and J. Scott, A Note on Performance Profiles for Benchmarking Software, ACM Transactions on Mathematical Software **43** no. 2 (2016), 1–5. https://doi.org/10.1145/2950048.

[131] S. Gratton, C. W. Royer, L. N. Vicente, and Z. Zhang, Direct search based on probabilistic feasible descent for bound and linearly constrained problems, Computational Optimization and Applications **72** no. 3 (2019), 525–559. https://doi.org/10.1007/s10589-019-00062-4.

[132] S. Gratton, C. Royer, L. Vicente, and Z. Zhang, Direct search based on probabilistic descent, SIAM Journal on Optimization **25** no. 3 (2015), 1515–1541. https://doi.org/10.1137/140961602.

[133] S. Gratton and L. Vicente, A Merit Function Approach for Direct Search, SIAM Journal on Optimization **24** no. 4 (2014), 1980–1998. https://doi.org/10.1137/130917661.

[134] L. Grippo, F. Lampariello, and S. Lucidi, Global convergence and stabilization of unconstrained minimization methods without derivatives, Journal of Optimization Theory and Applications **56** no. 3 (1988), 385–406. Available at http://www.ams.org/mathscinet-getitem?mr=89f:90165.

[135] I. Griva, S. Nash, and A. Sofer, Linear and Nonlinear Optimization, Society for Industrial and Applied Mathematics, 2009.

[136] A. Guerreiro and C. Fonseca, HyperVolume Sharpe-Ratio Indicator: Formalization and First Theoretical Results, in Parallel Problem Solving from Nature – PPSN XIV, 2016, pp. 814–823. https://doi.org/10.1007/978-3-319-45823-6_76.

[137] A. Guerreiro and C. Fonseca, An analysis of the HyperVolume Sharpe-Ratio Indicator, European Journal of Operational Research **283** no. 2 (2020), 614–629. https://doi.org/10.1016/j.ejor.2019.11.023.

[138] Y. Haimes, On a bicriterion formulation of the problems of integrated system identification and system optimization, IEEE Transactions on Systems, Man, and Cybernetics **1** no. 3 (1971), 296–297. https://doi.org/10.1109/TSMC.1971.4308298.

[139] M. Hansen and A. Jaszkiewicz, Evaluating the quality of approximations to the non-dominated set, IMM, Department of Mathematical Modelling, 1998.

[140] D. Hardin and E. Saff, Discretizing manifolds via minimum energy points, Notices of the AMS **51** no. 10 (2004), 1186–1194. Available at https://pdfs.semanticscholar.org/1a89/17bddab57cd22706bbffda13385c090a6f4a.pdf.

[141] D. Hardin and E. Saff, Minimal Riesz energy point configurations for rectifiable d-dimensional manifolds, Advances in Mathematics **193** no. 1 (2005), 174–204. https://doi.org/10.1016/j.aim.2004.05.006.

[142] M. Hasanoglu and M. Dolen, Multi-objective feasibility enhanced particle swarm optimization, Engineering Optimization **50** no. 12 (2018), 2013–2037. https://doi.org/10.1080/0305215X.2018.1431232.

[143] R. Hooke and T. Jeeves, “Direct Search" Solution of Numerical and Statistical Problems, Journal of the Association for Computing Machinery **8** no. 2 (1961), 212–229. https://doi.org/10.1145/321062.321069.

[144] S. Huband, P. Hingston, L. Barone, and L. While, A review of multiobjective test problems and a scalable test problem toolkit, IEEE Transactions on Evolutionary Computation **10** no. 5 (2006), 477–506. https://doi.org/10.1109/TEVC.2005.861417.

[145] H. Ishibuchi, R. Imada, Y. Setoguchi, and Y. Nojima, How to Specify a Reference Point in HyperVolume Calculation for Fair Performance Comparison, Evolutionary Computation **26** no. 3 (2018), 411–440. https://doi.org/10.1162/evco_a_00226.

[146] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima, Difficulties in specifying reference points to calculate the inverted generational distance for many-objective optimization problems, in IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making, 2014, pp. 170–177. https://doi.org/10.1109/mcdm.2014.7007204.

[147] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima, Modified Distance Calculation in Generational Distance and Inverted Generational Distance, in Evolutionary Multi-Criterion Optimization (A. Gaspar-Cunha, C. H. Antunes, and C. Coello, eds.), Springer, Cham, 2015, pp. 110–125. https://doi.org/10.1007/978-3-319-15892-1_8.

[148] J. Jahn, Vector Optimization: Theory, Applications, and Extensions, Springer, Berlin, 2004.

[149] J. Jahn, Introduction to the theory of nonlinear optimization, third ed., Springer, 2007. Available at http://www.bibsonomy.org/bibtex/232e94fe35083c3aa63e38f3d517e21e4/dblp.

[150] A. Jaszkiewicz, Improved quick hyperVolume algorithm, Computers and Operations Research **90** (2018), 72–83. https://doi.org/10.1016/j.cor.2017.09.016.

