

Landscape Analysis and Heuristic Search for Multi-objective Optimization

Arnaud Liefoghe
Habilitation à diriger des recherches



Date de soutenance : 15 juin 2022

Jury

Laurence Duchien, Université de Lille, examinatrice, présidente du jury

Gabriela Ochoa, University of Stirling, rapporteure

Frédéric Saubion, Université d'Angers, rapporteur

Christine Solnon, INSA Lyon, examinatrice

Thomas Stützle, Université Libre de Bruxelles, examinateur

El-Ghazali Talbi, Université de Lille, garant

Daniel Vanderpooten, Université Paris-Dauphine, rapporteur

© ⓘ ⓘ Arnaud Liefoghe. **Landscape Analysis and Heuristic Search for Multi-objective Optimization**. Habilitation thesis, Univ Lille, 2022.

arnaud.liefoghe@univ-lille.fr | <http://crystal.univ-lille.fr/~liefoghe>

Acknowledgements

First of all, I would like to thank the committee members for agreeing to take part in the jury: Gabriela Ochoa, Frédéric Saubion and Daniel Vanderpooten for reviewing the manuscript, Christine Solnon and Thomas Stütze for being external examiners, Laurence Duchien for chairing the committee, and El-Ghazali Talbi for being the guarantor of the habilitation. It was a great honor to count you among the jury members, thank you very much for your enthusiastic feedback and your insightful questions.

I would also like to thank my colleagues from the computer science department, as well as from the CRISAL and Inria research centers, in particular to all past and current members of the Bonus research group, including Nouredine, Bilel, El-Ghazali and Omar. Thanks also to all my (ex) colleagues from the University of Coimbra, from the JFLI and the University of Tokyo, and from my “home away from home” at Shinshu University in Nagano.

A big thanks as well to all the students and young researchers I had or I have the chance to co-supervise, in particular Geoffrey, Nicolas and Raphaël in Lille, Alexandre between Lille and Coimbra, and Fabio, Saúl, Martin, Miyako and Hugo between Lille and Nagano. I am very grateful to you all. It was great seeing you grow, and I wish you the best for the future.

I would not be complete without mentioning my past and present collaborators. A particularly huge thanks to Bilel and Sébastien, I cannot say enough how important their presence is throughout this manuscript. Special thanks also to Luís in Coimbra, and to Hernan-sensei and Tanaka-sensei in Nagano, our connection and your tremendous support were essential in my evolution. I do not forget the other collaborators with whom I had the chance to share part of the journey: Adrien, Matthieu, Jin-Kao, Manuel, Roberto, Darrell, Qingfu ... among many others!

To all of you, I am well aware that my research is the outcome of a collective effort, and this manuscript would not be the same without your valuable inputs and precious contributions. I was very lucky to meet you along the way, and I am sure this is only the beginning of a long friendship.

Last but not least, I would like to thank my friends and my family. To my parents, my sister and her family: thank you for your continuous support, I will never be able to say how grateful I am for all you have done for me. To my daughters: you are the sunshine of my life and I will always love you. To my wife: thank you very much for your patience and your daily help. You are the best teammate anyone could dream of, and whether you want it or not, you belong in everything I do — and this includes several places in this manuscript ;-)

Contents

Acknowledgements	iii
Contents	v
1 Introduction	1
1.1 Context	1
1.2 Overview	2
1.3 Supervision	3
1.4 Collaborations	3
1.5 Outline	5
2 Multi-objective Optimization	7
2.1 Definitions	7
2.2 Problems	8
2.3 Algorithms	10
2.4 Performance	13
3 Foundations of Multi-objective Landscapes and Search	15
3.1 Motivations	15
3.2 Features	17
3.3 Features vs. Performance	24
3.4 Scaling to Large Landscapes	28
3.5 Algorithm Selection	31
3.6 Further Contributions	33
3.7 Outlook	44
4 Set- and Indicator-based Multi-objective Search	49
4.1 Motivations	49
4.2 Quality Indicators	51
4.3 Local Optimal Sets	58
4.4 Further Contributions	66
4.5 Outlook	72
5 Efficient Multi-objective Search	77
5.1 Motivations	77
5.2 Decomposition	80
5.3 Parallel Search	81
5.4 Surrogate-assisted Search	89
5.5 Further Contributions	96
5.6 Outlook	102
6 Towards Landscape-aware Massive Optimization	105
Bibliography	109

Introduction

My journey in academia began as a Master student at the — back then — Université de Lille 1, when I had the opportunity to join the Dolphin research group for my Master thesis. After graduating, I pursued my research as a PhD student in the same group for three years, and I obtained my PhD degree in 2009. In 2010, I was a Postdoctoral Researcher with the University of Coimbra in Portugal. The same year, I was recruited as a Maître de Conférences (Associate Professor) with the — now — University of Lille. I am currently a member of the Bonus research group at the CRIStAL research center (Univ. Lille, CNRS, Centrale Lille, UMR 9189) and the Inria Lille-Nord Europe research center. In 2020, I benefited from a six-month CNRS sabbatical at the Japanese-French Laboratory for Informatics (JFLI, CNRS, IRL 3527), and I was an Invited Professor with the University of Tokyo in Japan. After a history of about ten years of collaboration and numerous exchanges and visits, I have also been appointed as a Collaborative Professor (honorary position) with Shinshu University in Japan since 2021.

This habilitation thesis provides a survey of my scientific work since the end of my PhD thesis. My research activities deal with the foundations, the design and the analysis of stochastic local search heuristic algorithms, with a particular interest on multi-objective optimization and landscape analysis.

1.1 Context	1
1.2 Overview	2
1.3 Supervision	3
1.4 Collaborations	3
1.5 Outline	5

1.1 Context

Solving an optimization problem consists in finding the best solution(s) with respect to one or multiple objectives. In *multi-objective optimization*, two or more objectives are to be optimized simultaneously. Such problems may occur in many real-world applications as common as finding the optimal path between two geographic locations, where one may aim at reducing the travel time while minimizing the travel cost. In this context, a decision maker might be interested in identifying the set of optimal trade-offs between the objectives, among which they could then select the solution that best fits their preferences. Such a set is known as the *Pareto set*.

Identifying the (exact) Pareto set is, however, not always feasible due to the computational complexity of the underlying problem, or to the black-box nature of the objectives. Indeed, when no assumption or additional information about the objectives are known or provided, the optimization method faces a *black-box* problem. The algorithm can only make a series of requests to an evaluation function that solely returns the objective-values of a solution provided as input, without any further information. On top of that, the conflicting nature of the objectives often severely impact the number of optimal trade-offs, and therefore the cardinality of the Pareto set to be identified. For these reasons, one must generally settle with a Pareto set approximation,

for which evolutionary algorithms and other search heuristics are a method of choice, although it remains impossible to recommend a priori which algorithm to select from the plethora of available methods for solving a given problem. This is precisely the context in which the research work presented in this manuscript lies.

1.2 Overview

The activities presented in this manuscript are articulated along three complementary research lines. Firstly, being interested in understanding more precisely the foundations and behavior of search heuristics, this led us to consider *landscape analysis* as a central concept in the first set of contributions. The main goal of landscape analysis is to inform about the structure of the search space, from the point of view of search algorithms. As such, a number of general-purpose features are proposed and analyzed for characterizing the landscape of black-box multi-objective combinatorial optimization problems. These landscape features allow us to better understand the difficulties that algorithms have to face depending on the problem being solved. Not only this, they subsequently allow us to predict algorithm performance, and to automate the tedious task of selecting the algorithm which is most likely to efficiently solve a previously-unseen problem.

Secondly, starting from the observation that multi-objective optimization aims at identifying a *set* of solutions, it becomes relevant to consider the search space as the collection of all *sets* of solutions. This is the principle of *set-based multi-objective search*. Originally proposed for performance assessment, a number of set preference relations from the literature can therefore naturally be considered as selection criteria for set-based multi-objective search. In this line, we start by clarifying the differences and similarities between sets for different set preference relations. We further specify local search components in such a setting, and we investigate the search difficulty as a function of the problem characteristics and the considered set preference relation.

In addition to investigating the fundamental issues presented above, we have thirdly contributed to the design and the improvement of *efficient* optimization approaches. This research line mainly aims at accelerating the convergence of multi-objective search algorithms, and thus improving their *anytime* performance profile; i.e., achieving a better approximation quality in a lower runtime. To this end, we rely heavily on the concept of *decomposition*, that consists in decomposing the considered multi-objective optimization problem into a number of scalar (single-objective) sub-problems that are solved concurrently and cooperatively. This allows us to propose a number of *distributed* approaches that incorporate a high level of *parallelism*, and that can be deployed on modern computing environments. In a complementary way, we consider surrogate models to the evaluation function, and we investigate their integration into the multi-objective search process in order to address particularly *expensive* optimization problems.

1.3 Supervision

The research presented in this manuscript is a result of joint works with a number of national and international students and young researchers. First of all, several contributions come from co-supervised PhD theses with the University of Lille:

- ▶ The PhD thesis of **Geoffrey Pruvost** was defended at the University of Lille in 2021. It contributes to the improvement of decomposition approaches for efficient multi-objective search, and is presented in Chapter 5.
- ▶ The PhD thesis of **Alexandre D. Jesus** will be defended in 2022, and was conducted in cotutelle between the University of Lille and the University of Coimbra in Portugal. Algorithm selection approaches for multi-objective search are presented in Chapter 3, and anytime performance models are presented in Chapter 4.
- ▶ The PhD thesis of **Nicolas Berveglieri** is conducted at the University of Lille and will also be defended in 2022. It deals with surrogate-assisted search for expensive multi-objective optimization, and is presented in Chapter 5.
- ▶ The PhD thesis of **Raphaël Cosson** is conducted at the University of Lille since 2019. It deals with landscape analysis and algorithm selection for decomposition approaches, and is presented in Chapter 3.

The research work of *external* co-supervised PhD students is presented in this manuscript as well:

- ▶ The PhD thesis of **Miyako Sagawa** was defended at Shinshu University in Japan in 2018. We present in Chapter 5 how discovering the importance of variables can positively guide the search process and accelerate multi-objective algorithms.
- ▶ The PhD thesis of **Hugo Monzón** was defended at Shinshu University in Japan in 2020. Empirical models are designed in Chapter 3 to capture the dynamics of evolving solutions in multi-objective search.

The collaborative work of co-supervised Postdoctoral researchers is also summarized below:

- ▶ **Fabio Daolio** conducted Postdoctoral research with the University of Lille and Shinshu University in Japan from 2014 to 2016. Together, we analyzed landscape features and performance prediction models that are presented in Chapter 3.
- ▶ **Saúl Zapotecas Martínez** also conducted Postdoctoral research with the University of Lille and Shinshu University in Japan from 2014 to 2016. We jointly contributed to multi-objective search decomposition, as presented in Chapter 5.

1.4 Collaborations

The work considered here is the outcome of a teamwork that would not have been made possible without the fruitful collaboration of a number

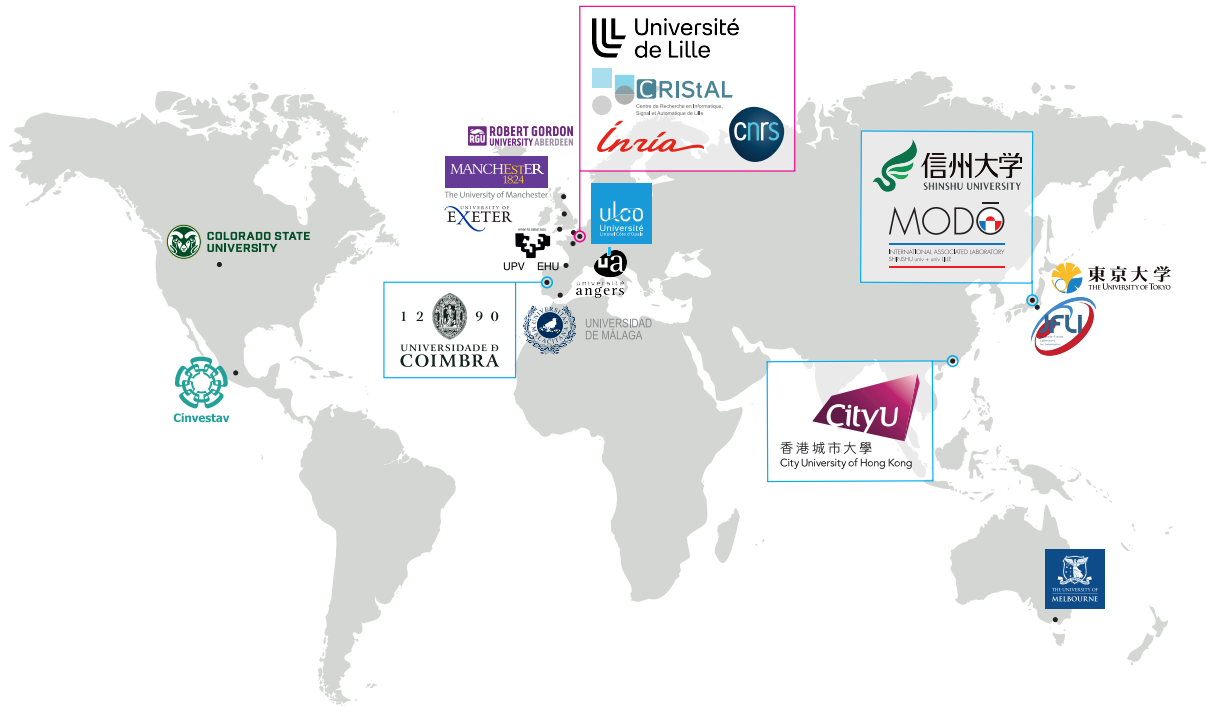


Figure 1.1: Main institutions and collaborations.

of senior researchers from national and international institutions, as summarized in Figure 1.1.

At the national level, we have a close collaboration with the University of the Littoral Opal Coast on almost all the subjects covered in this manuscript. The University of Angers is also an important collaborator, in particular on set- and indicator-based multi-objective search (Chapter 4), but also on the solving of problems from quadratic optimization and scheduling (Chapter 2).

At the European level, we have a sustained collaboration with the University of Coimbra in Portugal since 2010, both on the foundations of multi-objective search (Chapter 3) and on set- and indicator-based multi-objective search (Chapter 4). On the latter topic, we also collaborate regularly with the University of Manchester in the UK and the University of Málaga in Spain. More recently, we initiated collaborations with the University of the Basque Country in Spain, and with the University of Exeter and Robert Gordon University in the UK.

On the other continents, our main collaborators are located in Japan, notably at Shinshu University. Since 2012, our joint work focuses on landscape analysis (Chapter 3) and on the study and improvement of multi-objective search approaches (Chapter 5), in the context of the MODO international associated laboratory. In addition, a joint research project on decomposition-based multi-objective search (Chapter 5) was conducted with the City University of Hong Kong from 2016 to 2021. We also have more occasional collaborations with the University of Melbourne in Australia, Colorado State University in the USA, and Cinvestav in Mexico.

1.5 Outline

The manuscript is divided into six chapters, of which the remaining ones are organized as follows:

- ▶ In **Chapter 2**, we provide the necessary background on multi-objective optimization, we present the considered benchmark problems and state-of-the-art multi-objective search algorithms, and we discuss performance assessment in multi-objective search.
- ▶ In **Chapter 3**, we present our contributions on the foundations of multi-objective search and landscape analysis. We introduce and analyze multi-objective landscape features for small and large landscapes, we investigate feature-based prediction models for algorithm performance and automated algorithm selection, and we summarize some contributions related to these topics.
- ▶ In **Chapter 4**, we present our contributions on set- and indicator-based multi-objective search. We review a selection of quality indicators and we statistically analyze their degree of agreement. Then, we define set-based multi-objective landscapes and local search, and we study the number and properties of local optimal sets. We also review related contributions to this research line.
- ▶ In **Chapter 5**, we present our contributions on efficient multi-objective search. After introducing the core concept of decomposition, we propose a number of decentralized approaches and analyze their parallel efficiency. Besides, we investigate the integration of surrogate models in multi-objective search to accelerate convergence, and briefly discuss related contributions.
- ▶ In **Chapter 6**, we conclude the manuscript by presenting our future research plan on landscape-aware massive optimization, and by discussing a number of perspectives that go beyond those presented in previous chapters.

Before moving on to the next chapter, let us highlight that references to papers I co-authored appear in light blue. For instance, [Verel et al. \(2013\)](#) is a self-citation whereas [Zitzler et al. \(2003\)](#) refers to a paper of which I am not a co-author.

Multi-objective Optimization

In this chapter, we introduce the basic notions related to multi-objective optimization, with a particular focus on black-box multi-objective combinatorial optimization problems. We start by giving the definitions related to Pareto dominance and Pareto optimality. Then, we present some multi-objective optimization problems considered in the manuscript. Finally, we give an overview of the considered multi-objective optimization approaches, and we discuss how to assess algorithm performance.

- 2.1 Definitions 7
- 2.2 Problems 8
- 2.3 Algorithms 10
- 2.4 Performance 13

2.1 Definitions

Let us consider an arbitrary multi-objective optimization problem $f: X \mapsto Z$, with X being the *variable space*, $Z \subseteq \mathbb{R}^m$ the *objective space*, and $f = (f_1, \dots, f_i, \dots, f_m)$ an objective function vector such that each objective f_i is to be maximized, $i \in \{1, \dots, m\}$. As illustrated in Figure 2.1, each solution $x \in X$ is associated with an objective vector $z \in Z$ such that $z = f(x)$. In the combinatorial case, the variable space X is a discrete set. The *Pareto dominance relation*, whose concept is named after Vilfredo Pareto (Figure 2.2), is defined as follows. Given two objective vectors $z, z' \in Z$, z is (weakly) *dominated* by z' (denoted as $z \leq z'$) if $z_i \leq z'_i$ for all $i \in \{1, \dots, m\}$. They are *mutually non-dominated* if $z \not\leq z'$ and $z' \not\leq z$. They are *equivalent* if $z \leq z'$ and $z' \leq z$. As illustrated in Figure 2.3, an objective vector $z^* \in Z$ is *Pareto optimal* or *non-dominated* if there does not exist any $z \in Z, z \neq z^*$, such that $z^* \leq z$. Corresponding definitions can be formalized for solutions $x \in X$ by using their objective vector $z \in Z$ such that $z = f(x)$. In particular, a solution $x^* \in X$ is *Pareto optimal* or *non-dominated* if $f(x^*)$ is non-dominated. The *Pareto front* $Z^* \subseteq Z$ is the set of non-dominated objective vectors. The *Pareto set* $X^* \subseteq X$ is a set of solutions that maps to the Pareto front, i.e. $f(X^*) = Z^*$. The Pareto set and its corresponding Pareto front are illustrated in Figure 2.4.

Unfortunately, identifying the Pareto set of a multi-objective combinatorial optimization problem is often infeasible for two main reasons (Ehrgott, 2005). Firstly, most problems are intractable, in the sense that the number of Pareto optimal solutions typically grows exponentially with the problem size. Secondly, deciding if a solution belongs to the Pareto set may be NP-complete. Therefore, the decision maker often has to rely on a Pareto set *approximation*. This naturally excludes the applicability of exact methods (Ehrgott, 2005). There also exist a number of approximation methods with provable guarantee, which assume that a sequence of single-objective auxiliary problems can be solved in polynomial time (Herzel et al., 2021). In order to alleviate this hypothesis, our work mostly focuses on heuristics. Evolutionary multi-objective optimization algorithms and other multi-objective

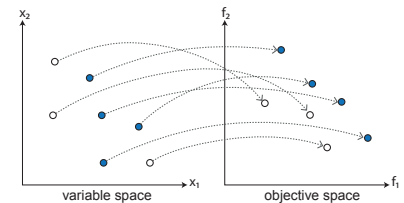


Figure 2.1: Variable and objective spaces.

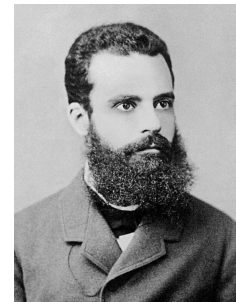


Figure 2.2: Vilfredo Pareto, Italian economist, 1848–1923.

https://upload.wikimedia.org/wikipedia/commons/f/fd/Vilfredo_Pareto_1870s2.jpg

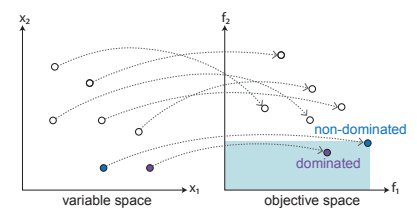


Figure 2.3: Pareto dominance.

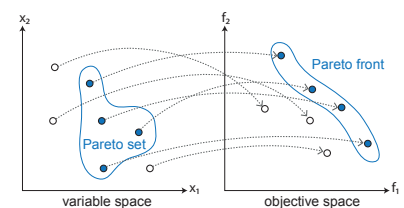


Figure 2.4: Pareto set and Pareto front.

search heuristics aim at identifying an approximation set of limited cardinality, ideally a subset of the Pareto set, that is to be presented to the decision maker for further consideration (Branke et al., 2008; Coello Coello et al., 2007; Deb, 2001). Such methods also have the advantage of not requiring any particular knowledge about the problem to be solved, the objective function vector being seen as a *black-box* that solely returns the objective values of solutions given as input.

2.2 Problems

Let us now introduce some multi-objective combinatorial optimization problems considered for benchmarking in the following chapters. We contributed a number of benchmark problem formulations, by paying a particular attention to the correlation among the objectives. In order to automate the design of a wide range of problems and thus increase their uptake within the community, we also proposed configurable benchmark problem generators. The source code of the generators, together with instance datasets and their corresponding reference fronts (either the Pareto fronts or best-known approximations) are made available at the following URL: <http://mocobench.sf.net>.

ρ mnk-Landscapes

A large part of our investigations consider ρ mnk-landscapes as a configurable benchmark for multi-objective combinatorial optimization. We proposed ρ mnk-landscapes as a problem-independent model for constructing multi-objective multimodal landscapes with objective correlation (Verel et al., 2013). They extend single-objective nk-landscapes (Kauffman, 1993) and multi-objective nk-landscapes with independent objectives (Aguirre and Tanaka, 2007). Solutions are binary strings of size n . The objective function vector $f = (f_1, \dots, f_i, \dots, f_m)$ is defined as $f: \{0, 1\}^n \mapsto [0, 1]^m$ such that each objective f_i is to be maximized. As in the single-objective case, the objective value $f_i(x)$ of a solution $x = (x_1, \dots, x_j, \dots, x_n)$ is an average value of the individual contributions associated with each variable x_j . Given objective f_i , $i \in \{1, \dots, m\}$, and variable x_j , $j \in \{1, \dots, n\}$, a component function $f_{ij}: \{0, 1\}^{k+1} \mapsto [0, 1]$ assigns a real-valued contribution for every combination of x_j and its k variable interactions $\{x_{j_1}, \dots, x_{j_k}\}$. These f_{ij} -values are uniformly distributed in $[0, 1]$. Thus, the individual contribution of a variable x_j depends on its own value, and on the values of $k < n$ variables other than x_j . The problem can be formalized as follows:

$$\begin{aligned} \max \quad & f_i(x) = \frac{1}{n} \sum_{j=1}^n f_{ij}(x_j | x_{j_1}, \dots, x_{j_k}) \quad i \in \{1, \dots, m\} \\ \text{s.t.} \quad & x_j \in \{0, 1\} \quad j \in \{1, \dots, n\} \end{aligned}$$

The variable interactions, non-linearity, or epistasis, i.e. the k variables that influence the contribution of x_j , are set uniformly at random

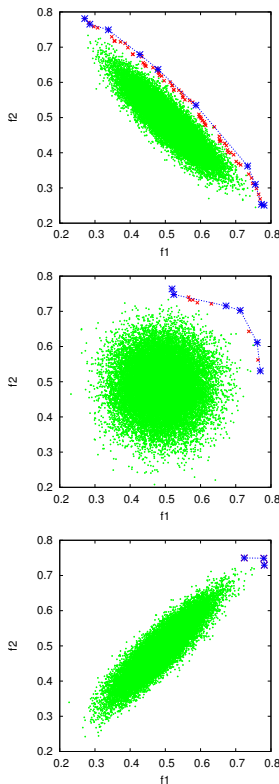


Figure 2.5: Objective space of ρ mnk-landscapes with a negative ($\rho = -0.9$), a null ($\rho = 0$), and a positive ($\rho = 0.9$) correlation among the objectives. Green points correspond to a sample (10%) of random objective vectors, blue and red points are supported and unsupported non-dominated objective vectors, respectively. The problem size is $n = 18$, the variable interaction degree is $k = 4$, and the number of objectives is $m = 2$.

among the $(n - 1)$ variables other than x_j , following the random neighborhood model from Kauffman (1993). By increasing the number of interactions k per variable from 0 to $(n - 1)$, problem instances can be gradually tuned from smooth to rugged. In ρmnk -landscapes, f_{ij} -values additionally follow a multivariate uniform distribution of dimension m , defined by an $m \times m$ positive-definite symmetric covariance matrix (c_{pq}) such that $c_{pp} = 1$ and $c_{pq} = \rho$ for all $p, q \in \{1, \dots, m\}$ with $p \neq q$, where $\rho > \frac{-1}{m-1}$ defines the correlation among the objectives; see Verel et al. (2013) for details. As illustrated in Figure 2.5, the positive (respectively, negative) objective correlation ρ decreases (respectively, increases) the degree of conflict among the objectives. By default, the same correlation coefficient ρ is used among all pairs of objectives, and the same variable interactions are set for all the objectives. Notice that general nk -landscapes are NP-hard in the single-objective case (Hoos and Stützle, 2005). Moreover, we empirically show in Verel et al. (2013) that the number of Pareto optimal solutions grows exponentially with the problem size, so that ρmnk -landscapes are intractable as well.