[151] S. Jiang, Y. Ong, J. Zhang, and L. Feng, Consistencies and Contradictions of Performance Metrics in Multiobjective Optimization, IEEE Transactions on Cybernetics **44** no. 12 (2014), 2391–2404. https://doi.org/10.1109/tcyb.2014.2307319.

[152] S. Jiang, S. Yang, and M. Li, On the use of hypervolume for diversity measurement of Pareto front approximations, in 2016 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, 2016, pp. 1–8. https://doi.org/10.1109/SSCI.2016.7850225.

[153] D. Jones, M. Schonlau, and W. Welch, Efficient Global Optimization of Expensive Black Box Functions, Journal of Global Optimization **13** no. 4 (1998), 455–492. https://doi.org/10.1023/A:1008306431147.

[154] C. Kelley, Implicit Filtering, Society for Industrial and Applied Mathematics, Philadephia, PA, 2011. https://doi.org/10.1137/1.9781611971903.

[155] M. Kidd, R. Lusby, and J. Larsen, Equidistant representations: Connecting coverage and uniformity in discrete biobjective optimization, Computers and Operations Research **117** (2020), 104872. https://doi.org/10.1016/j.cor.2019.104872.

[156] C. Kieslich, F. Boukouvala, and C. Floudas, Optimization of black-box problems using smolyak grids and polynomial approximations, Journal of Global Optimization **71** no. 4 (2018), 845–869. https://doi.org/10.1007/s10898-018-0643-0.

[157] S. Kirkpatrick, C. G. Jr., and M. Vecchi, Optimization by Simulated Annealing, Science **220** no. 4598 (1983), 671–680. Available at http://www.jstor.org/stable/1690046.

[158] J. Knowles, ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems, IEEE Transactions on Evolutionary Computation **10** no. 1 (2006), 50–66. https://doi.org/10.1109/tevc.2005.851274.

[159] J. Knowles and D. Corne, On metrics for comparing nondominated sets, in Evolutionary Computation, **1**, IEEE, Honolulu, HI, 2002, pp. 711–716. https://doi.org/10.1109/CEC.2002.1007013.

[160] T. Kolda, R. Lewis, and V. Torczon, Optimization by direct search: New perspectives on some classical and modern methods, SIAM Review **45** no. 3 (2003), 385–482. https://doi.org/10.1137/S003614450242889.

[161] T. Kolda, R. Lewis, and V. Torczon, A generating set direct search augmented Lagrangian algorithm for optimization with a combination of general and linear constraints, Tech. Report SAND2006-5315, Sandia National Laboratories, USA, 2006.

[162] T. Kolda, R. Lewis, and V. Torczon, Stationarity results for generating set search for linearly constrained optimization, SIAM Journal on Optimization **17** no. 4 (2006), 943–968. https://doi.org/10.1137/S1052623403433638.

[163] R. Lacour, K. Klamroth, and C. Fonseca, A box decomposition algorithm to compute the hyperVolume indicator, Computers and Operations Research **79** (2017), 347–360. https://doi.org/10.1016/j.cor.2016.06.021.

[164] J. Larson, M. Menickelly, and S. Wild, Derivative-free optimization methods, Acta Numerica **28** (2019), 287–404. https://doi.org/10.1017/S0962492919000060.

[165] M. Laumanns, R. Günter, and H. Schwefel, Approximating the Pareto set: Concepts, diversity issues, and performance assessment, Secretary of the SFB 531, 1999. Available at http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.407.1332&rep=rep1&type=pdf.

[166] M. Laumanns, E. Zitzler, and L. Thiele, A Unified Model for Multi-Objective Evolutionary Algorithms with Elitism, in Congress on Evolutionary Computation, **1**, Piscataway, New Jersey, USA, 2000, pp. 46–53. https://doi.org/10.1109/CEC.2000.870274.

[167] S. Le Digabel, Algorithm 909: NOMAD: Nonlinear Optimization with the MADS algorithm, ACM Transactions on Mathematical Software **37** no. 4 (2011), 44:1–44:15. https://doi.org/10.1145/1916461.1916468.

[168] S. Le Digabel and S. Wild, A Taxonomy of Constraints in Simulation-Based Optimization, Tech. Report G-2015-57, Les cahiers du GERAD, 2015. Available at http://www.optimization-online.org/DB_HTML/2015/05/4931.html.

[169] Y. Leung and Y. Wang, U-measure: a quality measure for multiobjective programming, IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans **33** no. 3 (2003), 337–343. https://doi.org/10.1109/TSMCA.2003.817059.

[170] R. Lewis and V. Torczon, A globally convergent augmented Lagrangian pattern search algorithm for optimization with general constraints and simple bounds, SIAM Journal on Optimization **12** no. 4 (2002), 1075–1089. https://doi.org/10.1137/S1052623498339727.