Objective Correlation

Although the greatest challenge of multi-objective optimization is often believed to be the number of objectives, we argue that the correlation among them is also crucially important. As already illustrated in Figure 2.5, a negative objective correlation substantially increases the number of Pareto optimal solutions for two-objective instances. Additional properties will also be discussed in view of objective correlation in the next chapters. Notice that there exists an earlier attempt to partially tune the objective correlation for multi-objective nk -landscapes (Knowles and Corne, 2007) and multi-objective quadratic assignment problems (Knowles and Corne, 2003a), where correlation is generated between the first and any other objective. By contrast, our methodology, based on a multivariate distribution, enables to finely tune the correlation for each pair of objectives, as reported in Figure 2.6. With this in mind, we employed a similar method to define or generalize other classes of multi-objective combinatorial optimization benchmark problems with a tunable objective correlation, including different variants from the binary knapsack problem (Liefoghe et al., 2013a), the multi-objective unconstrained binary quadratic programming problem (Liefoghe et al., 2014, 2015b) and the multi-objective traveling salesperson problem (Derbel et al., 2016).

Even when not explicitly defined by a benchmark parameter, practical multi-objective optimization problems might exhibit different degrees of conflict among the objectives. This is the case, for instance, of multi-objective permutation flowshop scheduling problems (Liefoghe et al., 2017a). The objectives are defined based on different mathematical expressions, ranging from the maximum completion time or makespan (C_{\max}), to the sum of completion times (C_{sum}), the maximum tardiness (T_{\max}), the sum of tardiness (T_{sum}), or the number of late jobs (T_{card}). The correlation among each pair of objectives is reported in Figure 2.7. We show in Liefoghe et al. (2017a) that this correlation influences instance properties and algorithm performance.

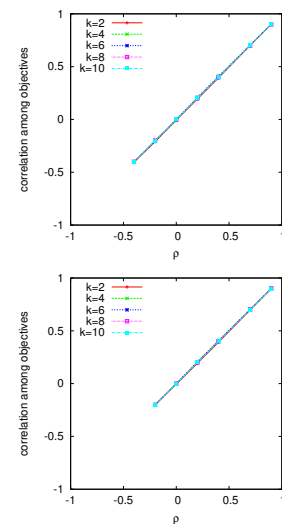
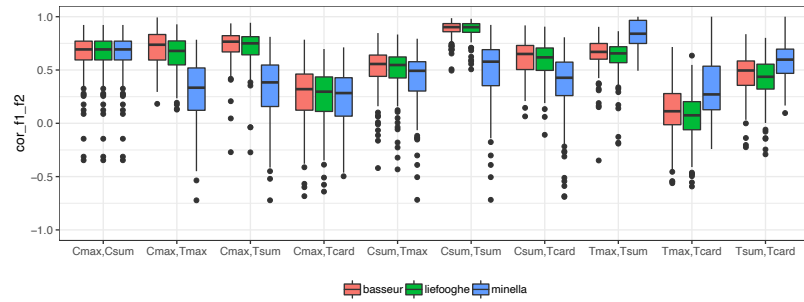


Figure 2.6: Spearman rank correlation among objective vectors for ρmnk -landscapes with respect to ρ . The problem size is $n = 18$, the variable interaction degree is $k \in \{2, 4, 6, 8, 10\}$ and the number of objectives is $m = 3$ (top) and $m = 5$ (bottom).

Figure 2.7: Spearman rank correlation among objective vectors for two-objective permutation flowshop scheduling problems with 8 jobs and 8 machines for different pairs of objectives (x-axis) and instance types (color).



A Glimpse on Other Problems

In addition to ρ mnk-landscapes, we investigated other multi-objective optimization problems. Apart from the knapsack (Liefoghe et al., 2013a), the unconstrained binary quadratic programming (Liefoghe et al., 2014, 2015b), the traveling salesperson (Derbel et al., 2016), and the permutation flowshop scheduling (Liefoghe et al., 2017a) problems mentioned above, it is worth mentioning the multi-objective quadratic assignment problem (Knowles and Corne, 2003a) as well standard benchmark test suites from multi-objective *continuous* optimization (Brockhoff et al., 2019; Deb et al., 2005; Huband et al., 2006; Zhang et al., 2009)¹. This provides a large spectrum of multi-objective optimization problems spanning a diversified range in terms of variable types and representations (binary strings, permutations, real-values), variable interactions (from linear to highly multimodal problems), variable interactions (from linear to highly multimodal problems), number of objectives (from bi- to multi- and many-objective problems) and the correlations among them.

1: Although not considered later in this document, we mention here our work on bi-objective pairwise sequence alignment in bioinformatics, for which we proposed extensions of dynamic programming algorithms for several problem variants with a novel pruning technique that efficiently reduces the number of states to be processed (Abbasi et al., 2013). The corresponding MOSAL software is available at: <http://mosal.dei.uc.pt>.

2.3 Algorithms

We describe below a number of general-purpose randomized search heuristics (or stochastic local search algorithms, evolutionary algorithms, metaheuristics) for multi-objective optimization. We start by exemplifying the difference between *local* and *evolutionary* multi-objective search, and then we focus on representative state-of-the-art multi-objective evolutionary algorithms, covering the classes of dominance-, indicator-, and decomposition-based approaches.

The source code of some algorithms and components developed in this work are made available in the Paradiseo open-source software framework (Cahon et al., 2004; Keijzer et al., 2001), currently available at the following URL: <https://nojhan.github.io/paradiseo>. As illustrated in Figure 2.8, Paradiseo is based on a modular design. We significantly contributed to the module for multi-objective optimization Paradiseo-MOE0 (Liefoghe et al., 2011), and to the module for local search and landscape analysis Paradiseo-M0 (Humeau et al., 2013). A recent summary of the features from the Paradiseo framework can be found in Dréo et al. (2021).

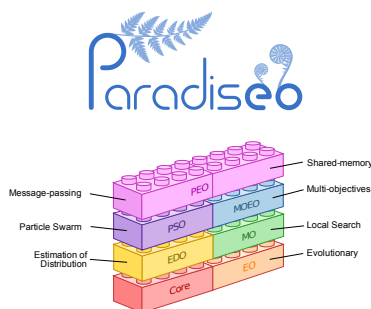


Figure 2.8: The Paradiseo modules.

J. Dréo, CC-BY-SA, <https://nojhan.github.io/paradiseo>

Local vs. Evolutionary Multi-objective Search

Let us assume that we are given a multi-objective optimization problem with n binary variables. We describe below two randomized multi-objective search heuristics based on *dominance*: (i) the Global Simple Evolutionary Multi-objective Optimizer (G-SEMO) from Laumanns et al. (2004a), a simple elitist steady-state *global* multi-objective evolutionary algorithm; and (ii) Pareto local search (PLS) from Paquete et al. (2007), a multi-objective *local* search. Both algorithms maintain an *unbounded archive* A of mutually non-dominated solutions. This archive is initialized with one random solution from the solution space. At each iteration, one solution is selected at random from the archive $x \in A$. In G-SEMO, each binary variable from x is independently flipped with a rate of $1/n$ in order to produce an offspring solution x' . The archive is then updated by keeping the non-dominated solutions from $A \cup \{x'\}$. In PLS, the solutions located in the neighborhood of x are evaluated. Let $\mathcal{N}(x)$ be the set of solutions located at a Hamming distance 1. The non-dominated solutions from $A \cup \mathcal{N}(x)$ are stored in the archive, and the current solution x is tagged as *visited* in order to avoid unnecessary re-evaluation of its neighborhood in subsequent iterations. This process is iterated until a stopping condition is satisfied. While G-SEMO does not have any explicit stopping rule, PLS naturally stops once all solutions from the archive are tagged as *visited*. When necessary, we consider a simple iterated version of PLS (I-PLS), that restarts from a solution randomly chosen from the archive and perturbed by stochastic mutation (Drugan and Thierens, 2012). While PLS is based on the exploration of the whole 1-bit-flip neighborhood from x , G-SEMO rather uses an ergodic operator, i.e. an independent bit-flip mutation. Hence, every iteration has a non-zero probability of reaching any solution from the solution space. This makes G-SEMO a *global* (evolutionary) search, as opposed to a *local* search like PLS.

Multi-objective Evolutionary Algorithms

By contrast with G-SEMO and PLS presented above, most multi-objective evolutionary algorithms uses a *fixed-size population*, although an external archive can be used to store all non-dominated solutions found so far during the search process. Selection in multi-objective evolutionary computation can be classified into the following classes; see, e.g., Coello Coello et al. (2007) or Liefvooghe et al. (2011):

- ▶ **Dominance-based approaches**, where the Pareto dominance relation is used as the main criterion to rank solutions from the population.
- ▶ **Indicator-based approaches**, where solutions are compared on the basis of a quality indicator, which thus represents the overall goal of the search process.
- ▶ **Decomposition-based approaches**, where the multi-objective optimization is decomposed into a number of (single-objective) sub-problems based on a scalarizing function, the sub-problems being optimized simultaneously and cooperatively.

We describe below three exemplary multi-objective evolutionary algorithms, namely NSGA-II (Deb et al., 2002), IBEA (Zitzler and Künzli, 2004), and MOEA/D (Zhang and Li, 2007). They were selected as representatives of the state-of-the-art in evolutionary multi-objective optimization, covering dominance-, indicator-, and decomposition-based approaches, respectively.

NSGA-II (Deb et al., 2002) is an elitist *dominance*-based multi-objective evolutionary algorithm using Pareto dominance for survival and parent selections. At a given iteration, the current population P_t is merged with its offspring Q_t , and is divided into non-dominated fronts $F = \{F_1, F_2, \dots\}$ based on the non-dominated sorting procedure (Goldberg, 1989). The front in which a given solution belongs to gives its rank within the population. Crowding distance is also calculated within each front. Selection is based on ranking, and crowding distance is used as a tie breaker. Survival selection consists in filling the new population P_{t+1} with solutions having the best (smallest) ranks. In case a front F_i overfills the population size, the required number of solutions from F_i are chosen based on their crowding distance. Parent selection for reproduction consists of binary tournaments between randomly-chosen solutions, following the lexicographic order induced by ranks first, and crowding distance next.

IBEA (Zitzler and Künzli, 2004) introduces a total order between solutions by means of a binary quality *indicator* I . Its selection mechanism is based on a pairwise comparison of solutions from the current population P_t with respect to I . A fitness value is assigned to each solution $x \in P_t$, measuring the “loss in quality” if x was removed from the current population:

$$Fitness(x) := \sum_{x' \in P \setminus \{x\}} (-e^{-I(x',x)/\kappa}) \quad (2.1)$$

where $\kappa > 0$ is a user-defined scaling factor. The survival selection mechanism is based on an elitist strategy that combines the current population P_t with its offspring Q_t . It iteratively removes the worst solution until the required population size is reached, and assigns the resulting population into P_{t+1} . Each time a solution is deleted, the fitness values of the remaining solutions are updated. Parent selection for reproduction consists of binary tournaments between randomly chosen solutions. Different indicators can be used within IBEA. The binary additive ε -indicator ($I_{\varepsilon+}$) is defined as follows by Zitzler and Künzli (2004):

$$I_{\varepsilon+}(x, x') := \max_{i \in \{1, \dots, m\}} \{f_i(x) - f_i(x')\} \quad (2.2)$$

It gives the minimum value by which a solution $x \in P_t$ has to, or can be, translated in the objective space in order to weakly dominate another solution $x' \in P_t$. Indicator-based multi-objective search will be discussed in more details in Chapter 4.

MOEA/D (Zhang and Li, 2007) is a *decomposition*-based multi-objective evolutionary algorithm that seek a high-quality solution in multiple regions of the objective space by decomposing the original (multi-objective) problem into a number of scalar (single-objective) sub-problems. Let μ be the population size. A set $(\lambda^1, \dots, \lambda^i, \dots, \lambda^\mu)$ of uniformly-distributed weighting coefficient vectors defines the scalar sub-problems, and a population $P = (x^1, \dots, x^i, \dots, x^\mu)$ is maintained such that each solution x^i maps to the sub-problem defined by λ^i . Different scalarizing functions can be used within MOEA/D. For instance, the weighted Chebyshev scalarizing function (Miettinen, 1999) is defined as follows:

$$g(x | \lambda) := \max_{i \in \{1, \dots, m\}} \lambda_i \cdot |z_i^* - f_i(x)| \quad (2.3)$$

such that x is a solution, λ is a weighting coefficient vector and z^* is a reference point. In addition, a neighboring relation is defined among sub-problems, based on the assumption that a given sub-problem is likely to benefit from the solution maintained in neighboring sub-problems. The neighborhood $B(i)$ is defined by considering the T closest weighting coefficient vectors for each sub-problem i . At each iteration, the population evolves with respect to a given sub-problem. Two solutions are selected at random from $B(i)$ and an offspring is produced by means of variation operators. Then, for each sub-problem $j \in B(i)$, the offspring is used to replace the current solution x^j if there is an improvement in terms of the scalarizing function. The algorithm iterates over sub-problems until a stopping condition is satisfied. Further considerations about MOEA/D and decomposition-based multi-objective search will be discussed in Chapter 5.

2.4 Performance

When benchmarking heuristic search algorithms, performance is typically measured in terms of (Hoos and Stützle, 2005):

- ▶ **Runtime**; i.e. how fast an algorithm hits a given target.
- ▶ **Approximation quality**; i.e. which quality level is achieved after an algorithm is run for a given budget.
- ▶ **Success rate**; i.e. how frequently an algorithm reaches a given target under a given budget.

Given that search heuristics are *anytime algorithms* (Dean and Boddy, 1988; Zilberstein, 1996), i.e. they have the ability to return valid solutions when interrupted at any time of the search process, considering different targets or budgets is important for assessing their anytime behavior. In addition, due to the stochastic nature of algorithms and problem instances, multiple runs are typically performed. Empirical results can then be summarized using graphical plots, descriptive statistics and statistical tests, for which we mostly rely on the R statistical computing and graphics environment (R Core Team, 2020), and the `ggplot2` package (Wickham, 2016). Whenever appropriate, we also consider multiple instances to render the robustness of an algorithm

when solving a given problem class, or on the contrary to show that an algorithm performs differently for different problems or instance types.

As in single-objective optimization, the budget can be measured in terms of CPU time, or in terms of calls to the evaluation function — the objective function vector. However, in multi-objective optimization, approximation quality needs to take into account that a whole *approximation set* is sought. A number of *quality indicators* have been proposed to measure the quality of multi-objective algorithms. Quality indicators are mathematically defined and presented in details in Chapter 4, where we also analyze their correlation and properties. For now, we simply give a brief description of the indicators considered in the next chapter. The *resolution* gives the proportional number of Pareto optimal solutions that have been identified. It is then assumed that the (optimal) Pareto set is known, which limits its use to small-size problems, or artificial problems for which this information is available. By contrast, the *hypervolume* (Zitzler and Thiele, 1998) gives the multi-dimensional area of the objective space dominated by an approximation set, with respect to a user-given reference point. The *epsilon* indicator (Zitzler et al., 2003) gives the minimum factor by which an approximation set has to be translated in the objective space in order to (weakly) dominate a reference set.

Foundations of Multi-objective Landscapes and Search

3

In this chapter, we expose and contrast the impact of landscape characteristics on the performance of search algorithms for multi-objective optimization problems. A sound and concise summary of features characterizing the structure of an arbitrary problem instance are identified and related to the expected performance of global and local dominance-based multi-objective search algorithms. We provide a critical review of existing features for multi-objective optimization, and we propose additional ones that do not require any global knowledge from the landscape, making them suitable for large-size black-box problem instances. Their intercorrelation and their association with algorithm performance are also analyzed. This allows us to assess the individual and the joint effect of landscape features on algorithm performance, and to highlight the main difficulties encountered by such search heuristics. By providing effective tools for multi-objective landscape analysis, we highlight that multiple features are required to capture problem difficulty, and we provide further insights into the importance of ruggedness and multimodality to characterize multi-objective landscapes. We further illustrate how these multi-objective landscape features can be used for automated algorithm selection. The main focus of the chapter is thus given to the research work presented in [Liefvooghe et al. \(2020a\)](#), which builds upon an incremental series of publications in this line ([Daolio et al., 2015, 2017](#); [Liefvooghe et al., 2013b, 2015a](#); [Verel et al., 2013](#)). Nevertheless, we conclude the chapter with a summary of related contributions, together with further considerations.

3.1 Motivations	15
3.2 Features	17
3.3 Features vs. Performance	24
3.4 Scaling to Large Landscapes	28
3.5 Algorithm Selection	31
3.6 Further Contributions	33
3.7 Outlook	44

3.1 Motivations

In single-objective optimization, landscape analysis has emerged as a valuable set of tools to characterize problem difficulty; see [Richter and Engelbrecht \(2014\)](#) or [Humeau et al. \(2013\)](#). An example of a single-objective landscape with two variables and one objective is given in [Figure 3.1](#). Contrary to problem-specific structural properties such as the average vertex degree in the minimum vertex cover problem ([Wagner et al., 2017](#)) or the maximum cost between two cities in the traveling salesman problem ([Mersmann et al., 2013](#)), landscape analysis aims at designing general-purpose features that do *not* depend on a specific problem class or problem domain. Instead, it tries to characterize the topology of black-box problems in the eye of the challenges that stochastic local search algorithms have to face when tackling them. Of particular interest is the number and distribution of local optima in the landscape, i.e. multimodality and ruggedness ([Kauffman, 1993](#); [Kerschke et al., 2016](#); [Merz, 2004](#); [Preuss, 2015](#); [Weinberger, 1990](#)). These features are empirically related to instance hardness and algorithm efficiency, and provide significant insights into the interplay between the problem structure and the behavior of search algorithms and their working components. Pioneering works on multi-objective landscape

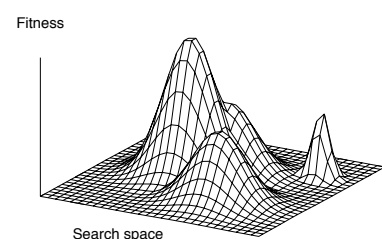


Figure 3.1: Illustration of a single-objective landscape.

S. Verel, <http://www-lisic.univ-littoral.fr/~verel/pictures/multimodalFitnessLandscape.jpg>

analysis include Aguirre and Tanaka (2007), Garrett and Dasgupta (2008, 2009), Knowles and Corne (2003a, 2007), Paquete et al. (2007), and Paquete and Stützle (2009). We build upon those by considering general-purpose problem features defined therein. In addition, we derive new landscape features that our analysis reveals as highly impactful for multi-objective search.

The purpose of designing features to characterize search difficulty is two-fold: (i) gathering a fundamental understanding of optimization problems and algorithms, eventually leading to a better algorithm design, and (ii) automatically predicting performance or selecting algorithm based on relevant features. More particularly, feature-based performance prediction consists of modeling the expected runtime or solution quality of a given algorithm applied to a problem instance exhibiting particular features. This, in turn, might also lead to algorithm selection (Kerschke et al., 2019; Smith-Miles, 2008) and configuration (Belkhir et al., 2017; Kadioglu et al., 2010), where the best-performing algorithm or configuration is to be selected from a set of competitors. As illustrated in Figure 3.2, this issue is not specific to optimization, and is known as the *algorithm selection problem* (Rice, 1976), which is one of the core concept of *autonomous search* (Hamadi et al., 2012). A statistical or machine learning regression or classification model is constructed by means of extensive experiments on a training set of instances, and this model is later used to predict the performance or to select between algorithms for previously-unseen instances. This research area has received a growing attention in recent years, mainly by relying on features that require a specific domain knowledge from the target combinatorial optimization problem; see e.g. Hutter et al. (2014), Mersmann et al. (2013), and Smith-Miles et al. (2014). Few exceptions can be found in Beham et al. (2018), Daolio et al. (2012), and Hutter et al. (2011). where the selection among an algorithm portfolio is performed using general-purpose landscape features related to local optimality, ruggedness and multimodality. Research in this line can also be found for *continuous* single-objective optimization (Bischl et al., 2012; Kerschke and Trautmann, 2019; Malan and Engelbrecht, 2014). However, to our knowledge, black-box landscape features have never been used for performance prediction or algorithm recommendation in the context of multi-objective combinatorial optimization. Although the statistical and machine learning models used in the single-objective case can be applied, multi-objective landscape features need to be carefully designed and analyzed, since existing single-objective features are not relevant for multi-objective optimization.

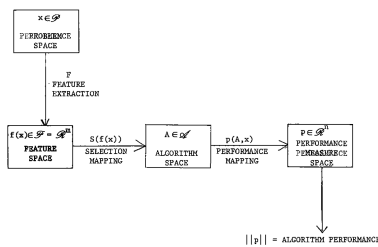


Figure 3.2: Algorithm selection model from Rice (1976).

This is precisely the purpose of the current study. Particularly, we first review and extend general-purpose features to characterize the different facets of difficulty encountered in multi-objective combinatorial optimization. Features include problem descriptors, such as the variable and objective space dimensions, *global* measures, that require the knowledge of all or part of the solution space, and *local* measures, that are computed from an affordable sample of solutions. Then, we analyze features correlation as well as their impact on the performance of two canonical multi-objective search algorithms, namely the global simple evolutionary multi-objective optimizer (Laumanns et al., 2004a)

and Pareto local search (Paquete et al., 2007) introduced in Section 2.3. We selected a *global* and a *local* elitist dominance-based algorithm, respectively, for the sake of clarifying our understanding of core multi-objective search components. Experiments are conducted on a family of multimodal pseudo-boolean optimization problems known as ρmnk -landscapes (Verel et al., 2013), and described in Section 2.2. By paying a particular attention to the computational cost induced by these features, we finally analyze their ability to predict algorithm performance and to select among a small algorithm portfolio. A sound statistical analysis allows us to highlight the main difficulties that dominance-based algorithms have to face, as well as the main differences induced by multi-objective search approaches.