[171] R. Lewis and V. Torczon, A Direct Search Approach to Nonlinear Programming Problems Using an Augmented Lagrangian Method with Explicit Treatment of Linear Constraints, Tech. report, College of William and Mary, 2010. Available at http://citeseerx.ist.psu.edu/viewdoc/summary?Doi=10.1.1.156.8677.

[172] M. Li, S. Yang, and X. Liu, Diversity Comparison of Pareto Front Approximations in Many-Objective Optimization, IEEE Transactions on Cybernetics **44** no. 12 (2014), 2568–2584. https://doi.org/10.1109/TCYB.2014.2310651.

[173] M. Li, S. Yang, and X. Liu, A Performance Comparison Indicator for Pareto Front Approximations in Many-Objective Optimization, in Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, ACM, New York, NY, USA, 2015, pp. 703–710. https://doi.org/10.1145/2739480.2754687.

[174] M. Li and X. Yao, Dominance Move: A Measure of Comparing Solution Sets in Multiobjective Optimization, Tech. Report 1702.00477, arXiv, 2017. Available at https://arxiv.org/abs/1702.00477.

[175] M. Li and X. Yao, Quality Evaluation of Solution Sets in Multiobjective Optimisation: A Survey, ACM Computing Surveys **52** no. 2 (2019), 26:1–26:38. https://doi.org/10.1145/3300148.

[176] M. Li, J. Zheng, and G. Xiao, Uniformity assessment for evolutionary multi-objective optimization, in IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), 2008, pp. 625–632. https://doi.org/10.1109/CEC.2008.4630861.

[177] G. Liuzzi and S. Lucidi, A derivative-free algorithm for inequality constrained nonlinear programming via smoothing of an ℓ_∞ penalty function, SIAM Journal on Optimization **20** no. 1 (2009), 1–29. https://doi.org/10.1137/070711451.

[178] G. Liuzzi, S. Lucidi, and F.Rinaldi, Derivative-free methods for bound constrained mixed-integer optimization, Computational Optimization and Applications **53** no. 2 (2012), 505–526. https://doi.org/10.1007/s10589-011-9405-3.

[179] G. Liuzzi, S. Lucidi, and F. Rinaldi, Derivative-Free Methods for Mixed-Integer Constrained Optimization Problems, Journal of Optimization Theory and Applications **164** no. 3 (2015), 933–965. https://doi.org/10.1007/s10957-014-0617-4.

[180] G. Liuzzi, S. Lucidi, and F. Rinaldi, A Derivative-Free Approach to Constrained Multiobjective Nonsmooth Optimization, SIAM Journal on Optimization **26** no. 4 (2016), 2744–2774. https://doi.org/10.1137/15M1037810.

[181] G. Liuzzi, S. Lucidi, and F. Rinaldi, An algorithmic framework based on primitive directions and nonmonotone line searches for black-box optimization problems with integer variables, Mathematical Programming Computation **12** no. 4 (2020), 673–702. https://doi.org/10.1007/s12532-020-00182-7.

[182] G. Liuzzi, S. Lucidi, F. Rinaldi, and L. Vicente, Trust-region methods for the derivative-free optimization of nonsmooth black-box functions, SIAM Journal on Optimization **29** no. 4 (2019), 3012–3035. https://doi.org/10.1137/19M125772X.

[183] G. Liuzzi, S. Lucidi, and M. Sciandrone, Sequential penalty derivative-free methods for nonlinear constrained optimization, SIAM Journal on Optimization **20** no. 5 (2010), 2614–2635. https://doi.org/10.1137/090750639.

[184] G. Lizarraga-Lizarraga, A. Hernandez-Aguirre, and S. Botello-Rionda, G-Metric: An M-ary Quality Indicator for the Evaluation of Non-dominated Sets, in

Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, ACM, New York, NY, USA, 2008, pp. 665–672. https://doi.org/10.1145/1389095.1389227.

[185] E. Lopez and C. Coello, IGD+-EMOA: A multi-objective evolutionary algorithm based on IGD+, in 2016 IEEE Congress on Evolutionary Computation (CEC), 2016, pp. 999–1006. https://doi.org/10.1109/CEC.2016.7743898.

[186] H. Markowitz, Portfolio selection, Journal of Finance **7** no. 1 (1952), 77–91.

[187] L. Martí, J. García, A. Berlanga, and J. Molina, A stopping criterion for multi-objective optimization evolutionary algorithms, Information Sciences **367-368** (2016), 700–718. https://doi.org/10.1016/j.ins.2016.07.025.

[188] H. Meng, X. Zhang, and S. Liu, New Quality Measures for Multiobjective Programming, in Advances in Natural Computation (L. Wang, K. Chen, and Y. Ong, eds.), Springer, Berlin, Heidelberg, 2005, pp. 1044–1048. https://doi.org/10.1007/11539117_143.

[189] A. Messac and C. Mattson, Normal Constraint Method with Guarantee of Even Representation of Complete Pareto Frontier, AIAA Journal **42** no. 10 (2004), 2101–2111. https://doi.org/10.2514/1.8977.