The chapter is organized as follows. In Section 3.2, we identify a substantial number of existing and original features that characterize black-box multi-objective landscapes. In Section 3.3, based on small-size ρmnk -landscapes, we analyze the correlation among features, and we measure their ability to predict algorithm performance as well as their impact on search efficiency. In Section 3.4, we extend our analysis to large-size instances by focusing on local features, i.e. features that can be computed efficiently. In Section 3.5, we experiment with feature-based automated algorithm selection. In Section 3.6, we complement the discussion by briefly summarizing tightly related research issues that we had the opportunity to study. In the last section, we conclude the chapter and discuss further research.

3.2 Features to Characterize Multi-objective Landscapes

In this section, we present the set of multi-objective landscape features considered in our analysis. We start with *global* features from Daolio et al. (2015, 2017) and Liefoghe et al. (2013b, 2015a). Since they require the knowledge of all and/or Pareto optimal solutions in order to be computed, this makes them impractical for performance prediction and algorithm selection. However, we decided to include them in order to measure and understand their impact on search performance. Next, we introduce a number of *local* features from Liefoghe et al. (2020a), which are based on a reasonable subset of solutions sampled during random and adaptive walks, making them affordable in practice for performance prediction. The whole set of features is listed in Table 3.1, together with the ρmnk -landscape parameters described in Section 2.2, i.e. ρ , m , n , and k .

Definition 3.2.1 *As in single-objective optimization, we define a multi-objective landscape as a triplet (X, \mathcal{N}, f) such that:*

- ▶ X is a variable space,
- ▶ $\mathcal{N}: X \mapsto 2^X$ is a neighborhood relation,
- ▶ $f: X \mapsto Z$ is a (black-box) objective function vector.

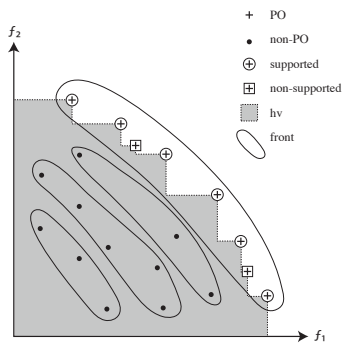


Figure 3.3: Illustration of global features extracted from the Pareto set, the Pareto front, and the solution space.

Global Features

Let us start with the subset of global features illustrated in Figure 3.3. In multi-objective combinatorial optimization, the number of Pareto optimal solutions is considered as an important aspect of difficulty. Generally speaking, the larger the Pareto set, the smaller the chance to identify all Pareto optimal solutions in an efficient manner. In that sense, most multi-objective combinatorial optimization problems are known to be *intractable*, i.e. the number of Pareto optimal solutions typically grows exponentially with the problem size (Ehrgott, 2005). As such, the proportion of Pareto optimal solutions in the solution space ($\#po$) is one of the most obvious facet to characterize problem difficulty; see, e.g., Aguirre and Tanaka (2007), Bazgan et al. (2013), Garrett and Dasgupta (2008), and Knowles and Corne (2003a, 2007). As illustrated in Figure 3.4, for ρmnk -landscapes, $\#po$ grows exponentially with the number of objectives and with their degree of conflict, while the number of variable interaction has a low influence (Verel et al., 2013). As such, when we have many and conflicting objectives, we expect the Pareto set cardinality to be very large, and to quickly become intractable. Pareto optimal solutions can further be classified into two categories: supported and non-supported solutions. A supported solution is an optimal solution of a weighted sum aggregation of the objectives, and is mapping to an objective vector that is located on the convex hull of the Pareto front (Ehrgott, 2005). As such, the proportion of supported solutions in the Pareto set ($\#supp$) might impact the general convexity of the Pareto front, as illustrated by previous studies on multi-objective landscape analysis (Knowles and Corne, 2003a). Similarly, the hypervolume (hv) covered by the (exact) Pareto front is shown to relate to the expected performance of multi-objective search algorithms (Aguirre and Tanaka, 2007). As discussed in Section 2.4, the hypervolume is one of the few recommended indicators for comparing solution sets in multi-objective optimization (Zitzler et al., 2003).

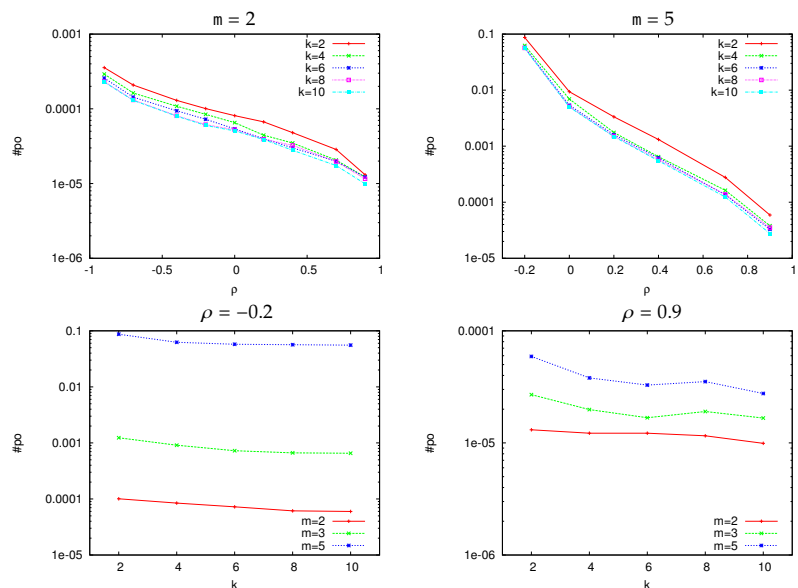


Figure 3.4: Average proportion of Pareto optimal solutions ($\#po$) with respect to objective correlation ρ for different k -values (top left $m = 2$, right $m = 5$), and with respect to the number of variable interactions k for different m -values (bottom left $\rho = -0.2$, right $\rho = 0.9$). The problem size is $n = 18$, thus allowing Pareto optimal solutions to be exhaustively enumerated.

Other relevant characteristics from the Pareto set deal with the distance and connectedness between Pareto optimal solutions. Knowles and Corne (2003a) study the maximal distance, in the variable space, between any pair of Pareto optimal solutions (`podist_max`). They denote this as the *diameter* of the Pareto set. For ρ mnk-landscapes, the distance measure is taken as the Hamming distance between binary strings, which is directly related to the bit-flip neighborhood operator. Similarly, the average distance between Pareto optimal solutions (`podist_avg`) can also be taken into account (Liefoghe et al., 2013b). Another measure capturing the dispersion of solutions is the entropy of the Pareto set (Knowles and Corne, 2003a), here measured as the entropy of (binary) variables from Pareto optimal solutions (`po_ent`). Extensions of the fitness-distance correlation, a widely acknowledged landscape measure (Jones and Forrest, 1995), to multi-objective optimization is discussed by Garrett and Dasgupta (2008) and Knowles and Corne (2003a). We here consider the correlation between the (Hamming) distance between Pareto optimal solutions and their Manhattan distance in the objective space (Knowles and Corne, 2003a). Another important property of the Pareto set topology is *connectedness* (Ehrgott and Klamroth, 1997; Gorski et al., 2011). The Pareto set is connected if all Pareto optimal solutions are connected with respect to a given neighborhood structure. This makes it possible for local search to identify the whole Pareto set by starting with one Pareto optimal solution. Let us define a graph such that each node corresponds to a Pareto optimal solution, and there is an edge between two nodes if the corresponding solutions are neighbors in the landscape. As shown by Paquete and Stützle (2009) and Liefoghe et al. (2013b), the degree of connectedness impacts the performance of multi-objective local search. We here consider the following related landscape features, illustrated in Figure 3.5: the proportion of connected components in this Pareto graph (`#cc`) (Paquete and Stützle, 2009), the proportion of isolated nodes (`#sing`) (Paquete and Stützle, 2009), the proportional size of the largest connected component (`#lcc`) (Liefoghe et al., 2013b; Verel et al., 2013), as well as the average distance between pairs of nodes (`lcc_dist`) and the proportion of hypervolume covered by the largest connected component from the Pareto graph (`lcc_hv`) (Liefoghe et al., 2020a).

The characteristics of the Pareto set and the Pareto front are, however, not the sole factors that impact the performance of multi-objective search algorithms. Aguirre and Tanaka (2007), Garrett and Dasgupta (2009), and Knowles and Corne (2007) analyze how the landscape affects the number of non-dominated fronts, and how this relates to search performance. As illustrated in Figure 3.3, the whole set of solutions from the search space is divided into different layers of mutually non-dominated solutions, following the principles of non-dominated sorting (Goldberg, 1989) used, e.g., in NSGA-II (Deb et al., 2002). To cater for this, we measure both the proportion of non-dominated fronts in the solution space (`#fronts`) (Aguirre and Tanaka, 2007; Garrett and Dasgupta, 2009; Knowles and Corne, 2007), and the entropy of the non-dominated front's size distribution (`front_ent`) (Liefoghe et al., 2020a). Finally, one of the main landscape features in single-objective

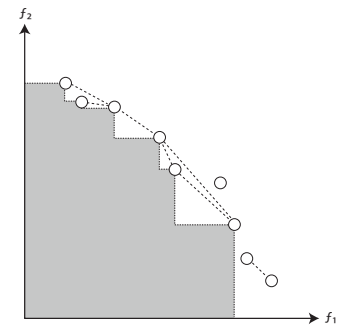


Figure 3.5: Illustration of global features extracted from the connectedness properties of the Pareto set.

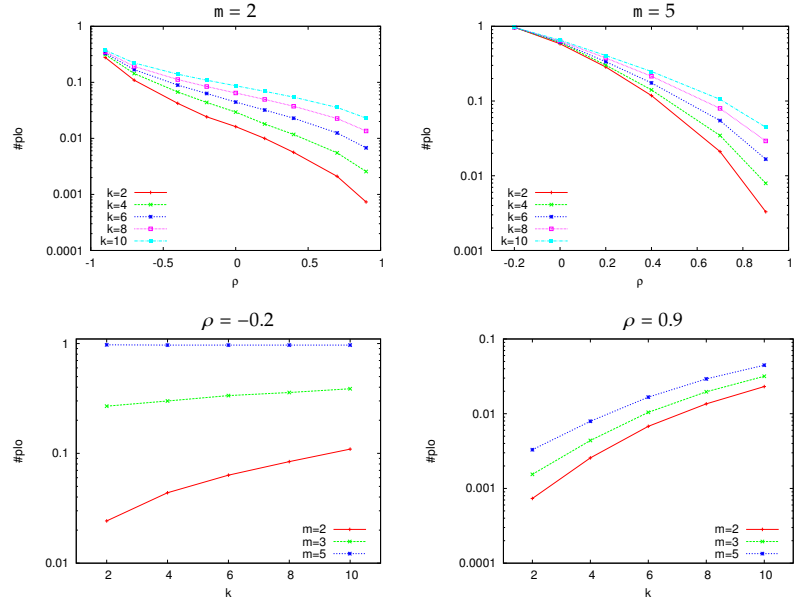


Figure 3.6: Average proportion of Pareto local optimal solutions ($\#plo$) with respect to objective correlation ρ for different k -values (top left $m = 2$, right $m = 5$), and with respect to the number of variable interactions k for different m -values (bottom left $\rho = -0.2$, right $\rho = 0.9$). The problem size is $n = 18$, thus allowing Pareto local optimal solutions to be exhaustively enumerated.

optimization is the number of local optima (Kauffman, 1993). Although multimodality is still largely overlooked in the multi-objective optimization literature, where the number of objectives is seen as the main source of difficulty, few recent studies have revealed its impact on multi-objective search performance (Kerschke et al., 2016; Paquete et al., 2007), including ours (Daolio et al., 2017; Verel et al., 2013). A Pareto local optimal solution is defined as follows by Paquete et al. (2007).

Definition 3.2.2 A Pareto local optimal solution is a solution $x \in X$ for which there does not exist any neighboring solution $x' \in \mathcal{N}(x)$ such that x is dominated by x' .

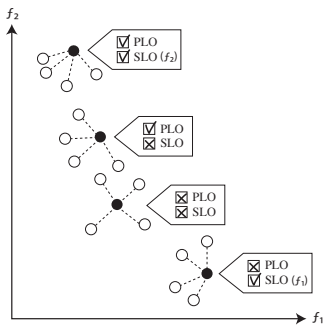


Figure 3.7: Illustration of Pareto local optimal solutions (PLO) and of single-objective local optimal solutions (SLO).

We measure the proportion of Pareto local optimal solutions in the solution space ($\#plo$). Additionally, we also consider the average number of single-objective local optima with respect to each separate objective function, proportional to the size of the solution space ($\#slo_avg$). In other words, $\#slo_avg$ corresponds to the proportion of local optima per objective, all m values (i.e. one per objective) being averaged. We expect $\#slo_avg$ to increase with the number of variable interactions k , as with single-objective nk -landscapes (Kauffman, 1993). We show in Figure 3.6 that $\#plo$ increases with k , but also with the number of objectives and their degree of conflict. Both features then capture different facets of multi-objective multimodality. The definitions of a Pareto local optimal solution and of a single-objective local optimal solution are illustrated in Figure 3.7.

Local Features

Unfortunately, computing the global features introduced above requires the solution space, or the Pareto set, to be exhaustively enumerated, which makes them impractical for large-size problems. Therefore, we consider local features, computed from the neighborhood of

a sample of solutions, which makes them relevant for performance prediction. In the following, we introduce two sampling strategies and a number of landscape measures. We consider a local feature as a combination of both.

Sampling. In single-objective landscape analysis, sampling is often performed by means of a *walk* over the landscape. A walk is an ordered sequence of solutions $(x_0, x_1, \dots, x_\ell)$ such that $x_0 \in X$, and $x_t \in \mathcal{N}(x_{t-1})$ for all $t \in \{1, \dots, \ell\}$ (Kauffman, 1993; Weinberger, 1990).

During a *random walk*, there is no particular criterion to pick the neighboring solution at each step; i.e. a random neighbor is selected. In the single-objective case, the first autocorrelation coefficient of (scalar) fitness values encountered during the random walk characterizes the *ruggedness* of the landscape (Moser et al., 2017; Weinberger, 1990): the larger this coefficient, the smoother the landscape. To accommodate the multi-objective nature of the landscape, different autocorrelation measures will be discussed below. In the case of a random walk, the length of the walk ℓ is a parameter that must be provided beforehand. The longer the length, the better the estimation.

By contrast, during an *adaptive walk*, an improving neighbor is selected at each step, as with a conventional hill climber. In this case, the length ℓ is the number of steps performed until the walk falls into a local optimum. This length is used as an estimator of the diameter of local optima's basins of attraction: assuming isotropy in the solution space, the longer the length, the larger the basins size, hence the lower the number of local optima (Kauffman, 1993). Multiple adaptive walks are typically performed to improve the estimation.

A random walk does not require any adaptation to the multi-objective case, except for the measure used to estimate the correlation coefficient, detailed next. As for the adaptive walk, we consider a very basic single solution-based multi-objective Pareto hill climber (Verel et al., 2013). The Pareto hill climber is initialized with a random solution. At each iteration, the current solution is replaced by a random dominating neighbor until it falls into a Pareto local optimal solution. The number of iterations of the Pareto hill climber is the length of the adaptive walk. The considered random and adaptive walks are illustrated in Figures 3.8 and 3.9, respectively.

Measures. Given an ordered sequence of solutions collected along a walk, we consider the following measures. For each solution from the sample, we explore its neighborhood, and we measure the proportion of dominated ($\#inf$), dominating ($\#sup$), and incomparable ($\#inc$) neighbors, as illustrated in Figure 3.10. We also consider the proportion of non-dominated solutions in its neighborhood ($\#lnd$), as well as the proportion of supported solutions therein ($\#lsupp$). In Figure 3.11, we illustrate some measures based on hypervolume: the average hypervolume covered by each neighbor (hv), the average difference between the hypervolume covered by each neighbor and the one covered by the current solution (hvd), and the hypervolume covered by the whole

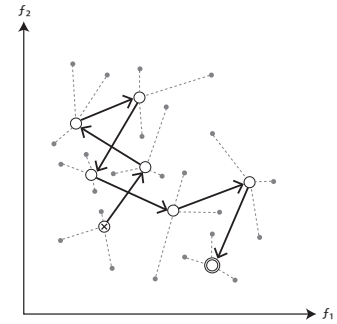


Figure 3.8: Illustration of a multi-objective random walk in the objective space. In this example, the walk length is set to $\ell = 7$.

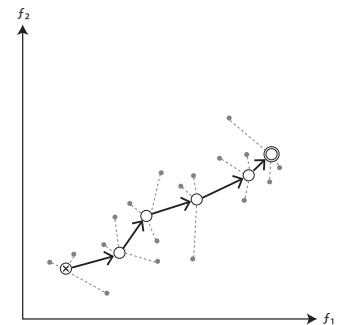


Figure 3.9: Illustration of a multi-objective adaptive walk in the objective space. In this example, the walk performs $\ell = 5$ steps until it falls into a Pareto local optimal solution.

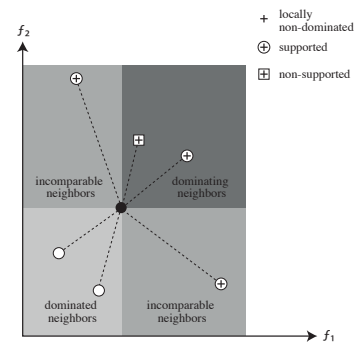


Figure 3.10: Illustration of local dominance measures collected along random and adaptive walks.

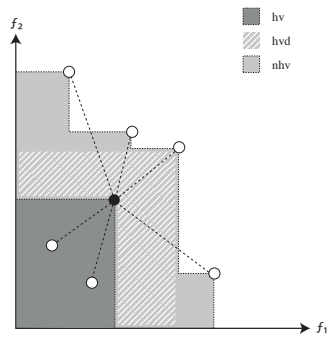


Figure 3.11: Illustration of local hypervolume measures collected along random and adaptive walks.

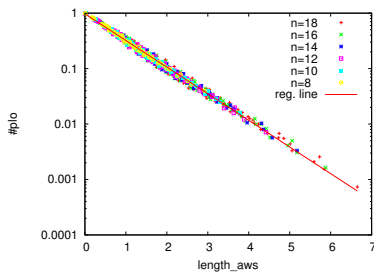


Figure 3.12: Scatter plot of the average proportion of Pareto local optimal solutions ($\#plo$) vs. the average length of 1000 independent multi-objective adaptive walks ($length_aws$) for different n -values. The correlation is 0.997, and the regression is: $\#plo = c \cdot 2^{\alpha \cdot length_aws}$, where $length_aws$ is the average length of adaptive walks, $c = 0.97$ and $\alpha = -1.60$.

neighborhood (nhv). The notions of dominance and hypervolume improvement that can be reached by a solution's neighborhood can be seen as measures of *evolvability* (Smith et al., 2002) for multi-objective optimization.

For samples collected by means of a random walk, we compute both the average value as well as the first autocorrelation coefficient of the measures reported above. Let us consider, for instance, the hv measure. When there is a strong correlation between the hypervolume of neighboring solutions observed at two consecutive steps of the random walk, we argue that it tends to be easier to improve locally by means of neighborhood exploration. On the contrary, when there is no correlation between the hypervolume of neighboring solutions, it is likely harder to improve locally. As such, the corresponding feature might characterize a facet of difficulty for multi-objective landscapes. We also use the random walk sample to estimate the degree of correlation between the objectives (f_cor_rws). The latter is expected to estimate ρ for ρmnk -landscapes. For adaptive walks, we simply compute average values for each measure, as well as walks length ($length_aws$). In Figure 3.12, $length_aws$ is shown to be a sharp estimator for $\#plo$. When the adaptive length is short, the diameter of the basin of attraction associated with a Pareto local optimal solution is short. Assuming that the volume of this basin is proportional to a power of its diameter, the number of Pareto local optimal solutions increases exponentially when the adaptive length decreases. This generalizes known results from single-objective nk -landscapes (Kauffman, 1993).

Summary

The ρmnk -landscape parameters as well as global and local features presented above are summarized in Table 3.1. All benchmark parameters are considered in our analysis, although only the problem size (n) and the number of objectives (m) are available in a black-box scenario. In terms of calls to the evaluation function, the computational complexity for global features from enumeration is $|X| = 2^n$. Given a random walk of length ℓ_{rws} and a neighborhood size η_{neig} , the computational complexity for random walk features is: $1 + (1 + \ell_{rws}) \cdot \eta_{neig} = \Theta(\ell_{rws} \cdot \eta_{neig})$. Similarly, the computational complexity for adaptive walk features is: $\eta_{aws} ((1 + \ell_{aws}) \cdot \eta_{neig} + e_{aws}) = \Theta(\eta_{aws}(\ell_{aws} \cdot \eta_{neig} + e_{aws}))$, where η_{aws} is the number of adaptive walks, ℓ_{aws} is the number of steps before the adaptive walk falls into a Pareto local optimal solution, and e_{aws} is the total number of evaluations performed for the walk to progress. However, we remark that $length_aws$ alone is cheaper to compute, as it does not require any neighborhood exploration apart from the evaluations performed for the walk to progress. Its complexity is just: $\eta_{aws} \cdot e_{aws} = \Theta(\eta_{aws} \cdot e_{aws})$. Similarly, the complexity of f_cor_rws alone is: $1 + \ell_{rws} = \Theta(\ell_{rws})$. We also remark that ℓ_{rws} , η_{aws} and η_{neig} must be defined by the user for feature estimation, whereas ℓ_{aws} and e_{aws} are relative to the landscape being considered. We relate these features with the performance of multi-objective search algorithms below.