[190] K. Miettinen, Nonlinear Multiobjective Optimization, 1999.

[191] J. Mockus, V. Tiesis, and A. Zilinskas, The application of Bayesian methods for seeking the extremum, pp. 117–129, North-Holand, 1978.

[192] J. Moré and S. Wild, Benchmarking Derivative-Free Optimization Algorithms, SIAM Journal on Optimization **20** no. 1 (2009), 172–191. https://doi.org/10.1137/080724083.

[193] V. Morovati, H. Basirzadeh, and L. Pourkarimi, Quasi-Newton methods for multiobjective optimization problems, 4OR **16** no. 3 (2017), 261–294. https://doi.org/10.1007/s10288-017-0363-1.

[194] V. Morovati and L. Pourkarimi, Extension of zoutendijk method for solving constrained multiobjective optimization problems, European Journal of Operational Research **273** no. 1 (2019), 44–57. https://doi.org/https://doi.org/10.1016/j.ejor.2018.08.018.

[195] S. Mostaghim and J. Teich, A New Approach on Many Objective Diversity Measurement, in Practical Approaches to Multi-Objective Optimization (J. Branke, K. Deb, K. Miettinen, and R. Steuer, eds.), Dagstuhl Seminar Proceedings no. 04461, Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, Dagstuhl, Germany, 2005. Available at http://drops.dagstuhl.de/opus/volltexte/2005/254.

[196] J. Müller, SOCEMO: Surrogate Optimization of Computationally Expensive Multiobjective Problems, INFORMS Journal on Computing **29** no. 4 (2017), 581–596. https://doi.org/10.1287/ijoc.2017.0749.

[197] J. Müller and M. Day, Surrogate Optimization of Computationally Expensive Black-Box Problems with Hidden Constraints, INFORMS Journal on Computing **31** no. 4 (2019), 689–702. https://doi.org/10.1287/ijoc.2018.0864.

[198] J. Nelder and R. Mead, A simplex method for function minimization, The Computer Journal **7** no. 4 (1965), 308–313. https://doi.org/10.1093/comjnl/7.4.308.

[199] Y. Nesterov and A. Nemirovskii, Interior-Point Polynomial Algorithms in Convex Programming, Society for Industrial and Applied Mathematics, 1994. https://doi.org/10.1137/1.9781611970791.

[200] J. Niebling and G. Eichfelder, A branch–and–bound-based algorithm for nonconvex multiobjective optimization, SIAM Journal on Optimization **29** no. 1 (2019), 794–821. https://doi.org/10.1137/18M1169680.

[201] J. Nocedal and S. Wright, Numerical Optimization, second ed., Springer Series in Operations Research and Financial Engineering, Springer, Berlin, 2006. Available at http://www.springer.com/mathematics/book/978-0-387-30303-1.

[202] N.Ploskas and N. Sahinidis, Review and comparison of algorithms and software for mixed-integer derivative-free optimization, Journal of Global Optimization **82** no. 3 (2021), 433–462. https://doi.org/10.1007/s10898-021-01085-0.

[203] L. Nuñez, R. Regis, and K. Varela, Accelerated Random Search for constrained global optimization assisted by Radial Basis Function surrogates, Journal of Computational and Applied Mathematics **340** (2018), 276–295. https://doi.org/10.1016/j.cam.2018.02.017.

[204] T. Okabe, Y. Jin, and B. Sendhoff, A critical survey of performance indices for multi-objective optimisation, in Evolutionary Computation, **2**, Canberra, Australia, 2003, pp. 878–885. https://doi.org/10.1109/CEC.2003.1299759.

[205] P. Pardalos, A. Žilinska, and J. Žilinskas, Non-Convex Multi-Objective Optimization, Springer International Publishing, 2017. https://doi.org/10.1007/978-3-319-61007-8.

[206] A. Pascoletti and P. Serafini, Scalarizing vector optimization problems, Journal of Optimization Theory and Applications **42** no. 4 (1984), 499–524. https://doi.org/10.1007/bf00934564.

[207] R. Poli, J. Kennedy, and T. Blackwell, Particle swarm optimization, Swarm Intelligence **1** no. 1 (2007), 33–57. https://doi.org/10.1007/s11721-007-0002-0.

[208] S. Prinz, J. Thomann, G. Eichfelder, T. Boeck, and J. Schumacher, Expensive multi-objective optimization of electromagnetic mixing in a liquid metal, Optimization and Engineering **22** no. 2 (2020), 1065–1089. https://doi.org/10.1007/s11081-020-09561-4.

[209] L. L. Pérez and L. Prudente, Nonlinear Conjugate Gradient Methods for Vector Optimization, SIAM Journal on Optimization **28** no. 3 (2018), 2690–2720. https://doi.org/10.1137/17M1126588.