Table 3.1: State-of-the-art and proposed multi-objective landscape features considered in our analysis. Features #po, #supp, podist_max, po_ent and fdc come from Knowles and Corne (2003a); #po and #fronts come from Knowles and Corne (2007); hv and #fronts come from Aguirre and Tanaka (2007); #po and fdc come from Garrett and Dasgupta (2008); #fronts comes from Garrett and Dasgupta (2009); #plo comes from Paquete et al. (2007); #cc and #sing come from Paquete and Stützle (2009); #lcc and length_aws come from Verel et al. (2013); podist_avg, hv_r1_rws and hvd_r1_rws come from Daolio et al. (2015, 2017) and Liefvooghe et al. (2013b); others come from Liefvooghe et al. (2020a).

benchmark parameters (4)	
n	number of (binary) variables
k_n	proportional number of variable interactions (epistatic links) : k/n
m	number of objectives
ρ	correlation between the objective values
global features from full enumeration (16)	
#po	proportion of Pareto optimal solutions
#supp	proportion of supported Pareto optimal solutions
hv	hypervolume-value of the (exact) Pareto front
podist_avg	average Hamming distance between Pareto optimal solutions
podist_max	maximal Hamming distance between Pareto optimal solutions (diameter of the Pareto set)
po_ent	entropy of binary variables from Pareto optimal solutions
fdc	fitness-distance correlation in the Pareto set (Hamming dist. in var. space vs. Manhattan dist. in obj. space)
#cc	proportion of connected components in the Pareto graph
#sing	proportion of isolated Pareto optimal solutions (singletons) in the Pareto graph
#lcc	proportional size of the largest connected component in the Pareto graph
lcc_dist	average Hamming distance between solutions from the largest connected component
lcc_hv	proportion of hypervolume covered by the largest connected component
#fronts	proportion of non-dominated fronts
front_ent	entropy of the non-dominated front's size distribution
#plo	proportion of Pareto local optimal solutions
#slo_avg	average proportion of single-objective local optimal solutions per objective
local features from random walk sampling (17)	
#inf_avg_rws	average proportion of neighbors dominated by the current solution
#inf_r1_rws	first autocorrelation coefficient of the proportion of neighbors dominated by the current solution
#sup_avg_rws	average proportion of neighbors dominating the current solution
#sup_r1_rws	first autocorrelation coefficient of the proportion of neighbors dominating the current solution
#inc_avg_rws	average proportion of neighbors incomparable to the current solution
#inc_r1_rws	first autocorrelation coefficient of the proportion of neighbors incomparable to the current solution
#lnd_avg_rws	average proportion of locally non-dominated solutions in the neighborhood
#lnd_r1_rws	first autocorrelation coefficient of the proportion of locally non-dominated solutions in the neighborhood
#lsupp_avg_rws	average proportion of supported locally non-dominated solutions in the neighborhood
#lsupp_r1_rws	first autocorrelation coefficient of the prop. of supported locally non-dominated solutions in the neighborhood
hv_avg_rws	average (single) solution's hypervolume-value
hv_r1_rws	first autocorrelation coefficient of (single) solution's hypervolume-values
hvd_avg_rws	average (single) solution's hypervolume difference-value
hvd_r1_rws	first autocorrelation coefficient of (single) solution's hypervolume difference-values
nhv_avg_rws	average neighborhood's hypervolume-value
nhv_r1_rws	first autocorrelation coefficient of neighborhood's hypervolume-value
f_cor_rws	estimated correlation between the objective values
local features from adaptive walk sampling (9)	
#inf_avg_aws	average proportion of neighbors dominated by the current solution
#sup_avg_aws	average proportion of neighbors dominating the current solution
#inc_avg_aws	average proportion of neighbors incomparable to the current solution
#lnd_avg_aws	average proportion of locally non-dominated solutions in the neighborhood
#lsupp_avg_aws	average proportion of supported locally non-dominated solutions in the neighborhood
hv_avg_aws	average (single) solution's hypervolume-value
hvd_avg_aws	average (single) solution's hypervolume difference-value
nhv_avg_aws	average neighborhood's hypervolume-value
length_aws	average length of adaptive walks

3.3 Landscape Features vs. Search Performance on Small Landscapes

Experimental Setup

Table 3.2: Benchmark parameter values for the set of small ρmnk -landscape instances. 30 instances are independently generated for each setting. A total of 60 480 problem instances is considered.

par.	values
n	{10, 11, 12, 13, 14, 15, 16}
k	{0, 1, 2, 3, 4, 5, 6, 7, 8}
m	{2, 3, 4, 5}
ρ	{-0.8, -0.6, -0.4, -0.2, 0.0, 0.2, 0.4, 0.6, 0.8, 1.0} s.t. $\rho > \frac{-1}{m-1}$

As detailed in Table 3.2, we consider small-size ρmnk -landscapes with a problem size $n \in \{10, 11, 12, 13, 14, 15, 16\}$ in order to enumerate the solution space exhaustively, as required by the computation of global features; a number of variable interactions (epistatic degree) $k \in \{0, 1, 2, 3, 4, 5, 6, 7, 8\}$, from linear to highly rugged landscapes; a number of objectives $m \in \{2, 3, 4, 5\}$, from bi-, to multi- and many-objective instances; and an objective correlation $\rho \in \{-0.8, -0.6, -0.4, -0.2, 0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$ such that $\rho > -1/(m-1)$. We generate 30 landscapes independently at random for each combination of instance settings. This represents a dataset of 60 480 small-size landscapes in total, exhibiting a large span of problem characteristics. For local features, we perform one random walk of length $\ell = 1\,000$, and $\eta_{\text{aws}} = 100$ independent adaptive walks, per instance. As in single-objective landscape analysis (Kauffman, 1993), multiple adaptive walks are performed to account for the stochasticity observed in their length, whereas a single long random walk is performed to obtain a large sample to better estimate the autocorrelation coefficients. For features based on hypervolume, given that all ρmnk -landscape’s objectives have a similar range and take their values in $[0, 1]$, we set the reference point to the origin. In terms of algorithms, we perform 30 independent runs of both G-SEMO and I-PLS on each instance. We are interested in the approximation quality found by each algorithm after reaching a maximum budget, here defined as a number of calls to the evaluation function. The stopping condition is set to a fixed budget of 10% of the solution space size, i.e. $\lceil 0.1 \cdot |X| \rceil = \lceil 0.1 \cdot 2^n \rceil$ calls of the evaluation function. This represents a budget of 103 evaluations for $n = 10$, up to 6 554 evaluations for $n = 16$. Performance quality is measured in terms of the multiplicative epsilon indicator (Zitzler et al., 2003), that is the epsilon approximation ratio to the exact Pareto front.

Correlation among Landscape Features

Figure 3.13 reports the correlation matrix and a hierarchical clustering of all features, as measured on the complete dataset of small-size instances. This highlights the similarities between features and their main association with either benchmark parameters: it is worth noticing that each cluster contains a benchmark parameter, as well as both global and local features.

Cluster associated with ruggedness (violet). All of the eight landscape features from the first autocorrelation coefficient of random walks measures strongly correlate with the proportional number of variable interactions (epistatic links) of ρmnk -landscapes ($k_n = k/n$).

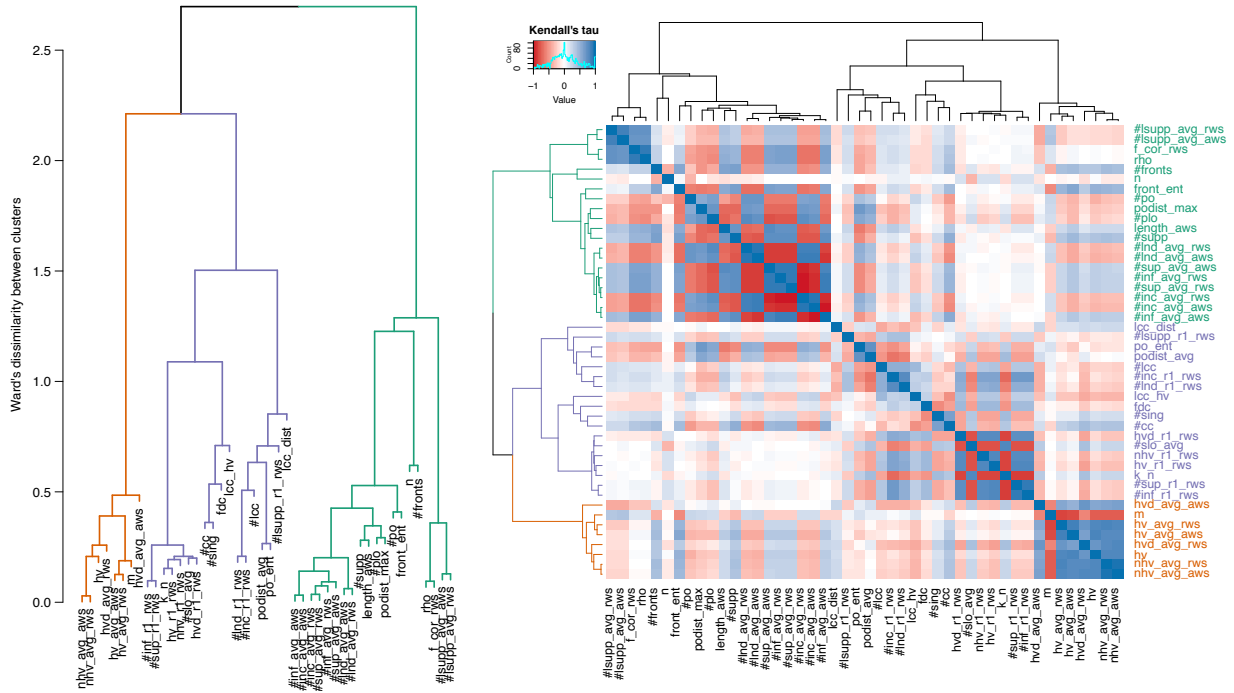


Figure 3.13: Features clustering (left) and features association (right) computed over the whole set of small instances. The strength of monotonic association between each pair of features is measured by the Kendall coefficient τ , which is a rank-based non-linear correlation measure. The distance between each pair of features is defined as $1 - |\tau|$. Ward’s hierarchical clustering (Murtagh and Legendre, 2014) is performed according to such distance (left figure, y-axis): the lower the split, the closer the features. Clustering is also used to reorder rows and columns of the symmetrical correlation matrix (Wardes et al., 2016) (right figure, see color legend in the middle): the darker the color, the higher the strength of association between the corresponding features. By cutting the clustering tree, we can group together the features that are more associated with each one of the benchmark parameters $\{\rho, m, n, k_n\}$; see branches and row label colors: green for ρ , violet for k_n , and orange for m .

Intuitively, those features are related to the ruggedness of the multi-objective landscape, which generalizes known results from single-objective landscape analysis (Weinberger, 1990): the ruggedness of the landscape increases with k_n . As in single-objective optimization, the average number of local optima per objective $\#s_{lo_avg}$ also correlates to k_n . All the features related with connectedness ($\#cc$, $\#sing$, $\#lcc$, lcc_dist , lcc_hv) belong to this same cluster, together with other features related to the distance between Pareto optimal solutions ($podist_avg$, po_ent , fdc), although the correlation with k_n is lower in this case.

Cluster associated with the number of objectives (orange). The features related to hypervolume that do not belong to the previous cluster (associated with ruggedness) are all negatively correlated with the number of objectives (m). Interestingly, features based on average hypervolume measures (hv , hvd , nhv) are closely related to one another, for samples from both random and adaptive walks. This means that the landscape evolvability, in terms of hypervolume, decreases with the objective space dimension, and so does the Pareto front hypervolume.

Cluster associated with objective correlation (green). This last cluster contains the highest number of features, all related to the correla-

tion among the objective values (ρ). We note that `f_cor_rws` is shown to highly correlate with ρ , and can thus be used as an estimator for black-box instances, for which ρ is typically unknown. Objective correlation seems to impact both the shape and the cardinality of the Pareto front (`#po`, `#supp`, `podist_max`). Similarly, local features based on dominance (`#inf`, `#inc`, `#sup`) are close to one another, both for random and adaptive walks. More interestingly, the proportion of Pareto local optimal solutions (`#plo`) and its estimator `length_aws` both belong to this cluster. Although `#slo_avg` belongs to the first cluster associated with ruggedness (see above), `#plo` seems to increase with the degree of conflicts between the objectives. Indeed, the objective correlation directly impacts the probability of dominance: the larger ρ , the smaller the chance to have a dominated or dominating neighbor, and the larger the chance to have an incomparable one, which directly impacts the number of Pareto local optimal solutions. The problem size `n` is also contained in this cluster, although it is only slightly correlated to other features, except for the proportional number of fronts (`#fronts`).

Feature-based Performance Prediction

To investigate the association between landscape features and empirical problem hardness, we build a regression model that predicts search performance based on different subsets of input features. More precisely, we predict the expected multiplicative epsilon indicator value reached by G-SEMO and I-PLS based on: all features, global features, local features, local features based on random walk, local features based on adaptive walk, benchmark parameters, and problem parameters available in a black-box scenario. Given the non-linearity observed in the data, we chose a tree-based regression model: an ensemble of extremely randomized trees (Geurts et al., 2006). It is a variant of the popular random forest model (Breiman, 2001) that differs in the way individual trees are built. While splitting a node, we do not only randomize the choice of input variable, but also the cut-point. Moreover, each tree uses the entire training data, rather than bootstrap replicas. In our experiments, we employ ensembles of 500 unpruned regression trees (Pedregosa et al., 2011). The prediction target is the approximation ratio to the exact Pareto front, measured every tenth of the total evaluations budget. That is, we model the search convergence curve with a multi-output regression. The mean square error (MSE), mean absolute error (MAE), coefficient of determination (R^2) and adjusted R^2 of the regression model for different sets of predictors are reported in Table 3.3. A score is the average score over the multiple outputs of a model. The closer MSE and MAE are to 0.0, the better. Conversely, R^2 reaches 1.0 when the predictions are perfect, and would be 0.0 for a constant model that always predicts the global average of the target value, irrespective of the input features. For each measure of accuracy, we report the average value on test and its standard deviation over a 10-fold cross-validation.

A general observation is that the MAE and the MSE are in accordance with each other, as shown by the relative ranking of each subset of features. The rank reflects any significant statistical difference on MAE

Table 3.3: 10-fold cross-validated regression accuracy obtained on small instances for different input features.

algo.	set of features	MAE		MSE		R ²		adjusted R ²		rank
		avg	std	avg	std	avg	std	avg	std	
G-SEMO	all features	0.007781	0.000055	0.000118	0.000002	0.951609	0.001463	0.951238	0.001474	1
	global features	0.008411	0.000064	0.000142	0.000003	0.943046	0.001665	0.942876	0.001670	2
	local features	0.009113	0.000072	0.000161	0.000003	0.932975	0.001555	0.932663	0.001562	3
	local features (random walk)	0.009284	0.000081	0.000167	0.000003	0.930728	0.001605	0.930510	0.001610	4
	local features (adaptive walk)	0.010241	0.000106	0.000195	0.000004	0.917563	0.002260	0.917399	0.002264	5
	{ ρ, m, n, k_n }	0.010609	0.000110	0.000215	0.000004	0.911350	0.002436	0.911292	0.002372	6
	{ m, n }	0.032150	0.000309	0.001545	0.000025	0.340715	0.011217	0.340497	0.011220	7
I-PLS	all features	0.008043	0.000052	0.000127	0.000002	0.944367	0.001429	0.943940	0.001440	1
	global features	0.008613	0.000054	0.000149	0.000002	0.936046	0.001479	0.935856	0.001484	2
	local features	0.009297	0.000081	0.000167	0.000003	0.925610	0.001900	0.925264	0.001909	3
	local features (random walk)	0.009485	0.000089	0.000173	0.000004	0.923032	0.001863	0.922789	0.001869	4
	local features (adaptive walk)	0.010336	0.000098	0.000198	0.000004	0.910670	0.002455	0.910493	0.002459	5
	{ ρ, m, n, k_n }	0.010817	0.000122	0.000223	0.000005	0.901888	0.002803	0.901823	0.002882	6
	{ m, n }	0.030523	0.000286	0.001423	0.000023	0.351707	0.009822	0.351493	0.009826	7

and MSE over the holdouts of each cross-validation iteration, with respect to a Mann-Whitney statistical test at a significance level of 0.05 with Bonferroni correction for multiple comparisons (Conover, 1999). In addition, when comparing G-SEMO and I-PLS, we observe almost no difference in the models accuracy. When analyzing the impact of the different subset of features, we can observe a poor performance when using solely m and n as input variables. This means that the problem input provided in a black-box scenario, i.e. the variable and objective space dimensions, is not sufficient to explain the performance of G-SEMO or I-PLS. Once we take the objective correlation ρ into account, and more importantly the proportional number of variable interactions k_n , we observe a significant increase in the model accuracy. For both algorithms, the R^2 exceeds 0.9: more than 90% of the variance in search performance between instances is explained by the ρmnk -landscape parameters. This is not a surprise since these four parameters define the way ρmnk -landscapes are constructed; see Section 2.2. However, let us remind that ρ and k_n are *unknown* in practice when solving a black-box problem instance. More interestingly, however, we see that the proposed local features, based on sampling, allow the model to obtain a better prediction accuracy than benchmark parameters. We attribute this to the fact that they are able to capture the variations between instances with the same parameters; i.e. the randomness in the construction of ρmnk -landscapes. This is particularly true for local features based on random walk, which contain more insightful information for search performance than the ones based on adaptive walk. Indeed, the regression accuracy obtained with the former subset of local features is almost as good as the combination of both. At last, we observe that global features, based on the enumeration of the solution space, obtain a better ranking, although the addition of local features seems to increase the predictive power of the regression model even more, as illustrated by the results obtained by the model using all features as input variables.

Importance of Features for Search Performance

Tree-based models also allow for the identification of which input features are the most important to make accurate predictions, which

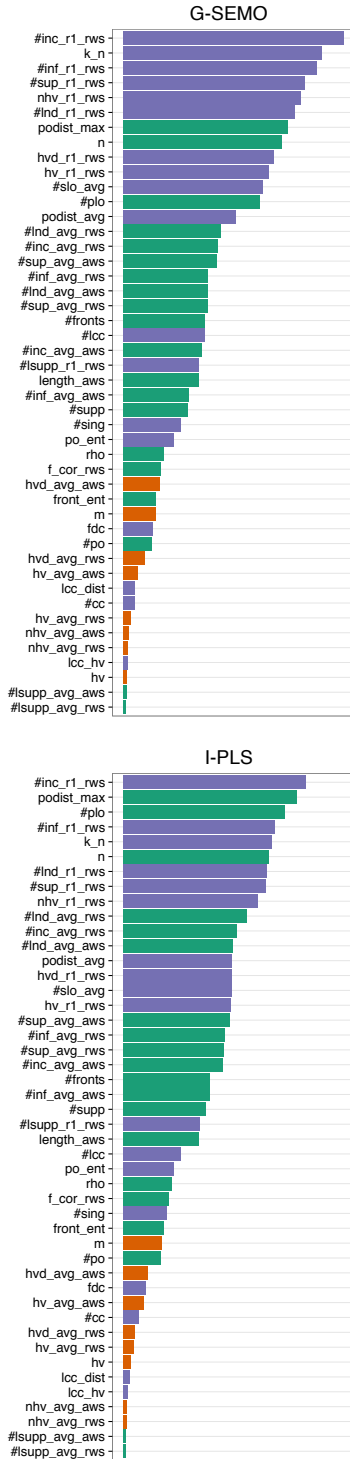


Figure 3.14: Relative importance of features (mean decrease in node impurity) from totally-randomized regression trees for small landscapes.

provides insight into the process being modeled (Louppe et al., 2013). In particular, we consider the measure of importance that relates to the decrease in node impurity after each split on a given predictor; the larger the decrease, the more important the predictor. Note that, in the regression case, node impurity is measured by variance. We derive our estimates from a large ensemble of 50 000 shallow and totally-randomized regression trees. Choosing the input variable to split on totally at random prevents correlated variables to mask one another, which would result in underestimating their relative importance (Louppe et al., 2013). Then, by using small trees, we strive to minimize the effect of having a finite sample set, which introduces noise in the node impurities as trees grow. The relative importance of features thus extracted, is depicted in Figure 3.14. For a given algorithm, features are sorted in decreasing order of importance, from top to bottom. Although the regression accuracy is similar for both algorithms, the most important features are different for G-SEMO and I-PLS.

For G-SEMO, the six most important features are all related to the ruggedness of the landscape (in violet). Apart from the proportional number of variable interactions k_n in ρmnk -landscapes, the others correspond to the first autocorrelation coefficient of the proportional number of dominated ($\#inf$), dominating ($\#sup$), and incomparable ($\#inc$) neighbors, the proportional number of non-dominated solutions in the neighborhood ($\#lnd$), and the hypervolume covered by the neighborhood (nhv) encountered along a random walk. Next in the ranking are those associated with objective correlation and dominance (in green), such as the diameter of the Pareto set ($podist_max$), which also correspond to the most important global feature. For I-PLS, features related to the ruggedness (in violet) and to the objective correlation (in green) seem equally important, and the features listed above also appear to be impactful. Most notably, the proportion of Pareto local optimal solutions ($\#plo$) seems of high importance; it appears in the 3rd place for I-PLS and only in the 12th place for G-SEMO. By contrast, the features associated with the number of objectives (in orange) are of low importance for the two algorithms. Interestingly, for both G-SEMO and I-PLS, the most important benchmark parameter is the proportional number of variable interactions k_n , followed by the problem size n , the objective correlation ρ , and only finally the number of objectives m .

3.4 Scaling to Large Landscapes

In this section, we extend our analysis to *large* instances. Since global features cannot be computed anymore, we investigate the ability of *local* features to explain algorithm performance for large dimensions.

Experimental Setup

As given in Table 3.4, we generate large-size ρmnk -landscapes by means of a design of experiments based on random latin hypercube sampling (Carnell, 2020). We consider problem sizes in the range $n \in \llbracket 64, 256 \rrbracket$, numbers of variable interactions $k \in \llbracket 0, 8 \rrbracket$, numbers of objectives $m \in \llbracket 2, 5 \rrbracket$, and objective correlations $\rho \in]-1, 1]$ such that $\rho > -1/(m-1)$. A total of 1 000 problem instances is considered, a single instance being generated for each parameter setting provided by the design of experiments. We consider all local features as well as benchmark parameters, and the same two multi-objective algorithms. We perform 30 independent runs per instance and per algorithm, with a fixed budget of 100 000 calls to the evaluation function. The performance quality is measured in terms of the multiplicative epsilon indicator to the best-known non-dominated set.

Table 3.4: Benchmark parameter values for the set of large ρmnk -landscape instances. One random instance is generated for each setting. A total of 1 000 problem instances is considered.

parameter	values
n	$\llbracket 64, 256 \rrbracket$
k	$\llbracket 0, 8 \rrbracket$
m	$\llbracket 2, 5 \rrbracket$
ρ	$]-\frac{1}{m-1}, 1]$

Correlation among Landscape Features

As performed in the previous section for small instances, we report the correlation between each pair of features and the corresponding clustering in Figure 3.15. Similar to our previous results, we obtain three clusters, each one being associated with one benchmark parameter.

The local features related to the first autocorrelation coefficient measured on random walks remain in the cluster associated to ruggedness (in violet), as it was with small instances. In addition, both features measuring the average difference between the hypervolume covered by each neighbor and the one covered by the current solution (hvd

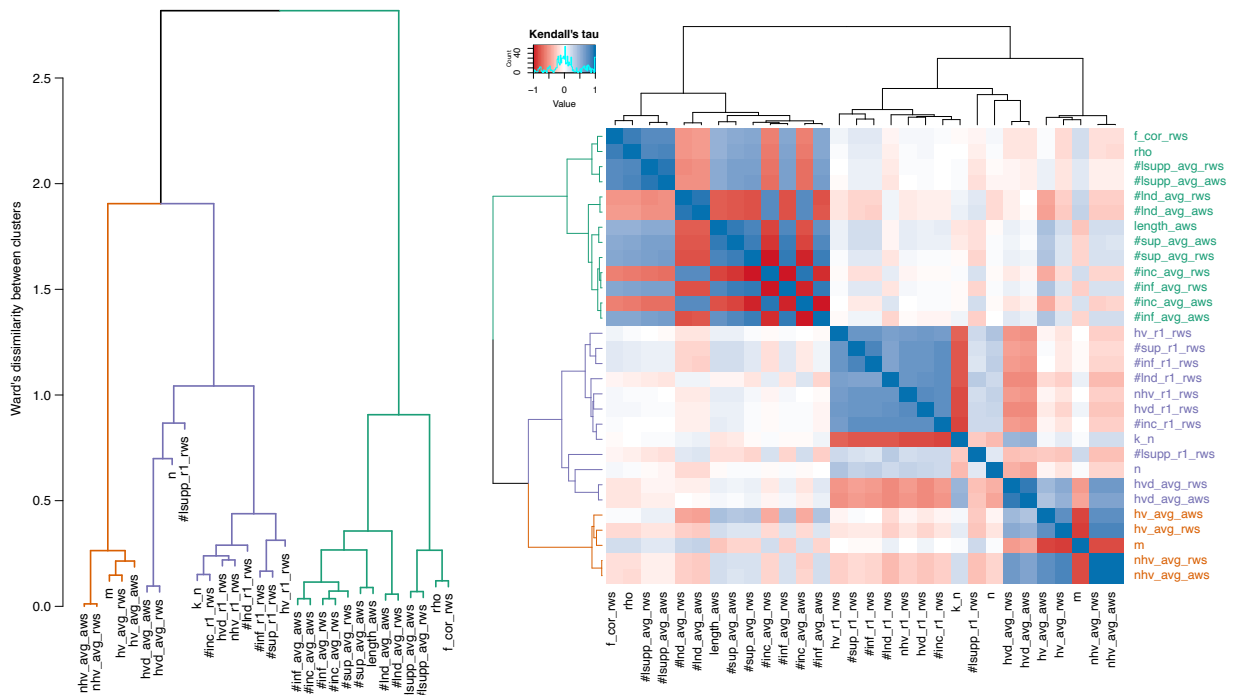


Figure 3.15: Features clustering (left) and features association (right) computed over the whole set of large instances; see Figure 3.13 for details.

Table 3.5: Random subsampling cross-validated regression accuracy obtained on large instances (50 iterations, 90/10 split).

algo.	set of features	MAE		MSE		R ²		adjusted R ²		rank
		avg	std	avg	std	avg	std	avg	std	
G-SEMO	all features	0.003049	0.000285	0.000017	0.000004	0.891227	0.024584	0.843934	0.035273	1
	local features	0.003152	0.000295	0.000018	0.000004	0.883909	0.026863	0.838126	0.037457	1
	local features (random walk)	0.003220	0.000314	0.000019	0.000004	0.878212	0.028956	0.849287	0.035833	1.5
	local features (adaptive walk)	0.003525	0.000329	0.000023	0.000006	0.854199	0.032339	0.834089	0.036799	5
	{ ρ , m , n , k_n }	0.003084	0.000270	0.000017	0.000003	0.892947	0.020658	0.888440	0.021528	1
	{ m , n }	0.010813	0.000830	0.000206	0.000030	-0.303336	0.188046	-0.330209	0.191923	6
I-PLS	all features	0.004290	0.000430	0.000034	0.000008	0.886568	0.026980	0.837249	0.038710	1
	local features	0.004359	0.000423	0.000035	0.000008	0.883323	0.027274	0.837309	0.038030	1
	local features (random walk)	0.004449	0.000394	0.000036	0.000008	0.879936	0.026335	0.851421	0.032589	1
	local features (adaptive walk)	0.004663	0.000403	0.000039	0.000008	0.871011	0.025903	0.853219	0.029476	3.5
	{ ρ , m , n , k_n }	0.004353	0.000320	0.000033	0.000006	0.889872	0.024505	0.885235	0.025537	1
	{ m , n }	0.016959	0.001473	0.000472	0.000077	-0.568495	0.228629	-0.600836	0.233343	6

moved to this cluster, but their correlation with the other features in the cluster is low. Similar observations can be made for the problem size n . We attribute this to the design of experiments of this new dataset for large instances. Once again, the features related to dominance (in green) are all very close to one another. They relate very much to the objective correlation (ρ) and to the proportion of Pareto local optimal solutions ($\#plo$). As with small instances, features related to hypervolume are correlated with the number of objectives (m , in orange). Overall, there are no major changes with respect to the previous dataset, which validates our study on small instances.

Feature-based Performance Prediction

The prediction accuracy of regression models predicting search performance for different subsets of input variables is reported in Table 3.5. Overall, the fitting quality is lower than for small instances. We attribute this to the smaller number of observations contained in the dataset for large instances (1 000 large instances, against 60 480 small instances). Once again, the results for G-SEMO and I-PLS are quite similar. As before, the objective correlation ρ and the proportional number of variable interactions k_n , which are unknown in a black-box scenario, are essential to understand search performance and to reach a good prediction accuracy. Surprisingly, the model using solely the variable space and objective space dimensions, n and m , has a negative R^2 , and performs worse than a model that always predict the average performance value. In this case, observed and predicted values can actually be far from each other, in particular for instances where algorithms are efficient. All other models obtain an R^2 larger than 0.8. This means that more than 80% of the variance in the algorithm's performance is explained by local features. The set of all local features has a similar predictive power than (known and unknown) benchmark parameters. Let us remind that in this dataset, a single instance is generated per instance setting, so that there is no variance between instances with the same parameters. As with small instances, the local features based on random walks have a higher predictive power than those based on adaptive walks, although the combination of both is always more accurate. Ultimately, local features allow the regression model to obtain a satisfying prediction accuracy. We ana-

lyze the importance of local features below, and then we study their relevance in the context of automated algorithm selection.

Importance of Features for Search Performance

The importance of features for both algorithms is reported in Figure 3.16. For G-SEMO, features related to ruggedness (in violet) are more relevant than others, followed by features related to objective correlation, such as the estimator of the proportion of Pareto local optimal solutions (`length_aws`), and to a smaller extent, to features that are associated with the number of objectives (`m`). Interestingly, for I-PLS, features related to dominance and objective correlation (in green) are clearly much more informative. Indeed, the average rank of those features is 7 for I-PLS, whereas it is only 19.08 for G-SEMO. Conversely, the average rank of features related to ruggedness (in violet) is 10.33 for G-SEMO, against 23.67 for I-PLS. For both algorithms, the average rank of features related to the number of objectives (in orange) is about the same, and the second most important one (16.6 for G-SEMO, against 18 for I-PLS). This highlights that landscape features impact local and global dominance-based multi-objective search algorithms differently.

3.5 Landscape-aware Automated Algorithm Selection

We conclude our analysis with a feature-based automated algorithm selection for large-size ρmnk -landscapes using a portfolio of three multi-objective search algorithms, namely NSGA-II (Deb et al., 2002), IBEA (Zitzler and Künzli, 2004) and MOEA/D (Zhang and Li, 2007), as described in Section 2.3. They were chosen as representatives of the state-of-the-art in the field, covering dominance-, indicator- and decomposition-based approaches, respectively. We rely on an out-of-the-box implementation with default parameters, as provided in jMetal 4.5 (Durillo and Nebro, 2011).

All three algorithms use a population of size 100, a 1-point crossover with a rate of 0.9, and a bit-flip mutation with a rate of $1/n$, under a fixed budget of 1 000 000 evaluations. Notice that the dataset contains 999 observations: one instance was discarded as there was no distinction among the algorithms. In order to predict the best-performing algorithm for solving a given instance, we build an ensemble of 500 extremely randomized *classification* trees, in contrast to the regression models discussed so far. The output class is simply whether (i) NSGA-II, (ii) IBEA, or (iii) MOEA/D performs better, on average, for a given instance, in terms of hypervolume. The classification accuracy, measured in terms of a cross-validated error rate, is reported in Table 3.6. In fact, we report two error rates. In the error rate of best average performance, an error is taken into account if the predicted algorithm differs from the best performing algorithm on average. Complementarily, in the error rate of best statistical rank, an error is taken

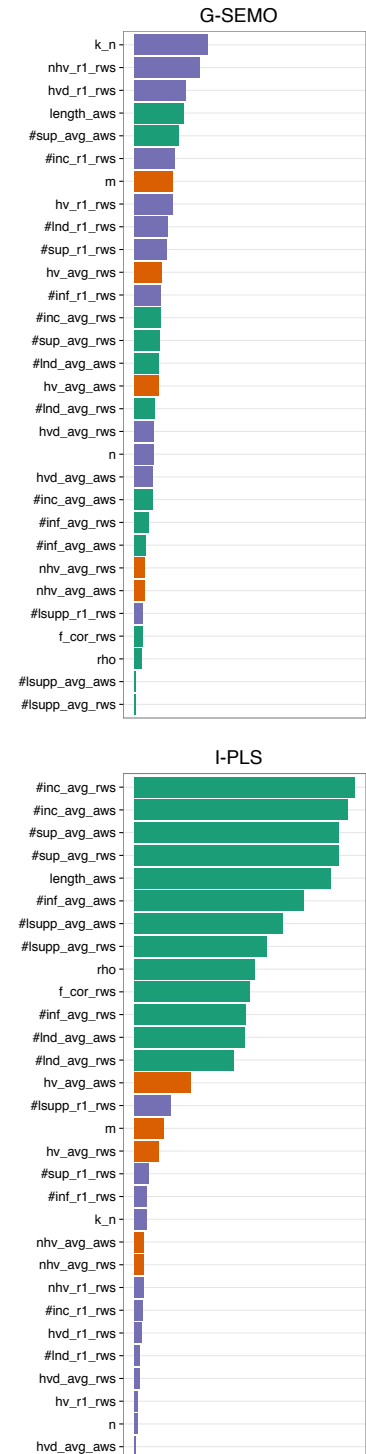


Figure 3.16: Relative importance of features (mean decrease in node impurity) from totally-randomized regression trees for large landscapes.

Table 3.6: Random subsampling cross-validated classification accuracy obtained on large ρmnk -landscapes (50 iterations, 90/10% split).

set of features	error rate of best average performance			error rate of best statistical rank		
	mean	std	rank	mean	std	rank
all features	0.122222	0.031033	1	0.012727	0.014110	1
local features	0.123030	0.030521	1	0.013737	0.014103	1
local features (random walk)	0.118788	0.029187	1	0.013333	0.012149	1
local features (adaptive walk)	0.130303	0.029308	1	0.015354	0.014026	1
{ ρ, m, n, k_n }	0.125859	0.028875	1	0.014141	0.013382	1
{ m, n }	0.413333	0.045533	6	0.197374	0.043778	6

into account only if the predicted algorithm is significantly outperformed by any other according to a Mann-Whitney statistical test at a significance level of 0.05 with Bonferroni correction.

Overall, the feature-based classification models are able to reach an error rate below 0.131 for the best average performance and below 0.016 for the best statistical rank. As such, one of the significantly best-performing algorithms is predicted in more than 98.4% of the cases. That is significantly more accurate than the basic approach based on the variable and objective space dimensions (n and m), which has an error rate of about 41.3% for the best average performance, and of 19.7% for the best statistical rank. Notice that a naive approach that always chooses the best algorithm on average (in our case, NSGA-II) has an error rate of more than 50%, while always selecting the algorithm with the best statistical rank (in our case, MOEA/D) would result in more than 12% of errors. We did not find any statistical difference between all other classification models, although the model with the lowest error always uses local features. Note that models built on features from random walks only are almost as good as any other model: this might provide a viable option to reduce the computational cost of the portfolio without altering much the prediction accuracy.

For the sake of providing a model that is easier to interpret, we construct another classifier based on a simple decision tree (Breiman et al., 1984; Therneau and Atkinson, 2018), as illustrated in Figure 3.17. Even with such a simple decision tree of depth three, the proposed features are able to distinguish between the algorithms with a cross-validated error rate on best average performance of 12.61%. The root of the decision tree is a feature related to the objective correlation ($\#lnd_avg_aws$), measured in terms of the proportion of locally non-dominated neighbors encountered along an adaptive walk. When there are few non-dominated solutions in the neighborhood, NSGA-II has more chances of being selected. This typically happens when the objectives are correlated. Indeed, on the left-hand side of the tree, NSGA-II outperforms IBEA and MOEA/D on 469 instances, whereas it is outperformed only 196 times. On the contrary, when there are more non-dominated solutions, MOEA/D shall be selected, as it performs better on 308 instances, against 26 for the other algorithms. In order to reduce the error rate on the left-hand side of the decision tree, two features are considered (hvd_avg_rws and $\#lnd_avg_rws$), both related to ruggedness. Roughly speaking, MOEA/D shall be preferred over NSGA-II for correlated objectives only when the landscape is relatively smooth. Overall, this emphasizes that a single feature is

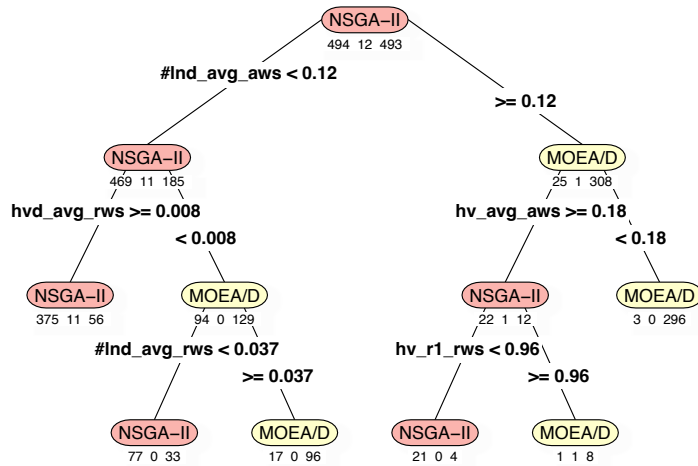


Figure 3.17: CART decision tree for automated algorithm selection on large landscapes. The nodes report the number of instances where NSGA-II, IBEA, and MOEA/D performs better on average, from left to right, respectively.

not enough to distinguish between the different algorithms, and that multiple features, in this case related to ruggedness and objective correlation, are required to design an accurate portfolio approach. This simple example illustrates the potential of automated algorithm selection based on multi-objective landscape features for large dimensions. Additional results on large-size multi-objective quadratic assignment problem instances are discussed in the next section.

In Yap et al. (2020), a complementary study conducted on the same dataset of large-size ρ mnk-landscapes further visualizes the relationship between landscape features and algorithm performance, following the instance space analysis methodology from Smith-Miles et al. (2014). An algorithm selection model based on support vector machines reaches a similar level of accuracy.

3.6 A Glimpse on Further Contributions

In this section, we give a brief overview of contributions related to landscape analysis and automated algorithm selection for multi-objective optimization.

Landscape Analysis and Automated Algorithm Selection for other Problem Classes

Under the general methodology defined above, we investigate the properties of *large-size* multi-objective quadratic assignment problems (mQAP) and how they impact the performance of multi-objective search algorithms (Liefvooghe et al., 2020b). The mQAP differs from ρ mnk-landscapes in multiple aspects, most notably the solution representation which is based on permutations and not binary strings. The landscape of a diversified dataset of mQAP instances is characterized by means of the previously-identified *local* features. As detailed in Table 3.7, we generate 1 000 large-size mQAP instances with the

Table 3.7: Benchmark parameter values for the set of (large-size) mQAP instances. One random instance is generated for each setting. A total of 1 000 mQAP instances is considered.

parameter	values
n	$\llbracket 30, 100 \rrbracket$
m	$\llbracket 2, 5 \rrbracket$
ρ	$[-1, 1]$
type	{uni, rl}

Table 3.8: Classification error obtained with different subset of features, measured on random subsampling cross-validation (100 repetitions, 80/20% split). Two values are reported: the error rate in predicting the algorithm with the best performance on average, and the error rate in predicting an algorithm that is not statistically outperformed by any other, according to a Mann-Whitney test at a significance level of 0.05 with Bonferroni correction. The dummy classifier always returns the most frequent best (in this case, MOEA/D).

set of features	classification error	error predicting stat. best
all features	.1078	.0063
local features	.1077	.0063
local features (adaptive walk)	.1125	.0065
local features (random walk)	.1114	.0062
{type, n, m, ρ }	.1197	.0072
{n, m}	.1962	.0332
random classifier	.6667	.3810
dummy classifier	.4200	.1040

generator from Knowles and Corne (2003a), following a design of experiments based on random latin hypercube sampling (Carnell, 2020). We consider a problem size in the range $n \in \llbracket 30, 100 \rrbracket$, a number of objectives $m \in \llbracket 2, 5 \rrbracket$, an objective correlation $\rho \in [-1, 1]$, and two instance types: uniform (uni) and real-like (rl). We notice that, although the problem size and the number of objectives are given, the type and the objective correlation are *unknown* in practice for unseen mQAP instances. The strengths and weaknesses of multi-objective search algorithms are then highlighted by relating their expected performance in view of landscape features. We consider the same algorithms (NSGA-II, IBEA and MOEA/D) with a population of size 100, an exchange mutation with a rate of 0.2, and a partially-mapped crossover (Goldberg, 1989) with a rate of 0.95. Each algorithm stops after 1 000 000 evaluations, and is executed 20 times per instance. We measure algorithm performance in terms of hypervolume relative deviation. Once again, our study on mQAP highlights that algorithms are not only impacted by the number of objectives, but that the ruggedness and multimodality of the multi-objective landscape are also crucially important to properly explain multi-objective search performance.

We also investigate the performance of a feature-based automated algorithm selection approach. Table 3.8 reports the error rates obtained by a random forest classification model (Breiman, 2001; Liaw and Wiener, 2002) based on different subsets of features. The feature-based models can predict the algorithm with the best average performance in about 90% of the cases, and an algorithm which is not statistically outperformed by any other in more than 99% of the cases. This is significantly

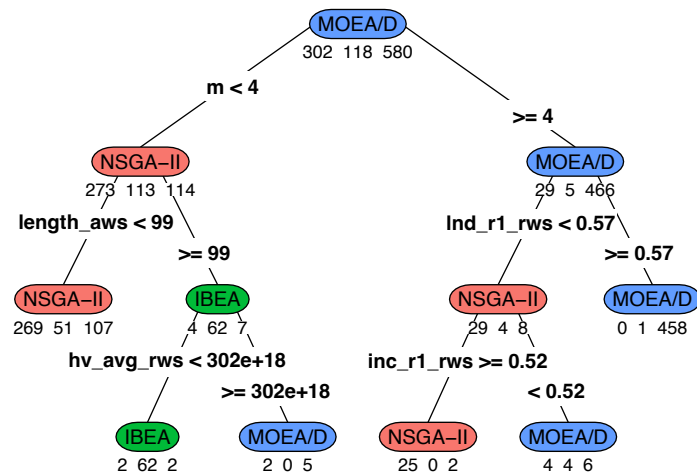


Figure 3.18: CART decision tree for automated algorithm selection on mQAP. The nodes report the number of instances where NSGA-II, IBEA, and MOEA/D performs better on average, from left to right, respectively.

more accurate than a random classifier, a dummy classifier that always predicts the most-frequent best algorithm (here, MOEA/D), and a classifier based on benchmark parameters only. A simple decision tree for mQAP is also provided in Figure 3.18, with an error rate of 17.5%.

By paying a particular attention to the cost of features, and by relying on low-cost features, we now deduce the budget allocated to the computation of features from the budget allocated to multi-objective search algorithms. As reported in Figure 3.19, the statistical rank of a feature-based automated evolutionary multi-objective optimization algorithm selection method (AUTO-EMOA for short) is 0.09, more than three times lower than the best standalone approach (MOEA/D, with 0.29). Among all instances seen during cross-validation, it was *not* significantly outperformed by any other approaches on 92% of the cases (82% for MOEA/D). As such, deducing a small part of the budget allocated to the search process for feature computation proves to be beneficial in order to gain knowledge about the tackled problem, and make a better-informed recommendation about the appropriate multi-objective search strategy to apply for solving it.

Regarding other problem classes, let us comment that preliminary studies about landscape analysis and algorithm design and selection for small-size bi-objective flowshop scheduling and unconstrained binary quadratic programming problems are reported in [Liefoghe et al. \(2017a, 2015b\)](#).

Connectedness of the Pareto Set

As mentioned in Section 3.2, *connectedness* is a particular multi-objective landscape property that relates to the Pareto set structure and to the neighborhood relation (Ehrgott and Klamroth, 1997; Gorski et al., 2011). For a given multi-objective landscape, a graph can be constructed such that each node represents a Pareto optimal solution, and an edge connects two nodes if the corresponding solutions are neighbors. The Pareto set is connected if the underlying graph is also connected, i.e. there is a path between any pair of nodes. As such, when connect-edness holds, a simple local search such as Pareto local search (Paquete et al., 2007) is able to identify the whole Pareto set by starting with at least one Pareto optimal solution, albeit its computational performance is obviously impacted by the tractability of the Pareto set — and of the neighborhood relation. Theoretical and empirical results about the intractability and the connectedness of the Pareto set for some multi-objective combinatorial optimization problem classes are reported in Table 3.9.

In [Liefoghe et al. \(2013a\)](#), we investigate the connectedness of three bi-objective knapsack problem variants with respect to simple neighborhood structures. Based on this property, a Pareto local search (PLS) algorithm is proposed and its performance is compared against exact algorithms in terms of runtime and approximation quality. Our experiments indicate that PLS is able to find a representative set of Pareto optimal solutions on most cases, and in much less time than exact dynamic programming algorithms. A summary of results is reported in

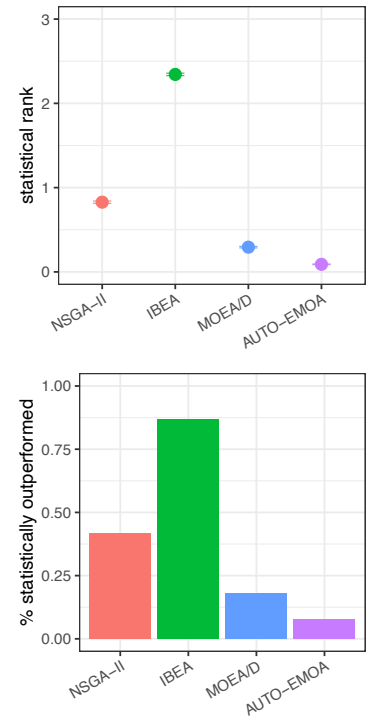


Figure 3.19: Performance of automated algorithm selection (AUTO-EMOA) compared against other algorithms for mQAP.

Table 3.9: Intractability and connectedness properties for some multi-objective combinatorial optimization problems. Given a property, either it is theoretically proven as true (✓) or false (✗), or there is empirical evidence that it is true (+) or false (−). When a ‘+’ is used for connectedness, it means that Pareto optimal solutions are highly clustered. For the knapsack problem with binary weights, the Pareto front is tractable, but the (complete) Pareto set is not.

problem class	intractable		connected	
shortest path problem	✓	Hansen (1979)	✗	Gorski et al. (2011)
spanning tree problem	✓	Hamacher and Ruhe (1994)	✗	Gorski et al. (2011)
assignment problem	✓	Ehrgott (2005)	✗	Gorski et al. (2011)
travelling salesperson problem	✓	Emelichev and Perepelitsa (1992)	+	Paquete and Stützle (2009)
quadratic assignment problem	+	Knowles and Corne (2003a)	−	Paquete and Stützle (2009)
conventional knapsack	+	Bazgan et al. (2009)	+	Paquete et al. (2008)
knapsack with binary weights	✗	Gorski et al. (2012)	✓	Gorski et al. (2012)
unconstrained knapsack	✓	Ehrgott (2005)	+	Liefooghe et al. (2013a)
knapsack with bounded cardinality	+	Liefooghe et al. (2013a)	+	Liefooghe et al. (2013a)
knapsack with fixed cardinality	+	Liefooghe et al. (2013a)	+	Liefooghe et al. (2013a)
unconstrained binary quadratic programming	✓	Liefooghe et al. (2014)	+	Liefooghe et al. (2015b)
long k -path problem	✓	Verel et al. (2011b)	✓	Verel et al. (2011b)
multiple k -path problem	✓	Verel et al. (2011b)	✗	Verel et al. (2011b)

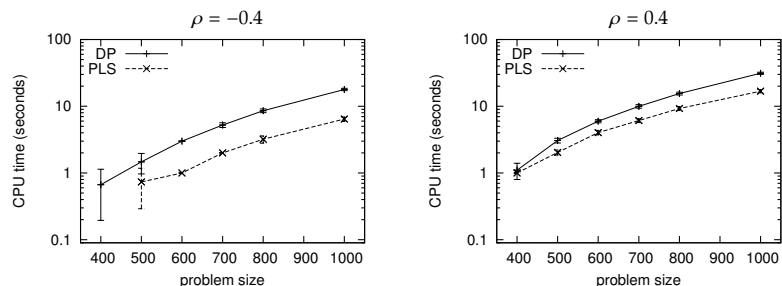
Table 3.10: Proportional size of the largest connected component in the Pareto graph (%larg) for three bi-objective knapsack problems, and proportion of Pareto optimal solutions (%reso) as well as multiplicative epsilon indicator value (eps) obtained by PLS.

knapsack	%larg	%reso	eps
unconstrained	99.9	99.9	1.00
bounded card.	67.5	63.6	1.18
fixed card.	100.0	100.0	1.00

Table 3.10. For the unconstrained knapsack problem, more than 99.9% of Pareto optimal solutions belong to the same graph component for all the instances we investigated. Very few solutions are missing on average, and at most three out of 89 851 in the worst case (with $n = 500$, and $\rho = 0.8$). By starting with one Pareto optimal solution, PLS is able to identify the Pareto set in many cases. For other instances, it leads to the identification of more than 99.9% of the Pareto set. The number of missing non-dominated solutions is negligible compared to the cardinality of the Pareto set. Furthermore, as reported in Figure 3.20, PLS performs very efficiently compared against an exact dynamic programming (DP) approach in terms of computational time. Indeed, the larger the instance size, the larger the gap between both algorithms in terms of CPU time. However, PLS appears to be slightly more efficient for correlated objectives ($\rho = -0.4$). When considering bi-objective knapsack problems with a bounded or a fixed cardinality constraint, our experiments suggest that small-sized neighborhood structures give rise to connected Pareto sets quite frequently, and independently of the size and of the structure of input data. This is particularly true for the knapsack with fixed cardinality, where the exact Pareto set was always found by PLS for all the instances we experimented.