[210] S. Qu, M. Goh, and F. Chan, Quasi-newton methods for solving multiobjective optimization, Operations Research Letters **39** no. 5 (2011), 397–399. https://doi.org/https://doi.org/10.1016/j.orl.2011.07.008.

[211] S. Qu, M. Goh, and B. Liang, Trust region methods for solving multiobjective optimisation, Optimization Methods and Software **28** no. 4 (2013), 796–811. https://doi.org/10.1080/10556788.2012.660483.

[212] R. Regis, Constrained optimization by radial basis function interpolation for high-dimensional expensive black-box problems with infeasible initial points, Engineering Optimization **46** no. 2 (2014), 218–243. https://doi.org/10.1080/0305215X.2013.765000.

[213] R. Regis, Multi-objective constrained black-box optimization using radial basis function surrogates, Journal of Computational Science **16** (2016), 140–155. https://doi.org/https://doi.org/10.1016/j.jocs.2016.05.013.

[214] R. Regis, On the convergence of adaptive stochastic search methods for constrained and multi-objective black-box optimization, Journal of Optimization Theory and Applications **170** no. 3 (2016), 932–959. https://doi.org/10.1007/s10957-016-0977-z.

[215] R. Regis, On the properties of positive spanning sets and positive bases, Optimization and Engineering **17** no. 1 (2016), 229–262. https://doi.org/10.1007/s11081-015-9286-x.

[216] R. Regis and S. Wild, CONORBIT: constrained optimization by radial basis function interpolation in trust regions, Optimization Methods and Software **32** no. 3 (2017), 552–580. https://doi.org/10.1080/10556788.2016.1226305.

[217] L. Rios and N. Sahinidis, Derivative-free optimization: a review of algorithms and comparison of software implementations, Journal of Global Optimization **56** no. 3 (2013), 1247–1293. https://doi.org/10.1007/s10898-012-9951-y.

[218] N. Riquelme, C. V. Lücken, and B. Baran, Performance metrics in multi-objective optimization, in Latin American Computing Conference, IEEE, 2015, pp. 1–11. https://doi.org/10.1109/CLEI.2015.7360024.

[219] R. Rockafellar, Augmented Lagrange Multiplier Functions and Duality in Nonconvex Programming, SIAM Journal on Control **12** no. 2 (1974), 268–285. https://doi.org/10.1137/0312021.

[220] G. Rudolph, O. Schütze, C. Grimme, C. Domínguez-Medina, and H. Trautmann, Optimal averaged Hausdorff archives for bi-objective problems: theoretical and numerical results, Computational Optimization and Applications **64** no. 2 (2016), 589–618. https://doi.org/10.1007/s10589-015-9815-8.

[221] L. Russo and A. Francisco, Quick HyperVolume, IEEE Transactions on Evolutionary Computation **18** no. 4 (2014), 481–502. https://doi.org/10.1109/TEVC.2013.2281525.

[222] L. Russo and A. Francisco, Extending quick hyperVolume, Journal of Heuristics **22** no. 3 (2016), 245–271. https://doi.org/10.1007/s10732-016-9309-6.

[223] J. h. Ryu and S. Kim, A derivative free trust region method for biobjective optimization, SIAM Journal on Optimization **24** no. 1 (2014), 334–362. https://doi.org/10.1137/120864738.

[224] P. Sampaio and P. Toint, A derivative-free trust-funnel method for equality-constrained nonlinear optimization, Computational Optimization and Applications **61** no. 1 (2015), 25–49. https://doi.org/10.1007/s10589-014-9715-3.

[225] S. Sayin, Measuring the quality of discrete representations of efficient sets in multiple objective mathematical programming, Mathematical Programming **87** no. 3 (2000), 543–560. https://doi.org/10.1007/s101070050011.

[226] J. Schott, Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization, Tech. report, Air force Institute Of Tech Wright-Patterson AFB OH, 1995. Available at http://hdl.handle.net/1721.1/11582.

[227] O. Schutze, X. Esquivel, A. Lara, and C. Coello, Using the Averaged Hausdorff Distance as a Performance Measure in Evolutionary Multiobjective Optimization, IEEE Transactions on Evolutionary Computation **16** no. 4 (2012), 504–522. https://doi.org/10.1109/TEVC.2011.2161872.

[228] Y. Sergeyev, D. Kvasov, and M. Mukhametzhanov, Operational zones for comparing metaheuristic and deterministic one-dimensional global optimization algorithms, Mathematics and Computers in Simulation **141** (2017), 96–109. https://doi.org/10.1016/j.matcom.2016.05.006.

[229] Y. Sergeyev, D. Kvasov, and M. Mukhametzhanov, On the efficiency of nature-inspired metaheuristics in expensive global optimization with limited budget, Scientific Reports **8** no. 1 (2018), 453. https://doi.org/10.1038/s41598-017-18940-4.