We also analyze the connectedness of bi-objective unconstrained binary quadratic programming problems in Liefooghe et al. (2015b). We show that 50 to 95% of the Pareto set belongs the same connected component. This allows us to propose a state-of-the-art algorithm for this problem class: a number of (approximate) supported non-dominated solutions are first identified by means of a scalarizing approach, and

Figure 3.20: CPU time of dynamic programming (DP) and Pareto local search (PLS) for bi-objective unconstrained knapsack problem instances.



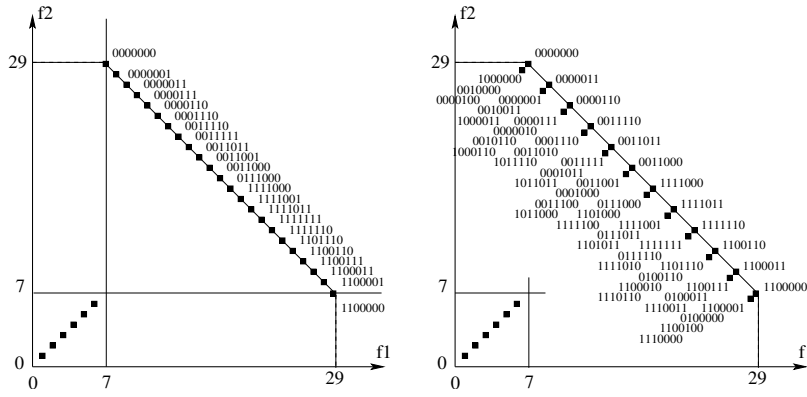


Figure 3.21: Objective space of the bi-objective long 2-path problem (left) and of the bi-objective multiple 2-path problem (right) of dimension $n = 7$. The points on the bottom-left correspond to dominated solutions.

this set is further improved by exploring its neighborhood and by maintaining non-dominated solutions with PLS. This two-phase local search significantly surpasses other considered algorithms over all the instances we experimented.

The performance of Pareto local search (Paquete et al., 2007) is often argued to relate to the connectedness property of the Pareto set. In order to study the joint effect of intractability and connectedness on Pareto local search (PLS), we introduce two illustrative bi-objective binary benchmark problems (Verel et al., 2011b). They extend the single-objective long path problem (Horn et al., 1994), and they are pictured in Figure 3.21. In the bi-objective long path problem (left), the Pareto set is intractable and connected. We theoretically and empirically show that the runtime required by PLS to find an approximation set of a given cardinality is exponential, whereas it is polynomial for G-SEMO (Laumanns et al., 2004a), a dominance-based evolutionary algorithm based on a stochastic bit-flip mutation operator. In the bi-objective multiple path problem (right), the Pareto set is also intractable, but it is not connected anymore. However, a number of shortcuts are incorporated by means of dominated solutions that lead to non-dominated ones. We empirically show that PLS outperforms G-SEMO in this case. The corresponding results are reported in Figure 3.22. As such, we argue that connectedness is not the only structural property that explains the success of Pareto local search, and that additional landscape features are to be taken into account in order to characterize the ability of multi-objective local search algorithms to identify a good Pareto set approximation.

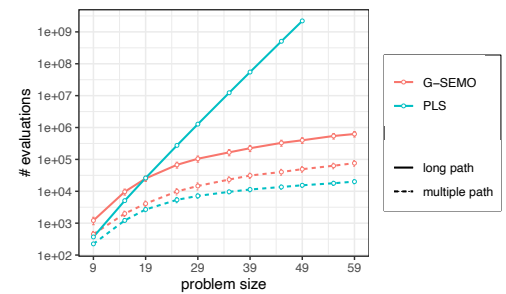


Figure 3.22: Number of evaluations required by PLS and G-SEMO to reach a hypervolume approximation ratio of 0.98 for bi-objective long 2-path and multiple 2-path problems. The approximation set cardinality is bounded by 128.

Visualizing Multi-objective Landscapes with the Pareto Local Optima Network

By going further into the graph representation of solutions in multi-objective landscapes, let us now consider not only Pareto optimal solutions, but all local optima. As detailed in our previous multi-objective landscape analysis, Pareto local optimal solutions are known to highly influence the dynamics and the performance of multi-objective search algorithms, especially those based on local search and Pareto dominance. Our study so far has investigated their impact on the difficulty of searching the landscape underlying a problem instance. However,

the community still lacks knowledge on the structure of Pareto local optimal solutions and the way it impacts the effectiveness of multi-objective search. Inspired by existing works on single-objective local optima networks (Daolio et al., 2012; Ochoa et al., 2008, 2014), we introduce the first multi-objective local optima network model as a step toward the visualization and fundamental understanding of multi-objective landscapes (Liefoghe et al., 2018a).

Given a multi-objective landscape (X, \mathcal{N}, f) , the Pareto local optimal solutions network (PLO-net) is constructed as follows.

Definition 3.6.1 A Pareto local optimal solutions network (PLO-net) is an undirected unweighted simple graph $G = (N, E)$ such that the set of vertices N are Pareto local optimal solutions, and there is an edge $e_{ij} \in E$ between two nodes x^i and x^j iff they are neighbors, i.e. $x^i \in \mathcal{N}(x^j)$ or $x^j \in \mathcal{N}(x^i)$.

Two solutions connected by an edge in the PLO-net are necessarily mutually non-dominated. Moreover, Pareto (global) optimal solutions are particular nodes of the PLO-net.

A visual inspection of the PLO-net is provided in Figure 3.23 for selected problem instances. PLO-nets are constructed by full enumeration of (small) ρmnk -landscapes. By extracting a number of features from network analysis, we are also able to measure the predictive power of PLO-net features with respect to the approximation quality, in terms of Pareto front resolution, of PLS (Paquete et al., 2007) and G-SEMO (Laumanns et al., 2004a) using random forest as a regression model (Breiman, 2001; Liaw and Wiener, 2002). As reported in Table 3.11, the addition of PLO-net features largely improves the prediction accuracy over simply considering ρmnk -landscape benchmark parameters. Some of the proposed PLO-net features appear to have a high impact on search performance. In particular, our results suggest that the number of Pareto local optimal solutions, and more critically their connections in the PLO-net, actually have a larger influence than the number of objectives and their degree of conflict (Liefoghe et al., 2018a).

Table 3.11: Variance explained by the performance prediction regression model for different subset of features.

	$\{\rho, m, n, k\}$	PLO-net feat.	both
PLS	70.31%	83.84%	84.77%
G-SEMO	67.01%	80.07%	81.37%

Automated Algorithm Selection of Anytime Algorithms

Our contributions so far mostly measure algorithm performance as the approximation quality reached for a fixed budget, or as the required runtime to reach a fixed target. However, a trademark of most heuristic search methods is that they identify solutions iteratively, so that they can potentially be interrupted at any time. The relative performance of different algorithms may yield different results depending on the budget or target, and these so-called *anytime algorithms* (Dean and Boddy, 1988; Dubois-Lacoste et al., 2015; Zilberstein, 1996) actually offer different trade-offs among runtime and approximation quality.

In Jesus et al. (2020), we define the performance profile P of an anytime algorithm as the proportion of runs in which a given approximation

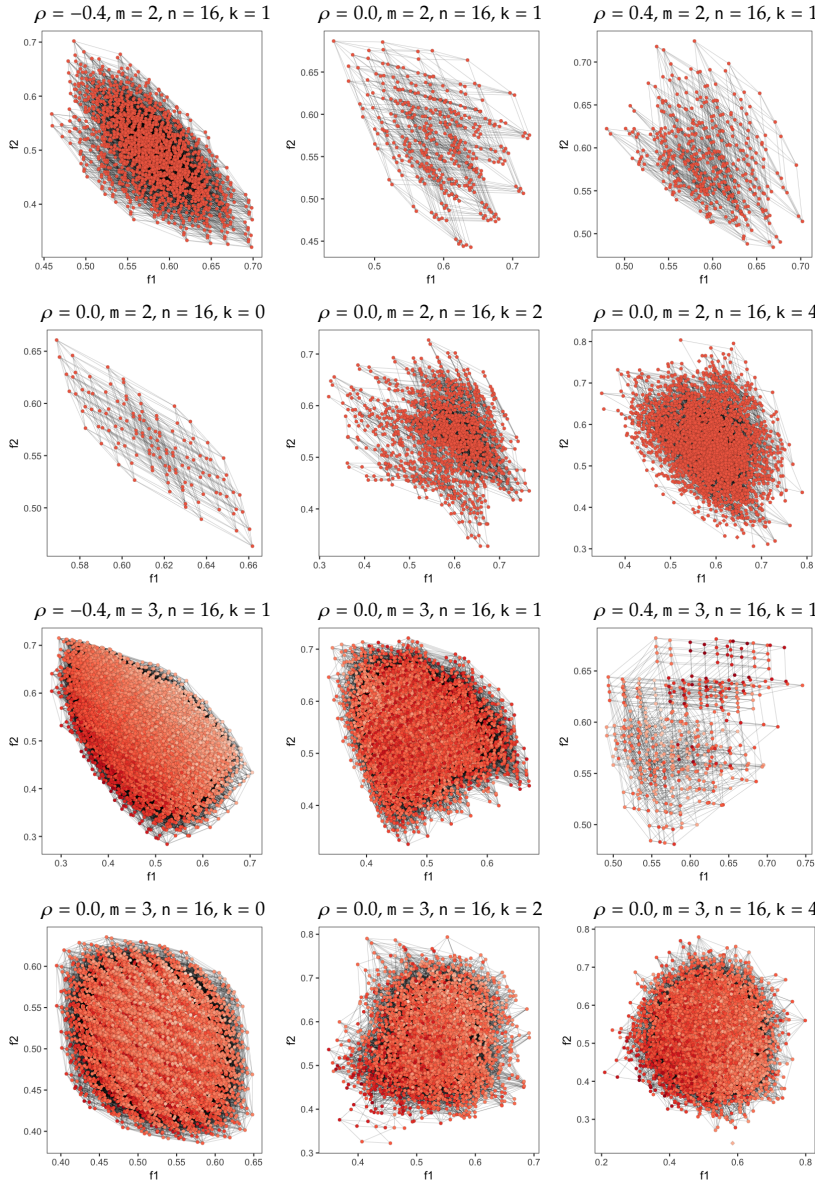


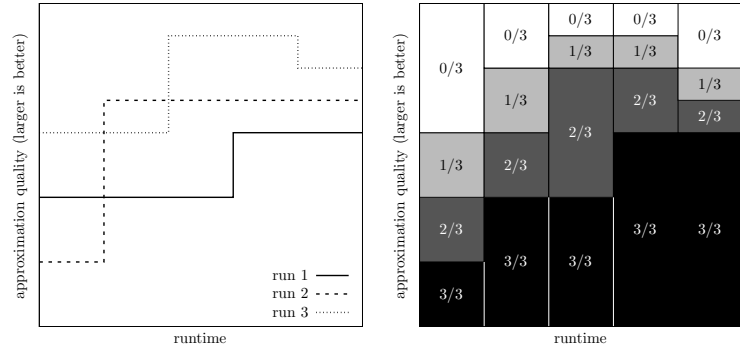
Figure 3.23: Exemplary PLO-nets. For $m = 3$ (bottom), a two-dimensional projection is displayed, the darker the node color, the higher the f_3 -value. Notice the different scales of axes.

quality $q \in Q$ is achieved at time $t \in T$. Figure 3.24 shows the performance traces for three runs of a hypothetical anytime algorithm, and its corresponding performance profile. In order to compare two performance profiles, we introduce a partial order with respect to a utility function w that denotes the anytime preferences of a decision maker.

Definition 3.6.2 Given two performance profiles P and P' , and a utility function $w: T \times Q \mapsto \mathbb{R}_0^+$, we define the relation $P \leq_w P'$ iff $P(t, q) \leq P'(t, q)$ for all $t \in T, q \in Q$ such that $w(t, q) > 0$.

We then introduce a number of scalar performance measures to differentiate between incomparable performance profiles, assuming that the utility function w is restricted to a bounded region of time and quality. These measures are order-preserving with respect to \leq_w and can return distinct values when applied to incomparable profiles. Based on this empirical model, we develop an automated selection methodology

Figure 3.24: Performance traces of three runs (left) and corresponding performance profile (right).



for anytime algorithms. It assumes that the preferences of the decision maker with respect to when an algorithm is going to be interrupted is given in terms of a utility function w . By gathering the performance traces of anytime algorithms on a set of training instances, and by mapping the features of a test instance to similar training instances, we are able to build an approximate performance profile for each algorithm. This allows the automated algorithm selection mechanism to make a decision on which algorithm to select for different budgets and targets.

We conduct experiments on the selection between an exact dynamic programming (DP) algorithm (Figueira et al., 2013) and a Pareto local search (PLS) approach (Paquete et al., 2007) for bi-objective binary knapsack problems with four instance types, as described by Bazgan et al. (2009). We consider fixed bounds for quality and varying upper

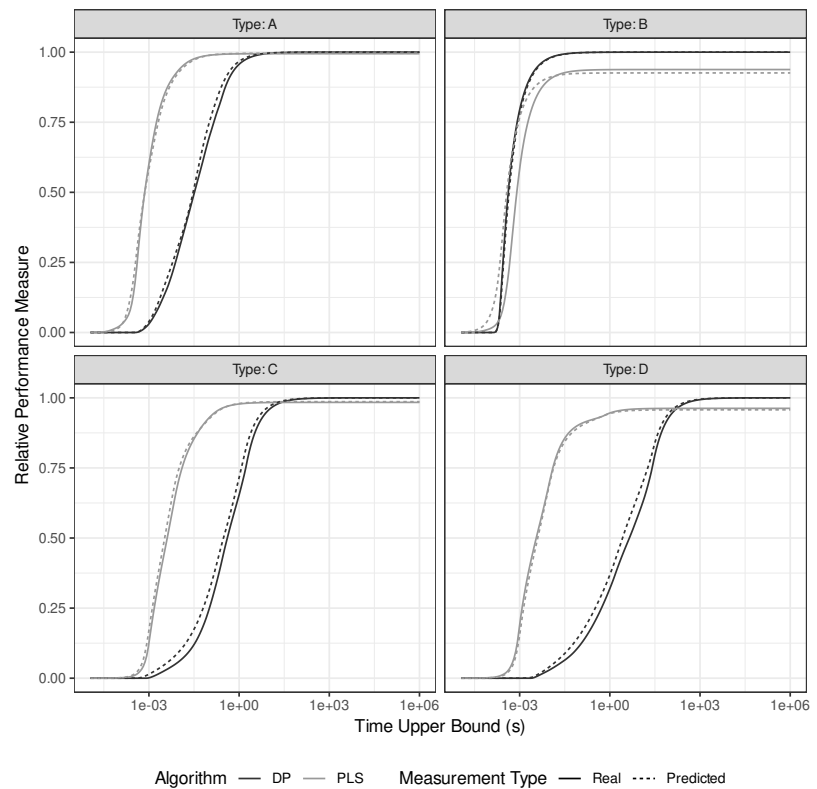


Figure 3.25: Relative performance measure for the “real” performance profiles (continuous lines) and for the performance profile predicted by our selection methodology (discontinuous lines) for knapsack problems with different instance types.

bounds on time in order to study the methodology over varying anytime preferences. Figure 3.25 shows the performance profile predicted by our proposed methodology against the actual performance profile of the considered algorithms for a given instance. This suggests that the predicted performance profile is often close to the real performance profile for both algorithms. In Table 3.12, we report the proportion of selection scenarios with the right selection; a greater value is better. We report the results for (i) our proposed methodology, (ii) a methodology that selects at random between both algorithms, (iii) a methodology that always selects the DP algorithm, and (iv) a methodology that always selects the PLS algorithm. The results indicate that the proposed selection methodology for anytime algorithms is significantly better than the other considered methodologies, achieving an accuracy greater than 96% in the considered scenarios, independently of the instance type.

Landscape-aware Automated Algorithm Configuration

In addition to automated algorithm selection, we are interested in automated algorithm *configuration*. Given a number of algorithm parameters, including categorical, ordinal and numerical parameters, offline automated algorithm configuration methods seek a good configuration, that is a particular choice of the parameter values that best suits the solving of some a priori unknown problem instances (Birattari, 2009; Hamadi et al., 2012; López-Ibáñez et al., 2016). The motivation is not only to get rid from the burden and bias of manual calibration, but more importantly to set up a principled approach for algorithm design, by systematically exploring the strengths and weaknesses of existing configurations when solving a whole family of optimization problems. A number of automated algorithm configuration or parameter tuning approaches have been proposed, ranging from experimental design (Adenso-Diaz and Laguna, 2006), to statistics (Bartz-Beielstein, 2006), heuristic search (Hutter et al., 2009), and racing (Birattari, 2009; Birattari et al., 2002; López-Ibáñez et al., 2016). As illustrated in Figure 3.26, based on a set of training instances, a good configuration, that is expected to generalize well on unseen instances, is identified. As such, the properties of the training set is a key issue in order to obtain a high accuracy of the selected configuration. For example, the heterogeneity of training instances was recently found to be a challenging issue when using racing in the context of SAT (López-Ibáñez et al., 2016). In fact, we argue that a single algorithm configuration might not be suitable to best suit a whole set of heterogeneous instances exhibiting different properties. Following Hutter et al. (2011), Kadioglu et al. (2010), and Xu et al. (2010), we advocate for the discovery of multiple configurations that can then be mapped accurately with respect to the features characterizing a problem instance (Liefvooghe et al., 2017b). However, in contrast with previous works, we are concerned with *black-box* optimization problems. We then rely on low-cost landscape features to be automatically extracted and injected into the algorithm configuration process.

Table 3.12: Prediction accuracy of anytime selection scenarios for for knapsack problems with different instance types.

	A	B	C	D
Proposed	.969	.969	.971	.989
Random	.498	.501	.502	.503
DP	.479	.814	.453	.378
PLS	.521	.186	.547	.622

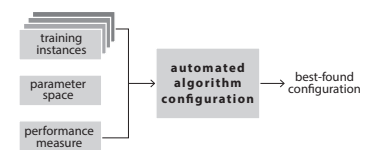
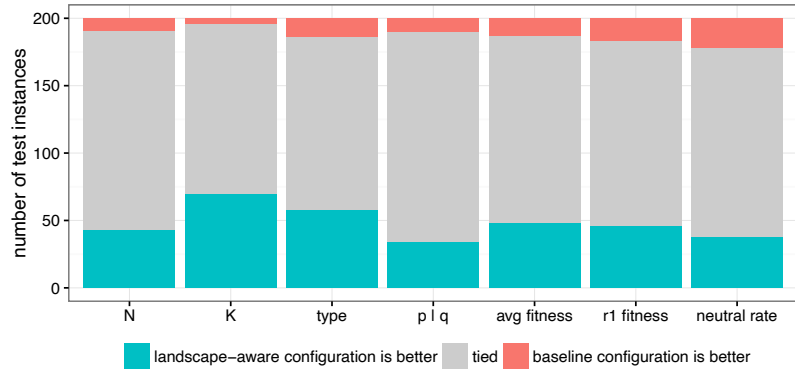


Figure 3.26: Principles of offline automated algorithm configuration.

Figure 3.27: Number of test instances where the landscape-aware configuration with respect to each feature is significantly better, tied or worse than the baseline configuration.



By partitioning the set of training instances into different groups based on the value of landscape features, we conduct an independent training phase in parallel for each group, thus ending up with multiple algorithm configurations, one per group. During the test phase, the appropriate configuration is selected based on the feature value of the considered instance. As a byproduct, we are able to derive a novel landscape-aware methodology that complement existing automated algorithm configuration techniques. By fairly taking the extra computational cost induced by our methodology into account, we investigate the gain of deciding which parameter configuration to choose for an unseen instance based on general-purpose low-cost landscape features. We conducted preliminary experiments on (single-objective) nk-landscapes by considering heterogeneous instances in terms of variable interaction and neutrality, and a conventional memetic algorithm with tunable population size, variation operators, crossover and mutation rates. As reported in Figure 3.27, our empirical findings reveal that a landscape-aware iterated racing method considering one feature at a time is able to identify significantly better configurations than standard iterated racing (López-Ibáñez et al., 2016). Our work can actually be viewed as a first step towards the setting up of more powerful and finely tuned landscape-aware approach. We expect that such a simple high-level algorithm portfolio builder would serve as a basic template for the design of a more advanced and principled approach for solving heterogeneous optimization problems.

Compartmental Models to Capture the Dynamics of Multi-objective Populations

In Monzón et al. (2017, 2020), we track the dynamics of the population handled by evolutionary multi-objective optimization algorithms, i.e. the set of solutions with fixed cardinality that is maintained and evolved by the search process. More particularly, the population's dynamics is captured by means of a compartmental model (Godfrey, 1983). A compartmental model stratifies the population into non-overlapping compartments according to the status of each individual. In the case of multi-objective optimization, we use Pareto dominance to separate solutions into groups. By learning the changes in these groups throughout the iterations, the model captures the evolution

name	abbrv	formula
non-dominated solutions	ND	$\{x \mid x \in ND_t\}$
new non-dominated solutions	NDnew	$\{x \mid x \in ND_t \wedge x \notin \cup_{i=1}^{t-1} ND_i\}$
old non-dominated solutions	NDold	$\{x \mid x \in ND_t \wedge x \in \cup_{i=1}^{t-1} ND_i\}$
dominated solutions	DOM	$\{x \mid x \in P \wedge x \notin ND_t\}$
Pareto optimal solutions	PO	$\{x \mid x \in ND_t \wedge x \in PO\}$
non-dominated non-PO solutions	NDNP	$\{x \mid x \in ND_t \wedge x \notin PO\}$

Table 3.13: Generational search-assessment indices with respect to non-dominated population at iteration t ND_t with respect to ND_{t-1} and/or Pareto optimal (PO) solutions.

of the population, and therefore the dynamics of the considered algorithm. The model parameters offer a compact representation that can further be used for algorithm comparison and to explain their behavior, while the model itself is able to estimate the state of future iterations, provided that the initial conditions are given.

To observe the changes in the population, we introduce a number of generational search assessment indices (Aguirre et al., 2014). Some of them are listed in Table 3.13. Based on these indices, we are able to derive two- and three-compartment models that describe the relationship between solutions from the population. For instance, Figure 3.28 (top) illustrates a two-compartment model that uses the Pareto dominance relation to split the population into dominated (DOM) and non-dominated (ND) solutions. Figure 3.28 (bottom) illustrates a three-compartment model that further splits the population into Pareto optimal solutions (PO), dominated solutions (DOM), and currently non-dominated solutions that are not Pareto optimal (NDNP). Given the values in each compartment at iteration t and the model parameters ($\alpha, \bar{\alpha}, \beta, \bar{\beta}, \gamma, \bar{\gamma}$) for a given algorithm, we can estimate the values of the compartments at future iterations. Model parameters can be estimated for a particular configuration based on a set of training instances. This way, we are able to characterize the population’s dynamics of several evolutionary algorithms for small multi-objective nk-landscapes with different number of objectives (Monzón et al., 2017). Some examples are provided in Figures 3.29–3.30. They provide a valuable information to the algorithm designer, either to verify the working mechanisms of the considered algorithms, or to discover unforeseen behavior triggered by algorithm design choices.

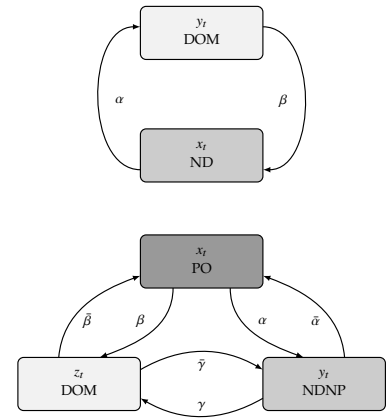


Figure 3.28: An exemplary two-compartment model (top) and three-compartment model (bottom).

We extend our analysis to large-size multi-objective nk-landscapes by considering generational search assessment indices that do not rely on the knowledge of the Pareto set (Monzón et al., 2020). They are based

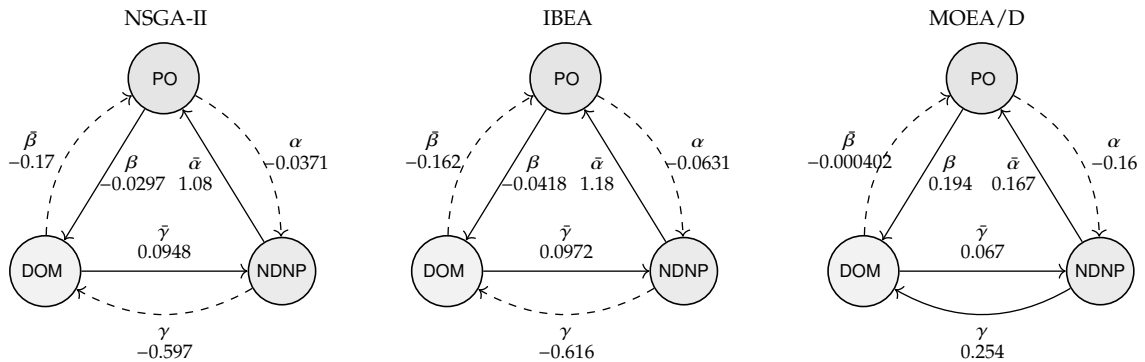


Figure 3.29: Three-compartment model of NSGA-II, IBEA and MOEA/D for a population size of 200 and 3 objectives.

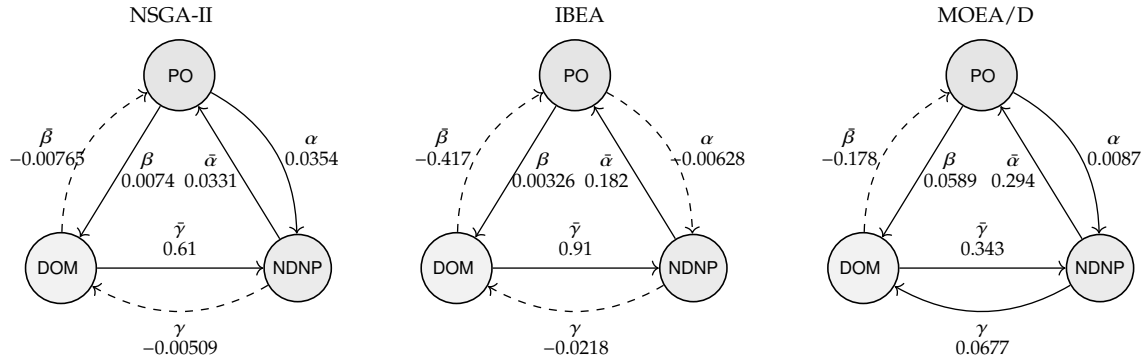


Figure 3.30: Three-compartment model of NSGA-II, IBEA and MOEA/D for a population size of 200 and 5 objectives.

on non-dominated solutions, and when they appear in the population. More particularly, we differentiate between non-dominated solutions that were recently found (ND_{new} in Table 3.13), non-dominated solutions that were found previously (ND_{old}) and dominated solutions (DOM). They provide some information about the discovery of new non-dominated solutions, and then about the rate of progress of the algorithm. This, in turn, was shown to correlate with search performance, in terms of the hypervolume covered by all the solutions found so far by the algorithm, for different population sizes and numbers of objectives (Monzón et al., 2020).

3.7 Outlook and Current Investigations

In this chapter, we investigated the potential of landscape analysis to explain and predict the performance of search heuristics for black-box multi-objective combinatorial optimization problems. We reviewed the state-of-the-art of multi-objective landscape analysis, and we proposed new *general-purpose* features characterizing the landscape, which are affordable for high-dimensional problems thanks to their local nature. By analyzing their association and relevance to search performance, we highlighted the insightful information they are able to capture regarding problem difficulty. In the context of performance prediction, our data-driven analysis revealed the crucial importance of considering *multiple* features in order to reach a good prediction accuracy. In fact, this appears to be the case even when measuring the effect of landscape features *within* instances from the same class; i.e. instances with the same benchmark parameters (Daolio et al., 2015, 2017). From a benchmarking point-of-view, we showed that not only the number of objectives, but also their degree of conflict, are jointly important for search performance. Even more notably, ruggedness and multimodality, which are often overlooked in the multi-objective optimization literature, constitute crucial dimensions that complements the portrait of multi-objective landscapes. By extending results from single-objective landscape analysis, we were able to design affordable walk-based features to characterize the ruggedness and multimodality of multi-objective landscapes. Interestingly, relevant features are not the same for the considered algorithms, which allows us to understand

what makes an instance more difficult to solve for a given algorithm. Ultimately, a number of automated algorithm selection scenarios from an algorithm portfolio allowed us to emphasize that, by leveraging the proposed landscape features, one can automatically select the most appropriate algorithm for unseen (large-size) instances from different problem classes with a high accuracy.

In terms of features and problem understanding, we further explored the impact of connectedness on Pareto local search. Although connectedness certainly does not capture all facets of search difficulty, it allowed us to propose particularly efficient approaches for some problem classes. We also extended the concept of local optima network to multi-objective optimization by means of Pareto local optimal solutions. The proposed PLO-net provides a way to visualize multi-objective landscapes, and features extracted from the network were shown to strongly correlate with search performance. However, we believe that PLO-nets are too large, and we are currently investigating different ways of building *compressed* networks (Ochoa et al., 2017) in order to improve the visualization while maintaining a high level of information. We also devised a number of metrics to characterize the population dynamics of multi-objective heuristics, and we showed how compartmental models could provide a compact representation of the search dynamics for different algorithms, population sizes, and number of objectives. In terms of prediction methodologies, we first extended existing frameworks for automatically selecting from a set of *anytime* algorithms, assuming that the runtime-vs-approximation quality preferences are only known at the time of algorithm selection. We further extended the automated algorithm configuration framework for constructing an *ensemble* of configurations, in order to deal with heterogeneous problem instances. Overall, we contributed to defining a new standard for benchmarking multi-objective optimization approaches. Following a recent survey from Bartz-Beielstein et al. (2020), our contributions are concerned with the following goals of benchmarking activities in optimization: (i) the visualization and basic assessment of algorithms and problems, especially by illustrating the search behavior; (ii) the sensitivity of performance in algorithm design and problem characteristics, most notably by characterizing algorithm performance by problem features and vice versa; and (iii) performance extrapolation, whether being on performance regression or on automated algorithm design, selection, and configuration.

We are currently working on making the proposed multi-objective landscape features publicly available, together with data from benchmarking analyses, by extending our previous work on single-objective landscape analysis in Paradiseo-MO (Humeau et al., 2013). We acknowledge that the features considered in this chapter certainly do not reflect the full portrait of multi-objective problems and algorithms. For instance, they mostly deal with dominance- and indicator-based search, and a recent study reveals that new features based on decomposition can complement existing ones quite well (Cosson et al., 2021). We also recently extended the landscape features to the *continuous* domain, on the basis of a fixed-size sampling of the search space, thus allowing for a fine control over cost when aiming for an efficient application of

automated performance prediction and algorithm selection (Liefoghe et al., 2021). Our experiments reveal that the landscape features deliver a similar prediction accuracy than benchmark parameters for multi-objective interpolated continuous optimization problems. Whether in the continuous or in the combinatorial domain, the underlying feature parameters in terms of sampling and sample size also need to be carefully considered in order to reach the best prediction accuracy (Cosson et al., 2022; Liefoghe et al., 2021). One research line is to allow the algorithms to benefit from the solutions sampled during the computation of features, rather than starting their search process from scratch. At the moment, we are also investigating how features can assist the construction of the set of training instances, and the allocation of the budget in terms of running algorithms on training instances.

In terms of general problem characteristics, whereas this chapter concentrates on problems with up to five objectives, in a late study we closely investigated the influence of the number of objectives — up to twenty — on many-objective landscapes, and the practical behavior of common procedures and algorithms for coping with *many objectives* (Allmendinger et al., 2021). We then make use of our theoretical and empirical findings to derive practical recommendations to support algorithm design. Although the influence of the number of objectives is a preponderant issue in multi-objective optimization, this chapter highlights that it should not be overshadowed by the multimodality of multi-objective combinatorial landscapes. On a recent survey, Grimme et al. (2021) make a similar observation for continuous multi-objective optimization, but also point out that the perspective is inherently different from the combinatorial domain. Integrating ideas and concepts from both continuous and combinatorial domains is surely a challenging but important task for the future of multi-objective landscapes and search. In addition, this chapter considers problems whose objectives are homogeneous. By contrast, problems with *heterogeneous* objectives are those where the objectives differ in one or several aspects, such as scaling, landscape, evaluation time, or theoretical and practical difficulty (Eichfelder et al., 2015). Previous research focused on problems where the heterogeneity arises in evaluation times or latencies, that is, when each objective takes a different amount of time to be evaluated (Allmendinger et al., 2015; Allmendinger and Knowles, 2021). In Santana et al. (2022), we investigate problems with heterogeneous objectives in terms of multimodality. We show that a solution network and features extracted from heterogeneous objectives thoroughly characterize multi-objective landscapes with a different level of heterogeneity among the objectives. We are currently analyzing how heterogeneous objectives influence the difficulty of scalarizing sub-problems, and how to cope with them in decomposition-based multi-objective search. Additionally, problems under constraints, which are not considered in this manuscript, have recently been studied by means of landscape analysis. A current topic is to use landscape features in order to guide constraint handling in heuristic search (Malan and Moser, 2019). In fact, two recent independent studies employ and adapt our landscape features in order to characterize constrained multi-objective optimization problems (Alsouly et al., 2022; Vodopija et al., 2021).

Extending and analyzing our feature-based automated performance prediction and algorithm selection methodology by considering more practical scenarios would allow us to increase our understanding of the landscape structure exhibited by black-box multi-objective optimization problems. Consequently, we plan to consider other multi-objective problem and algorithm classes, to study how algorithm component choices impact search performance over different multi-objective landscapes, and how this could help improving the design and configuration of multi-objective search algorithms. More generally, we believe that landscape features are valuable tools for automated algorithm selection, but also for building the portfolio of algorithms to choose from through automated algorithm configuration. Not only this, we argue they also have a great potential to provide the algorithm designer with insights into the relationships between problems and algorithms, and also to assist them at identifying areas of the problem space in which efficient algorithms are still lacking, and where algorithm design requires improvement. In the next chapter, we pursue our investigations on the foundations of a more recent paradigm for multi-objective optimization, namely set-based multi-objective search.

Set- and Indicator-based Multi-objective Search

4

A large spectrum of quality indicators have been proposed so far to assess the performance of Pareto set approximations. Such indicators assign a real value to a given solution set that reflects an aspect of its approximation quality. This is an important issue in multi-objective optimization, not only to compare the performance of different algorithms, but also to improve their internal selection mechanisms. This chapter first reviews a selection of state-of-the-art indicators, and statistically analyzes their degree of agreement — or disagreement — one another. Afterwards, we consider multi-objective optimization as a set problem, in which the search space consists of all sets of solutions. By leveraging a neighborhood and a preference relation among sets, we define set-based multi-objective landscapes. We further formalize a set-based local search paradigm by combining set-based multi-objective optimization with neighborhood search. As such, a set-based local search seeks a whole set of solutions that jointly optimizes a set preference relation, such as set dominance, hypervolume or epsilon indicator. This allows us to adapt and enhance the comprehensive design of set-based multi-objective search. Indeed, local search has shown good performance for several multi-objective combinatorial optimization problems. However, it naturally stops in a local optimal set, and the question remains open as to how local optimal sets under different set preference relations relate to each other. We thus report an in-depth experimental analysis on local optimal sets under different settings. Our results reveal important implications for multi-objective local search. This chapter gives a particular focus on the works presented in [Liefvooghe and Derbel \(2016\)](#) and [Liefvooghe et al. \(2018b\)](#), the latter being based on preliminary publications ([Basseur et al., 2013](#); [López-Ibáñez et al., 2014](#); [Verel et al., 2011a](#)). We then briefly present related contributions to this research topic and discuss further research.

4.1 Motivations	49
4.2 Quality Indicators	51
4.3 Local Optimal Sets	58
4.4 Further Contributions	66
4.5 Outlook	72

4.1 Motivations

A number of *set quality indicators* have been proposed since the late 1990s, and are still refined today for comparing the output of multi-objective optimization approaches. By defining a total order on Pareto set approximations, they are particularly relevant when the partial order induced by the Pareto dominance relation is not sufficient to discriminate among approximations sets. However, given their different background, structural properties and focus in terms of quality, it is with no surprise that the order obtained with respect to different indicators are sometimes contradictory. For instance, it might be the case that an approximation set obtained by an Algorithm A is pictured to be better than the one from Algorithm B with respect to some indicator, while the opposite is true for another indicator; see, e.g., [Knowles and Corne \(2002\)](#). In addition, quality indicators can be seen as a support for multi-criteria decision making, in the sense that they can be

used to provide the decision maker with a representative subset of a potentially very large set of trade-offs, and thus for presenting a compact and reliable representation of the Pareto set. In that respect, any indicator actually makes some assumptions about the decision maker preferences (Zitzler et al., 2008).

The properties of state-of-the-art quality indicators have been studied in terms of computational complexity, parameter dependency, scaling invariance, and monotonicity with respect to set dominance relations (Knowles et al., 2006; Talbi, 2009; Zitzler et al., 2008, 2003). For instance, the proportion of results from indicators that are not compliant with the Pareto dominance relation is experimented by Knowles et al. (2006). However, the relation between any two indicators is far from being well understood, and it remains unclear what differences in quality or interpretation each indicator is able to provide. Intuitively, this also depends on many factors, such as the shape of the Pareto front, the distribution of solutions in the objective space, or even the underlying indicator parameters. For instance, the hypervolume is known to be affected by the choice of its reference point (Auger et al., 2012; Knowles and Corne, 2003b), to favor convex regions over concave regions (Zitzler and Thiele, 1998), and to place more emphasis on knee points (Auger et al., 2012; Beume et al., 2007). Similarly, the optimal distribution of solutions for the epsilon indicator depends on the shape of the Pareto front (Bringmann et al., 2015). For all these reasons, it might be interesting to measure the agreements and disagreements that these quality indicators have when assessing one approximation set better than another, depending on the problem characteristics, and given different levels of approximation quality. In this chapter, we adopt a statistical analysis in order to experimentally investigate to what extent indicators agree with each other on the induced ranking of approximation sets (Liefvooghe and Derbel, 2016). This analysis gives a first step towards a better understanding of the connection between quality indicators, and might provide important implications for performance assessment, algorithm design and decision making.

More recently, those quality indicators have been plugged onto the design principles of multi-objective search algorithms. Such indicator-based approaches seek an approximation set that optimizes the indicator value; see, e.g., Bader and Zitzler (2011), Beume et al. (2007), Brockhoff et al. (2015), Knowles (2002), Zitzler et al. (2007), and Zitzler and Künzli (2004). As such, the goal of the search process can be explicitly formalized as a *set problem* (Zitzler et al., 2010): the search space is made of *sets* of solutions, and not single solutions. Unlike the previous chapter, where the definition of multi-objective landscapes was based at the solution level — being solutions from the Pareto set or from the whole solution set — we here propose to define a *set-based multi-objective landscape* in order to deal with the search space properties at the set level (Verel et al., 2011a). This definition is based on a search space made of feasible sets, a neighborhood relation among sets, and a preference relation among sets, such as a set dominance relation or a quality indicator. Following this view, we further extend the concept of set-based multi-objective optimization with the aim of formalizing a *set-based local search* (Basseur et al., 2013). The purpose is

not to introduce a novel algorithm, since existing approaches already share similar principles (Beume et al., 2007; Knowles, 2002; Laumanns et al., 2004b; Paquete et al., 2007). Instead, this paradigm synthesizes, abstracts and extends a large class of multi-objective approaches by formulating or re-formulating them in terms of local search algorithms. Moreover, it provides a unified framework for the design, analysis and comparison of different approaches from a common terminology and classification.

As in the single-objective case, the components from the landscape play an important role on the design of local search. The behavior and performance of the search process and the landscape characteristics are, to a large extent, induced by the definition of these components. However, although the concept of local optimality is well-studied in single-objective optimization, its extension and properties in multi-objective optimization are much less understood. The implications of these different aspects are still open to investigation and may guide the design of new multi-objective algorithms. In fact, even global search methods, such as multi-objective evolutionary algorithms, can be seen as iteratively identifying better-quality local optima, without any guarantee of finding a *global optimum*. However, so far, no work has examined how various definitions of local optimality relate to each other. In this chapter, we extend previous work on local optimal sets by considering various types of local optima induced by different set preference relations, and by analyzing their properties (Liefvooghe et al., 2018b). This allows us to report important findings for multi-objective landscapes with direct implications for multi-objective search.

The chapter is organized as follows. In Section 4.2, we describe the quality indicators under consideration in our study, and we provide a correlation analysis among them on a number of benchmarks. In Section 4.3, we define set-based multi-objective landscapes, together with the concept local optimality in a way that matches the single-objective case but allows the use of set preference relations based on dominance or indicators. In addition, we describe an adaptive walk that follows the set-based local search paradigm for sampling local optimal sets, and we report and discuss experiments with respect to different set preference relations for problems with a varying number of variable interactions and objectives. In Section 4.4, we give a brief summary of complementary contributions on indicator-based multi-objective search. In Section 4.5, we finally summarize our main findings and list remaining open questions.

4.2 Quality Indicators

Definitions

Let us recall that we consider an arbitrary multi-objective optimization problem $f: X \mapsto Z$, such that X is the variable space, and $Z \subseteq \mathbb{R}^m$ is the objective space. A (unary) *quality indicator* is a function $2^X \rightarrow \mathbb{R}$ that assigns each approximation set to a (scalar) value reflecting its

quality (Zitzler et al., 2008). In the following, we introduce a number of conventional quality indicators from the multi-objective optimization literature. The reader is referred to Knowles and Corne (2002), Knowles et al. (2006), and Zitzler et al. (2008, 2003) for broader reviews. Let $A \subseteq X$ be a set of mutually non-dominated solutions; i.e. a Pareto set approximation, or approximation set. Let $R \subseteq Z$ be a reference set, ideally the exact Pareto front when it is discrete. In the following, we assume that any solution in A is weakly dominated by an objective vector in R ; i.e. $\forall a \in A, \exists r \in R$, such that $f(a) \leq r$. In other words, any approximation set A is weakly dominated by the reference set R (Zitzler et al., 2003).

Inverted generational distance (igd). The inverted generational distance (Coello Coello and Reyes-Sierra, 2004) is an inverted version of the generational distance (Van Veldhuizen and Lamont, 1998); see also Schütze et al. (2012) for a detailed explanation. It gives the average distance between any point from the reference set R and its closest point from the approximation set A :

$$\text{igd}(A) := \frac{1}{|R|} \sqrt{\sum_{r \in R} \min_{a \in A} \|f(a) - r\|_2^2}$$

The euclidean distance (L2-norm) in the objective space is typically used for distance calculation. Obviously, the smaller the igd value, the closer the approximation set from the reference set. An indicator value of $\text{igd}(A) = 0$ actually implies $R \subseteq f(A)$.

Epsilon indicators (eps). The epsilon indicator family (Zitzler et al., 2003) gives the minimum factor by which the approximation set has to be translated in the objective space in order to (weakly) dominate the reference set. The *additive* epsilon indicator (eps^+) is based on an additive factor:

$$\text{eps}^+(A) := \max_{r \in R} \min_{a \in A} \max_{i \in \{1, \dots, m\}} (r_i - f_i(a))$$

The *multiplicative* version (eps^\times) is based on a multiplicative factor, and assumes that all objective values are strictly positives:

$$\text{eps}^\times(A) := \max_{r \in R} \min_{a \in A} \max_{i \in \{1, \dots, m\}} (r_i / f_i(a))$$

Both epsilon indicator versions are to be minimized, and $\text{eps}^+(A) = 0$ or $\text{eps}^\times(A) = 1$ implies that $R \subseteq f(A)$.

R-metrics (r). The family of R-metrics (Hansen and Jaszkiewicz, 1998) are based on a set of scalarizing functions. A scalarizing function $g : X \rightarrow \mathbb{R}$ maps a solution to a scalar value based on specified parameters. A typical example is the (weighted) Chebyshev scalarizing function:

$$g(x | \lambda) := \max_{i \in \{1, \dots, m\}} \lambda_i \cdot |z_i^* - f_i(x)|$$

where $x \in X$ is a solution, $z^* \in \mathbb{R}^m$ is the ideal point such that $z_i^* = \min_{x \in X} f_i(x)$, $i \in \{1, \dots, m\}$, and $\lambda \in \mathbb{R}^m$ is a weighting coefficient vector. By defining a set of uniformly-defined weighting coefficient vectors Λ such that for all $\lambda \in \Lambda$, $\lambda = (\lambda_1, \dots, \lambda_i, \dots, \lambda_m)$, $\lambda_i \geq 0$ and $\sum_{i=1}^m \lambda_i = 1$, the r2 and r3 indicators can be defined as follows:

$$r2(A) := \frac{1}{|\Lambda|} \sum_{\lambda \in \Lambda} \left(\min_{r \in R} g(r | \lambda) - \min_{a \in A} g(a | \lambda) \right)$$

$$r3(A) := \frac{1}{|\Lambda|} \sum_{\lambda \in \Lambda} \frac{\min_{r \in R} g(r | \lambda) - \min_{a \in A} g(a | \lambda)}{\min_{r \in R} g(r | \lambda)}$$

Once again, both r2 and r3 indicators are to be minimized, and $r2(A) = 0$ or $r3(A) = 0$ implies $R \subseteq f(A)$.

Relative hypervolume (rhv). The hypervolume indicator (Zitzler and Thiele, 1998; Zitzler et al., 2003) gives the multidimensional volume of the portion of the objective space that is weakly dominated by an approximation set:

$$hv(A) := \int_{z^{min}}^{z^{max}} \alpha_A(z) dz$$

such that:

$$\alpha_A(z) := \begin{cases} 1 & \text{if } \exists a \in A \text{ such that } z \leq f(a) \\ 0 & \text{otherwise} \end{cases}$$

In practice, only the lower-bound vector $z^{min} \in \mathbb{R}^m$ is required to compute the hypervolume; this parameter is the *hypervolume reference point*. In the following, we are interested in the *relative hypervolume indicator* (rhv), that is the relative deviation of the approximation set's hypervolume to the reference set's hypervolume:

$$rhv(A) := \frac{hv(R) - hv(A)}{hv(R)}$$

This allows us to consider minimizing indicator values as well, such that $rhv(A) = 0$ means that $R \subseteq f(A)$. The hypervolume and relative hypervolume deviation are illustrated in Figure 4.1.

Properties

We summarize below some properties of quality indicators in terms of monotonicity, scaling invariance, parameter setting, and computational complexity (Knowles et al., 2006; Zitzler et al., 2008, 2003).

Monotonicity. Given two approximation sets $A, B \in 2^X$, we recall that A is *weakly dominated* by B ($A \preceq_{\text{dom}} B$), if for all $b \in B$ there exists an $a \in A$ such that $a \leq b$. A quality indicator is order-preserving or monotonic⁴ if it does not disagree with the (partial) order induced by the dominance relation.