[230] B. Shahriari, K. Swersky, Z. Wang, R. Adams, and N. D. Freitas, Taking the human out of the loop: A review of Bayesian optimization, Proceedings of the IEEE **104** no. 1 (2015), 148–175. https://doi.org/10.1109/JPROC.2015.2494218.

[231] C. Shannon, A mathematical theory of communication, ACM SIGMOBILE mobile computing and communications review **5** no. 1 (2001), 3–55. https://doi.org/10.1145/584091.584093.

[232] S. Sharma and G. Rangaiah, Multi-Objective Optimization Applications in Chemical Engineering, ch. 3, pp. 35–102, John Wiley & Sons, Ltd, 2013. https://doi.org/10.1002/9781118341704.ch3.

[233] O. Shir, M. Preuss, B. Naujoks, and M. Emmerich, Enhancing Decision Space Diversity in Evolutionary Multiobjective Algorithms, in

Evolutionary Multi-Criterion Optimization (M. Ehrgott, C. Fonseca, X. Gandibleux, J.-K. Hao, and M. Sevaux, eds.), Springer, Berlin, Heidelberg, 2009, pp. 95–109. https://doi.org/10.1007/978-3-642-01020-0_12.

[234] P. Shukla, N. Doll, and H. Schmeck, A Theoretical Analysis of Volume Based Pareto Front Approximations, in Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation, ACM, New York, NY, USA, 2014, pp. 1415–1422. https://doi.org/10.1145/2576768.2598348.

[235] R. Slowinski and J. Teghem, Stochastic Versus Fuzzy Approaches to Multiobjective Mathematical Programming under Uncertainty, Springer, 1990. https://doi.org/10.1007/978-94-009-2111-5.

[236] S.Qu, C. Liu, M. Goh, Y. Li, and Y. Ji, Nonsmooth multiobjective programming with quasi-newton methods, European Journal of Operational Research **235** no. 3 (2014), 503–510. https://doi.org/10.1016/j.ejor.2014.01.022.

[237] E.-G. Talbi, Metaheuristics: from design to implementation, **74**, John Wiley & Sons, June 2009. Available at https://www.ebook.de/de/product/8201863/talbi_el_ghazali_talbi_metaheuristics.html.

[238] B. Talgorn, C. Audet, M. Kokkolaras, and S. Le Digabel, Locally weighted regression models for surrogate-assisted design optimization, Optimization and Engineering **19** no. 1 (2018), 213–238. https://doi.org/10.1007/s11081-017-9370-5.

[239] K. Tan, T. Lee, and E. Khor, Evolutionary Algorithms for Multi-Objective Optimization: Performance Assessments and Comparisons, Artificial Intelligence Review **17** no. 4 (2002), 251–290. https://doi.org/10.1023/A:1015516501242.

[240] S. Tavares, C. Brás, A. Custódio, V. Duarte, and P. Medeiros, Parallel Strategies for Direct Multisearch, Tech. report, Arxiv, 2021. Available at http://arxiv.org/abs/2105.03000.

[241] J. Thomann, A trust region approach for multi-objective heterogeneous optimization, Ph.D. thesis, Ilmenau, March 2019, Technische Universität Ilmenau, Dissertation, 2019. Available at https://www.db-thueringen.de/receive/dbt_mods_00038302.

[242] J. Thomann and G. Eichfelder, A Trust-Region Algorithm for Heterogeneous Multiobjective Optimization, SIAM Journal on Optimization **29** no. 2 (2019), 1017–1047. https://doi.org/10.1137/18M1173277.

[243] Y. Tian, X. Zhang, R. Cheng, and Y. Jin, A multi-objective evolutionary algorithm based on an enhanced inverted generational distance metric, in 2016 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2016, pp. 5222–5229. https://doi.org/10.1109/cec.2016.7748352.

[244] V. Torczon, On the convergence of pattern search algorithms, SIAM Journal on Optimization **7** no. 1 (1997), 1–25. https://doi.org/10.1137/S1052623493250780.

[245] H. Trautmann, T. Wagner, and D. Brockhoff, R2-EMOA: Focused Multiobjective Search Using R2-Indicator-Based Selection, in Learning and Intelligent Optimization (G. Nicosia and P. Pardalos, eds.), Springer, Berlin, Heidelberg, 2013, pp. 70–74. https://doi.org/10.1007/978-3-642-44973-4_8.

[246] T. Ulrich, J. Bader, and L. Thiele, Defining and Optimizing Indicator-Based Diversity Measures in Multiobjective Search, in Parallel Problem Solving from Nature, PPSN XI (R. Schaefer, C. Cotta, J. Kołodziej, and G. Rudolph, eds.), Springer, Berlin, Heidelberg, 2010, pp. 707–717. https://doi.org/10.1007/978-3-642-15844-5_71.

[247] T. Ulrich, J. Bader, and E. Zitzler, Integrating Decision Space Diversity into HyperVolume-based Multiobjective Search, in Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, ACM, New York, NY, USA, 2010, pp. 455–462. https://doi.org/10.1145/1830483.1830569.

[248] B. Van Dyke and T. Asaki, Using QR Decomposition to Obtain a New Instance of Mesh Adaptive Direct Search with Uniformly Distributed Polling Directions, Journal of Optimization Theory and Applications **159** no. 3 (2013), 805–821. https://doi.org/10.1007/s10957-013-0356-y.

[249] D. Van Veldhuizen, Multiobjective evolutionary algorithms: classifications, analyses, and new innovations, Tech. report, School of Engineering of the Air Force Institute of Technology, Dayton, Ohio, 1999. Available at http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.42.1823.

[250] D. Van Veldhuizen and G. Lamont, On measuring multiobjective evolutionary algorithm performance, in Proceedings of the 2000 Congress on Evolutionary Computation, **1**, 2000, pp. 204–211. https://doi.org/10.1109/CEC.2000.870296.

[251] A. Vargas and J. Bogoya, A Generalization of the Averaged Hausdorff Distance, Computación y Sistemas **22** no. 2 (2018), 331–345. https://doi.org/10.13053/CyS-22-2-2950.

[252] L. N. Vicente, Worst case complexity of direct search, EURO Journal on Computational Optimization **1** no. 1 (2013), 143–153. https://doi.org/10.1007/s13675-012-0003-7.

[253] L. Vicente and A. Custódio, Analysis of direct searches for discontinuous functions, Mathematical Programming **133** no. 1-2 (2012), 299–325. https://doi.org/10.1007/s10107-010-0429-8.

[254] T. Wagner, H. Trautmann, and D. Brockhoff, Preference Articulation by Means of the R2 Indicator, in Evolutionary Multi-Criterion Optimization (R. Purshouse, J. Fleming, C. Fonseca, S. Greco, and J. Shaw, eds.), Springer, Berlin, Heidelberg, 2013, pp. 81–95. https://doi.org/10.1007/978-3-642-37140-0_10.

[255] T. Wagner, H. Trautmann, and L. Martí, A Taxonomy of Online Stopping Criteria for Multi-Objective Evolutionary Algorithms, in Evolutionary Multi-Criterion Optimization (R. Takahashi, K. Deb, E. Wanner, and S. Greco, eds.), Springer, Berlin, Heidelberg, 2011, pp. 16–30. https://doi.org/10.1007/978-3-642-19893-9_2.

[256] T. Wagner, H. Trautmann, and B. Naujoks, OCD: Online Convergence Detection for Evolutionary Multi-Objective Algorithms Based on Statistical Testing, in Evolutionary Multi-Criterion Optimization (M. Ehrgott, C. Fonseca, X. Gandibleux, J.-K. Hao, and M. Sevaux, eds.), Springer, Berlin, Heidelberg, 2009, pp. 198–215. https://doi.org/10.1007/978-3-642-01020-0_19.

[257] W. Wang, T. Akhtar, and C. Shoemaker, Integrating ε-dominance and rbf surrogate optimization for solving computationally expensive many-objective optimization problems, Journal of Global Optimization (2021). https://doi.org/10.1007/s10898-021-01019-w.

[258] L. While, L. Bradstreet, and L. Barone, A Fast Way of Calculating Exact HyperVolumes, IEEE Transactions on Evolutionary Computation **16** no. 1 (2012), 86–95. https://doi.org/10.1109/TEVC.2010.2077298.

[259] M. Wiecek, M. Ehrgott, and A. Engau, Continuous multiobjective programming, in Multiple Criteria Decision Analysis, Springer New York, 2016, pp. 739–815. https://doi.org/10.1007/978-1-4939-3094-4_18.

[260] A. Wierzbicki, On the completeness and constructiveness of parametric characterizations to vector optimization problems, Operations-Research-Spektrum **8** no. 2 (1986), 73–87. https://doi.org/10.1007/BF01719738.

[261] S. Wild, R. Regis, and C. Shoemaker, ORBIT: Optimization by Radial Basis Function Interpolation in Trust-Regions, SIAM Journal on Scientific Computing **30** no. 6 (2008), 3197–3219.

[262] S. Wild and C. Shoemaker, Global convergence of radial basis function trust region derivative-free algorithms, SIAM J. Optimization **21** no. 3 (2011), 761–781. https://doi.org/10.1137/09074927X.

[263] J. Wu and S. Azarm, Metrics for Quality Assessment of a Multiobjective Design Optimization Solution Set, Journal of Mechanical Design **123** no. 1 (2000), 18–25. https://doi.org/10.1115/1.1329875.

[264] W. Xia and C. Shoemaker, GOPS: efficient RBF surrogate global optimization algorithm with high dimensions and many parallel processors including application to multimodal water quality PDE model calibration, Optimization and Engineering **22** no. 4 (2020), 2741–2777. https://doi.org/10.1007/s11081-020-09556-1.

[265] I. Yevseyeva, A. Guerreiro, M. Emmerich, and C. Fonseca, A Portfolio Optimization Approach to Selection in Multiobjective Evolutionary Algorithms, in Parallel Problem Solving from Nature – PPSN XIII, 2014, pp. 672–681. https://doi.org/10.1007/978-3-319-10762-2_66.

[266] J. Yuan, H.-L. Liu, Y.-S. Ong, and Z. He, Indicator-based Evolutionary Algorithm for Solving Constrained Multi-objective Optimization Problems, IEEE Transactions on Evolutionary Computation (2021), 1–1. https://doi.org/10.1109/TEVC.2021.3089155.

[267] S. Zapotecas-Martínez and C. C. Coello, MONSS: A multi-objective nonlinear simplex search approach, Engineering Optimization **48** no. 1 (2015), 16–38. https://doi.org/10.1080/0305215x.2014.992889.

[268] D. Zhan and H. Xing, Expected improvement for expensive optimization: a review, Journal of Global Optimization **78** no. 3 (2020), 507–544. https://doi.org/10.1007/s10898-020-00923-x.

[269] R. Zhang and D. Golovin, Random hypervolume scalarizations for provable multi-objective black box optimization, in Proceedings of the 37th International Conference on Machine Learning, Proceedings of Machine Learning Research **119**, 2020, pp. 11096–11105. Available at http://proceedings.mlr.press/v119/zhang20i.html.

[270] J. Zhao, L. Jiao, F. Liu, V. Fernandes, I. Yevseyeva, S. Xia, and M. Emmerich, 3D fast convex-hull-based evolutionary multiobjective optimization algorithm, Applied Soft Computing **67** (2018), 322–336. https://doi.org/10.1016/j.asoc.2018.03.005.

[271] K. Zheng, R. Yang, H. Xu, and J. Hu, A new distribution metric for comparing Pareto optimal solutions, Structural and Multidisciplinary Optimization **55** no. 1 (2017), 53–62. https://doi.org/10.1007/s00158-016-1469-3.

[272] A. Zhou, A. Jin, Q. Zhang, B. Sendhoff, and E. Tsang, Combining Model-based and Genetics-based Offspring Generation for Multi-objective Optimization Using a Convergence Criterion, in IEEE International Conference on Evolutionary Computation, IEEE, 2006, pp. 892–899. https://doi.org/10.1109/CEC.2006.1688406.

[273] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. Suganthan, and Q. Zhang, Multiobjective evolutionary algorithms: A survey of the state of the art, Swarm and Evolutionary Computation **1** no. 1 (2011), 32–49. https://doi.org/https://doi.org/10.1016/j.swevo.2011.03.001.

[274] E. Zitzler, Evolutionary algorithms for multiobjective optimization : methods and applications, Ph.D. thesis, Institute of Technology, Zurich, Swiss Federal, 1999.

[275] E. Zitzler, D. Brockhoff, and L. Thiele, The HyperVolume Indicator Revisited: On the Design of Pareto-compliant Indicators Via Weighted Integration, in Evolutionary Multi-Criterion Optimization (S. Obayashi, K. Deb, C. Poloni,

T. Hiroyasu, and T. Murata, eds.), Springer, Berlin, Heidelberg, 2007, pp. 862–876. https://doi.org/10.1007/978-3-540-70928-2_64.

[276] E. Zitzler, K. Deb, and L. Thiele, Comparison of Multiobjective Evolutionary Algorithms: Empirical Results, Evolutionary Computation **8** no. 2 (2000), 173–195. https://doi.org/10.1162/106365600568202.

[277] E. Zitzler, J. Knowles, and L. Thiele, Quality assessment of Pareto set approximations, in Multiobjective Optimization, Springer, Berlin, Heidelberg, 2008, pp. 373–404. https://doi.org/10.1007/978-3-540-88908-3_14.

[278] E. Zitzler and S. Künzli, Indicator-Based Selection in Multiobjective Search, in Parallel Problem Solving from Nature - PPSN VIII (X. Yao, E. Burke, J. Lozano, J. Smith, J. Merelo-Guervós, J. Bullinaria, J. Rowe, P. Tiňo, A. Kabán, and H.-S. Schwefel, eds.), Springer, Berlin, Heidelberg, 2004, pp. 832–842. https://doi.org/10.1007/978-3-540-30217-9_84.

[279] E. Zitzler and L. Thiele, Multiobjective optimization using evolutionary algorithms – A comparative case study, in Parallel Problem Solving from Nature – PPSN V (A. Eiben, T. Bäck, M. Schoenauer, and H. Schwefel, eds.), Springer, Berlin, Heidelberg, 1998, pp. 292–301. https://doi.org/10.1007/BFb0056872.

[280] E. Zitzler, L. Thiele, M. Laumanns, C. Fonseca, and V. da Fonseca, Performance assessment of multiobjective optimizers: an analysis and review, IEEE Transactions on Evolutionary Computation **7** no. 2 (2003), 117–132. https://doi.org/10.1109/tevc.2003.810758.