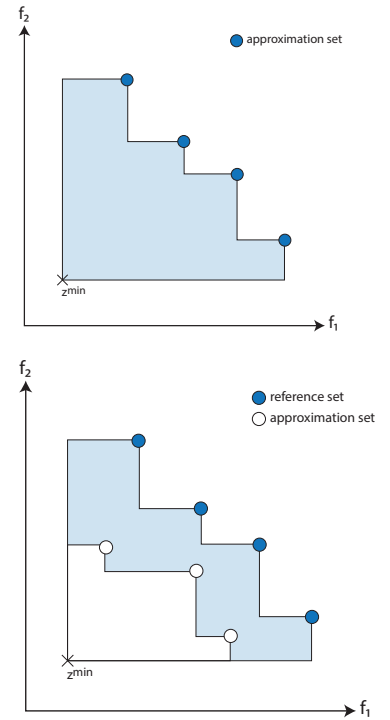


Figure 4.1: hypervolume (hv, top) and hypervolume relative deviation (rhv, bottom).

4: Pareto-compliant in Knowles et al. (2006) and Zitzler et al. (2003).

Definition 4.2.1 An indicator $I: 2^X \rightarrow \mathbb{R}$ is order-preserving with respect to the weak-dominance set preference relation iff:

$$A \preceq_{dom} B \implies I(B) \leq I(A) \quad (4.1)$$

Definition 4.2.2 An indicator $I: 2^X \rightarrow \mathbb{R}$ is strictly order-preserving with respect to the weak-dominance set preference relation iff:

$$(A \preceq_{dom} B) \wedge \neg(B \preceq_{dom} A) \implies I(B) < I(A) \quad (4.2)$$

All the indicators presented above are order-preserving, with the exception of *igd*, despite being regularly used for performance assessment. However, the hypervolume is the only known indicator that is *strictly* order-preserving (Zitzler et al., 2003); i.e., any approximation A that is dominated by an approximation B always have a strictly lower hypervolume value. Notice that an empirical analysis of the degree of monotonicity for some non-order-preserving indicators are reported by Ishibuchi et al. (2015) and Knowles et al. (2006).

Scaling invariance. An indicator is scaling invariant if the order among approximation sets induced by the indicator values remains the same when applying a monotonic transformation of the objective function values. However, as the indicators under consideration all explicitly make use of the objective function values, none of them actually satisfies this scaling invariance property, except when the hypervolume reference point is transformed accordingly.

Parameters. In our definitions of quality indicators, a reference set R is always required, although the hypervolume (*hv*) could be used without any reference set. In addition, the definition of *r2* and *r3* is based on the ideal point and on a user-given number of weighting coefficient vectors, while the definition of *hv* and *rhv* is based on a reference point. In fact, the ordering of the approximation sets induced by the hypervolume is known to be affected by the setting of this reference point (Auger et al., 2012; Ishibuchi et al., 2017; Zitzler et al., 2008).

Computational complexity. Since an in-depth experimental analysis may require the comparison of a large number of approximation sets, and given that an indicator can potentially be integrated into the search process of multi-objective algorithms, the computational resources required to compute an indicator value is an important feature of its characteristics. The computational complexity for *igd*, *eps* and the *r* metrics is polynomial in the number of objectives, the approximation set cardinality and the reference set cardinality (as well as the number of weighting coefficient vectors for *r2* and *r3*), whereas it is exponential in the number of objectives for the hypervolume (Zitzler et al., 2008). Although efficient algorithms exist for $m = 3$ (Beume et al., 2009) and $m = 4$ (Guerreiro and Fonseca, 2018), the best-known algorithm for

computing the hypervolume in the general case has an exponential complexity with respect to m (Chan, 2013).

Correlation Analysis

In Liefoghe and Derbel (2016), we measure the agreements and disagreements among quality indicators when assessing the quality of approximation sets. We adopt a statistical analysis in order to experimentally investigate by how much quality indicators agree with each other on the induced ranking of approximation sets. Our analysis extends results from Jiang et al. (2014) and Wessing and Naujoks (2010) by systematically analyzing the non-parametric rank correlations between quality indicators, and by contrasting their association across a large spectrum of approximation quality and problem classes.

We compute the indicator values of a sample of approximation sets over well-established multi-objective continuous functions from the CEC 2009 special session and competition (Zhang et al., 2009). They have been specifically designed to resemble complicated real-life optimization problems, and they present different properties in terms of dimension, separability, multi-modality, and shapes of the Pareto front such as convexity, concavity, discontinuities, or gaps. We consider the following strategies for sampling a subset of approximation sets, each one being repeated 1 000 times for every problem:

- ▶ **low-Q**: non-dominated solutions from a sample of 100 solutions generated uniformly at random;
- ▶ **med-Q**: the approximation set obtained by NSGA-II (Deb et al., 2002) after 1 000 generations, using default settings and a population of size 100;
- ▶ **high-Q**: a sample of 100 solutions selected at random from the reference set.

In order to measure the association between the indicator values, we consider the Kendall rank correlation coefficient τ (Kendall, 1948), which is a rank-based non-linear correlation coefficient measure. This allows us to focus on the *ranking* of approximation sets obtained within each indicator, i.e. by how much do the indicators rank the approximation sets similarly. In other words, we are not interested in the correlation between the values obtained by each indicator, but rather on the underlying ranking they obtain within the sample of approximation sets. Given two indicators I_1 and I_2 and a pair of approximation sets A and B , the pair is said to be *concordant* if $I_1(A) > I_1(B) \wedge I_2(A) > I_2(B)$, or if $I_1(A) < I_1(B) \wedge I_2(A) < I_2(B)$. On the contrary, the pair is said to be *discordant* if $I_1(A) > I_1(B) \wedge I_2(A) < I_2(B)$, or if $I_1(A) < I_1(B) \wedge I_2(A) > I_2(B)$. If $I_1(A) = I_1(B)$ or $I_2(A) = I_2(B)$, the pair is neither concordant nor discordant. The Kendall coefficient τ quantifies the difference between the proportion of concordant and discordant pairs among all possible pairs:

$$\tau = \frac{(\% \text{ concordant pairs}) - (\% \text{ discordant pairs})}{\% \text{ pairs}}$$

The coefficient τ ranges in $[-1, 1]$, from complete disagreement ($\tau = -1$) to perfect agreement ($\tau = 1$). When τ is approximately zero, the indicator values are uncorrelated.

Figure 4.2 reports the Kendall rank correlation coefficient between all pairs of indicators for each problem and each sampling strategy. We split the correlation values into different groups, from a very high negative correlation ($\tau < -0.75$) to a very high positive correlation ($\tau > 0.75$), as well as an additional group containing test cases which were reported to be non-significant by the Kendall coefficient test, with a p -value of 0.05. The figure provides the correlation between any pair of indicators (on the x - and y -axes) for each problem (from top to bottom) and each sampling strategy (from left to right). The higher the correlation degree, the higher the agreement between the two corresponding indicators, the darker the corresponding area on the heat-map. It is worth noting that we investigate two settings for the hypervolume's reference point: (i) $z_i^{\max} = f^{\max}$, and (ii) $z_i^{\max} = 1.1 \cdot f^{\text{worst}}$ for all $i \in \{1, \dots, m\}$, such that f^{\max} is the maximum objective function value for the problem under consideration, and f^{worst} is the worst objective function value found for a given problem and a given sampling strategy. The corresponding indicator is denoted by $\text{rhv}(\max)$ and $\text{rhv}(\text{worst})$, respectively.

Overall, we observe that the indicators under consideration are never in conflict with each other, as there is never a significant amount of negative correlation. However, there does not exist any two indicators that fully agree with each other on any of the problems; i.e., τ is always strictly lower than 1.00. The few test cases where the τ value is larger than 0.98 actually correspond to indicators from the same family. This confirms that the performance of multi-objective optimizers cannot be assessed universally with a single indicator, and that each indicator actually measures a different facet of approximation quality. Interestingly, the correlation of epsilon indicators with the others is overall very low. This means that eps^+ and eps^\times actually focus on complementary aspects with respect to other indicators, but also that they do not capture all facets of approximation quality. The same reasoning applies for igd . For this reason, it would be worth revisiting the data from the CEC 2009 competition, where igd was the only considered indicator, in order to enhance our knowledge and understandings of the competing algorithms by means of supplementary indicators. By contrast, rhv shows a high correlation with $r2$ and $r3$ for low- and medium-quality approximations. As a consequence, it would be worth investigating more thoroughly the estimation of the computationally-prohibitive hypervolume indicator with $r2$ or $r3$ in order to speed-up the selection process of hypervolume-based approaches, a research line that was investigated by Brockhoff et al. (2015) and Shang et al. (2020). For hypervolume, we also remark that the reference point is particularly impactful for high-quality approximation sets, the agreement between both settings being particularly low in this case.

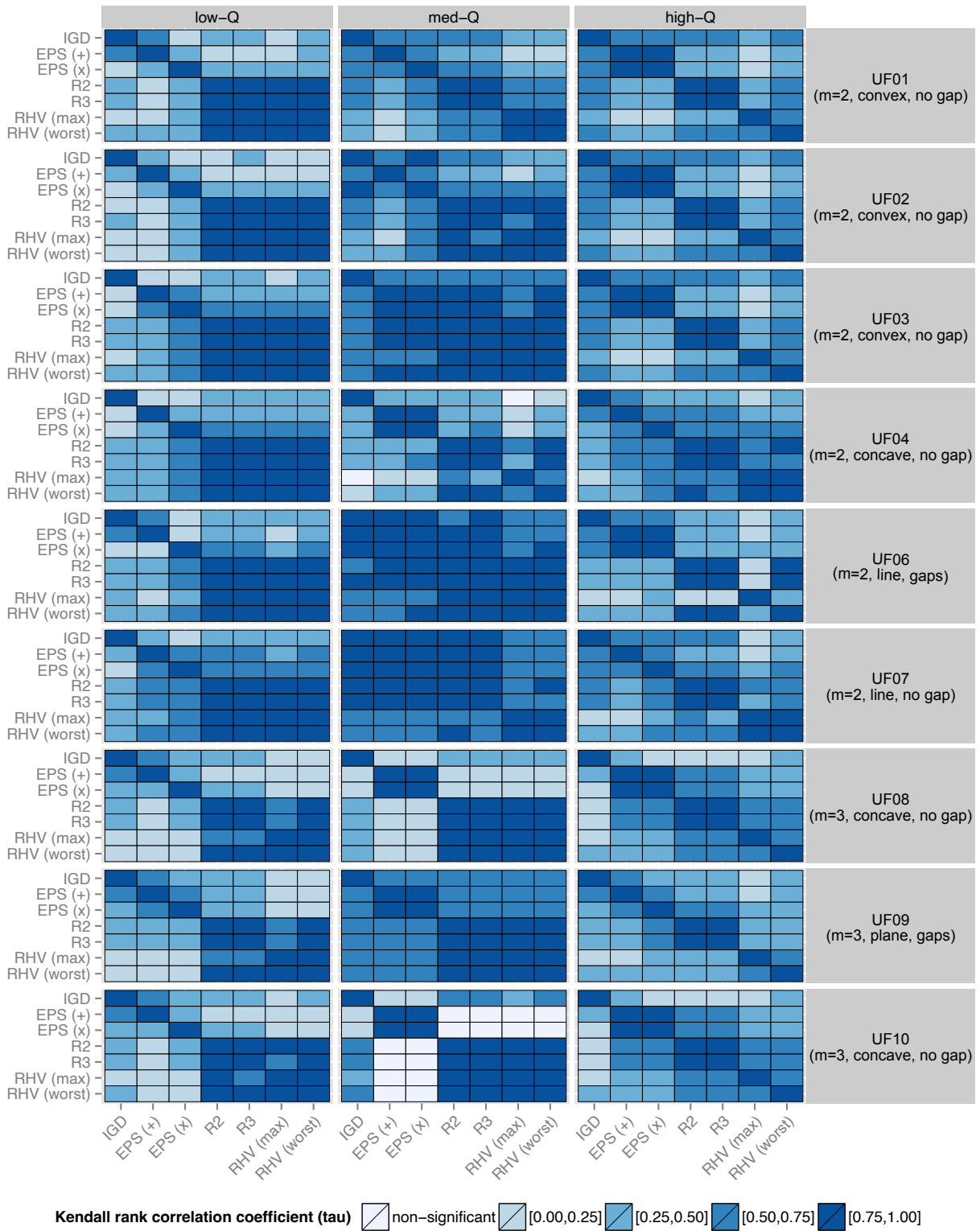


Figure 4.2: Heat-map Kendall rank correlation τ for each pair of set quality indicators (displayed on both axes), each sampling strategy (low-Q, med-Q, high-Q) and each problem function (UF01–UF10).

4.3 Local Optimal Sets

Let us now investigate the difficulty of searching for high-quality approximation sets with respect to dominance and quality indicators, and the properties of local optima for set-based multi-objective search. The concept of local optimality is not well understood in multi-objective optimization. The difficulties arise from the fact that the search space is actually the set of all mutually non-dominated sets of solutions, possibly bounded in size, thus the neighborhood can be seen as operating on sets. Moreover, the preference relation is usually defined in terms of Pareto dominance, but it may also be any other quality indicator that induces a pre-order, such as the hypervolume.

Paquete et al. (2007) provide definitions of local optimality with respect to solution- and set-dominance, i.e. Pareto local optimal solutions (PLO-solutions) considered in Chapter 3 and Pareto local optimal sets (PLO-sets). They also relate them to the convergence point of Pareto local search. We later introduced the notion of *set-based landscape*, and we measured ruggedness and non-linearity for fixed-size sets of solutions, using the hypervolume as the preference relation (Verel et al., 2011a). That study was further extended to the quality of local optimal sets (LO-sets) and the convergence profile of hypervolume-based local search under different notions of set neighborhoods (Basseur et al., 2013). More recently, we showed that the estimated number of PLO-sets is correlated with the number of objectives or their degree of conflict (López-Ibáñez et al., 2014). It is also known that the number of PLO-solutions increases linearly with the problem non-linearity (Verel et al., 2013), but that the number of unbounded PLO-sets decreases (López-Ibáñez et al., 2014). Finally, we observed that bounding the sets in cardinality does not change these trends, but increases the number of bounded PLO-sets significantly by a factor that depends on the cardinality of the unbounded PLO-sets. We examine below how various definitions of local optimality relate to each other by considering various types of local optima, induced by different set preference relations (dominance, epsilon, hypervolume), and by analyzing their properties. This section is based on Liefooghe et al. (2018b).

Definitions

Set-based multi-objective landscape. In set-based multi-objective optimization, the search space is defined as the collection of sets of solutions (Zitzler et al., 2010). In Verel et al. (2011a), we define a *set-based landscape* as follows.

Definition 4.3.1 A set-based multi-objective landscape is defined as a triplet $(\Sigma, \mathbb{N}, \preceq)$ such that:

- ▶ $\Sigma \subset 2^X$ is a collection of feasible sets,
- ▶ $\mathbb{N}: \Sigma \mapsto 2^\Sigma$ is a neighborhood relation between sets,
- ▶ \preceq is a pre-order (preference relation) between sets.

Different examples for the definition of the search space for populations, unbounded and bounded archives are discussed in [Basseur et al. \(2013\)](#) and [Verel et al. \(2011a\)](#). In the following, we restrict the search space to sets of mutually non-dominated solutions and we consider that the cardinality of the sets is bounded by $\mu \in \mathbb{N}^+$, that is:

$$\Sigma := \{A \in 2^X : |A| \leq \mu \wedge \forall x, x' \in A, x \neq x' \implies \neg(x \leq x')\}$$

Let $A, B \in \Sigma$ be two approximation sets. We consider the additive epsilon indicator (eps^+) and the hypervolume (hv), and we define the corresponding set preference relations:

$$A \preceq_{\text{eps}} B \iff \text{eps}^+(A) \leq \text{eps}^+(B)$$

$$A \preceq_{\text{hv}} B \iff \text{hv}(A) \geq \text{hv}(B)$$

We recall that eps^+ (respectively, hv) is order-preserving (respectively, strictly order-preserving) with respect to the weak-dominance set preference relation (\preceq_{dom}). We define the *strict* partial order ($<$) of a given partial order (\preceq) as:

$$A < B \iff (A \preceq B) \wedge \neg(B \preceq A)$$

Sets A and B are *incomparable* if neither $A < B$ nor $B < A$ holds.

Local optimal sets. Given a set-based landscape $(\Sigma, \mathbb{N}, \preceq)$, the definition of local optima can be adapted as follows.

Definition 4.3.2 (Local optimal set, LO-set $(\Sigma, \preceq, \mathbb{N})$) *A set $A \in \Sigma$ is a local optimal set iff $\forall B \in \mathbb{N}(A) \setminus A, \neg(B \preceq A)$.*

Definition 4.3.3 (Strict LO-set, sLO-set $(\Sigma, \preceq, \mathbb{N})$) *A set $A \in \Sigma$ is a strict local optimal set iff $\forall B \in \mathbb{N}(A) \setminus A, A < B$.*

Under the definitions above, a *Pareto local optimum set* (Paquete et al., 2007) is an LO-set where the pre-order is the set-dominance relation \preceq_{dom} . It would be a *strict* LO-set under the same definitions if there is no $B \in \mathbb{N}(A)$ such that A and B are incomparable. As another example, a multi-objective local search based on hypervolume (\preceq_{hv}) stops on an LO-set $A \in \Sigma$ if there exists no neighboring set $B \in \mathbb{N}(A)$ that has a larger hypervolume value. It stops on a strict LO-set if all neighboring sets have a (strictly) smaller hypervolume value than the current set. Therefore, the proposed definitions allow us to compare various types of LO-sets under a common terminology.

An adaptive walk to sample local optimal sets. Following the definitions of strict and non-strict LO-sets, we define a set-based adaptive walk in Algorithm 1, where the first improving neighboring set encountered during neighborhood exploration is accepted. This set-based local search is analogous to a classical single-objective first-improvement local search (or *hill-climber*), known as *adaptive walk* in landscape analysis. In Algorithm 1, μ initial solutions are randomly generated and

Algorithm 1: Set-based adaptive walk

Input : Set cardinality bound $\mu \in \mathbb{N}^+$, neutral $\in \{\text{TRUE}, \text{FALSE}\}$,
 partial order $\preceq \in \{\preceq_{\text{dom}}, \preceq_{\text{eps}}, \preceq_{\text{hv}}\}$

Output: Approximation set A

```

1  $A \leftarrow \emptyset$ 
2 for  $i \leftarrow 1$  to  $\mu$  do
3    $x \leftarrow \text{RandomSolution}()$ 
4    $A \leftarrow \text{FilterDominated}(A \cup \{x\})$ 
5 repeat // main loop
6   for each  $x' \in \{\mathcal{N}(x) \setminus A \mid x \in A\}$  do // random order
7      $A' \leftarrow \text{FilterDominated}(A \cup \{x'\})$ 
8     if  $|A'| \leq \mu$  then
9        $A \leftarrow A'$ 
10      goto line 19
11     for each  $x'' \in A$  do // random order
12        $A' \leftarrow \{A \cup x'\} \setminus \{x''\}$ 
13       if  $A' < A$  then //  $A'$  better than  $A$ 
14          $A \leftarrow A'$ 
15         goto line 19
16       else if neutral  $\wedge A' \preceq A$  then //  $A'$  not worse than  $A$ 
17          $A \leftarrow A'$ 
18         goto line 19
19 until  $A$  is a (s)LO-set or no budget left or cutoff reached

```

added to a non-dominated archive A , which represents the current solution-set. Then, a main loop explores each neighboring solution x' of each element in A in a random order without replacement. If this neighbor x' is non-dominated with respect to any solution in A and the cardinality of A is smaller than μ , then A can be trivially improved by adding solution x' . Otherwise, the algorithm explores all sets that are constructed by replacing one solution from A with x' . If the resulting set improves over A , it is accepted. In the case of a *neutral* walk, the solution is also accepted if the resulting set is incomparable with A . Otherwise, the procedure explores the next neighboring set. The main loop stops once all neighbors have been explored, returning a (possibly strict) LO-set, when a budget of solution evaluations has been consumed, or when there is a number of steps without any strict improvement. A step is here defined as a change in the current set, i.e. an iteration of the main loop (lines 5–19).

The proposed adaptive walk shares similarities with existing multi-objective local search methods. Compared against PLS (Paquete et al., 2007) and SEMO (Laumanns et al., 2004a), we consider mutually non-dominated sets of *bounded* cardinality. The neighbors of a set are the same as those in SEMO (Laumanns et al., 2004a); i.e., given $A \in \Sigma$:

$$B \in \mathcal{N}(A) \iff |B \setminus A| \leq 1 \wedge \forall b \in B \setminus A, \exists a \in A \text{ such that } b \in \mathcal{N}(a)$$

The main difference is that we explore neighboring sets *without* replacement, which allow us to detect when the walk falls into a (possibly strict) LO-set. Other set neighborhood relations are defined and in-

vestigated in Basseur et al. (2013). As in single-objective local search, the proposed non-neutral adaptive walk always falls into an LO-set, whereas a neutral walk may either eventually fall into a strict LO-set, or terminate without reaching any type of LO-set. By using this adaptive walk, we can experimentally estimate the number, quality, and dissimilarity of various types of LO-sets, as shown below.

Experimental Analysis

Experimental setup. We consider 15 ρmnk -landscapes with the following settings: a problem size $n = 16$, a number of variable interactions (epistatic degree) $k \in \{0, 1, 2, 4, 8\}$, a number of objectives $m \in \{2, 3, 5\}$, and no objective correlation $\rho = 0$. We generate one instance at random for each combination of instance settings. We run the adaptive walk (Algorithm 1) with respect to the set preference relations $\{\preceq_{\text{dom}}, \preceq_{\text{hv}}, \preceq_{\text{eps}}\}$ and with various set cardinality bounds $\mu \in \{2, 4, 8, 16, 32\}$. The reference set for computing eps is the (exact) Pareto front. The reference point for computing hv is set to the origin. We experiment with both neutral and non-neutral walks. In order to ensure a reasonable runtime for neutral walks, we set a maximum budget of 10^7 evaluations and a cutoff of 30 consecutive iterations of the main loop without improvement. The neighborhood relation among solutions (\mathcal{N}) is defined by the *1-bit-flip* operator; i.e., two solutions are neighbors if the Hamming distance between them is one. We replicate each experiment 30 times with different random seeds.

Number of local optimal sets. We start by investigating the number of LO-sets of each type, that is, for each set preference relation (\preceq_{dom} , \preceq_{hv} , and \preceq_{eps}) and either strict or non-strict definition (LO and sLO). After running the adaptive walk as described in the previous section, we simply count how many of the sets returned at the end of the runs satisfy the definition of each type of LO-set. Results are shown in Figure 4.3 for selected settings.

The first observation is that non-neutral walks using a particular set preference relation always find a non-strict LO-set according to the same relation, in every run. That is, a walk based on \preceq_{dom} (respectively, \preceq_{eps} , \preceq_{hv}) always falls into an $\text{LO}_{\preceq_{\text{dom}}}$ (respectively, $\text{LO}_{\preceq_{\text{eps}}}$, $\text{LO}_{\preceq_{\text{hv}}}$). Moreover, the LO-set where a given walk falls into might be the same at different executions, as observed, for instance $m = 2$, $k = 0$ with $\mu = 32$. This suggests that there is a single LO-set in this case, which is not a surprise because the corresponding landscape is linear ($k = 0$) and its Pareto set cardinality is lower than $\mu = 32$.

We did not notice any difference between neutral or non-neutral walks with \preceq_{hv} , which suggests that neighboring LO-sets with the same hypervolume value are rare, and therefore there is no neutrality in the corresponding landscapes. By contrast, we observe a large neutrality for \preceq_{dom} and \preceq_{eps} , as shown by the large differences between neutral and non-neutral walks in such cases. In fact, the neutral walk using \preceq_{dom} is only able to find a $\text{sLO}_{\preceq_{\text{dom}}}$ when μ is large, and when

Table 4.1: Benchmark parameter values.

parameter	values
n	16
k	$\{0, 1, 2, 4, 8\}$
m	$\{2, 3, 5\}$
ρ	0.0

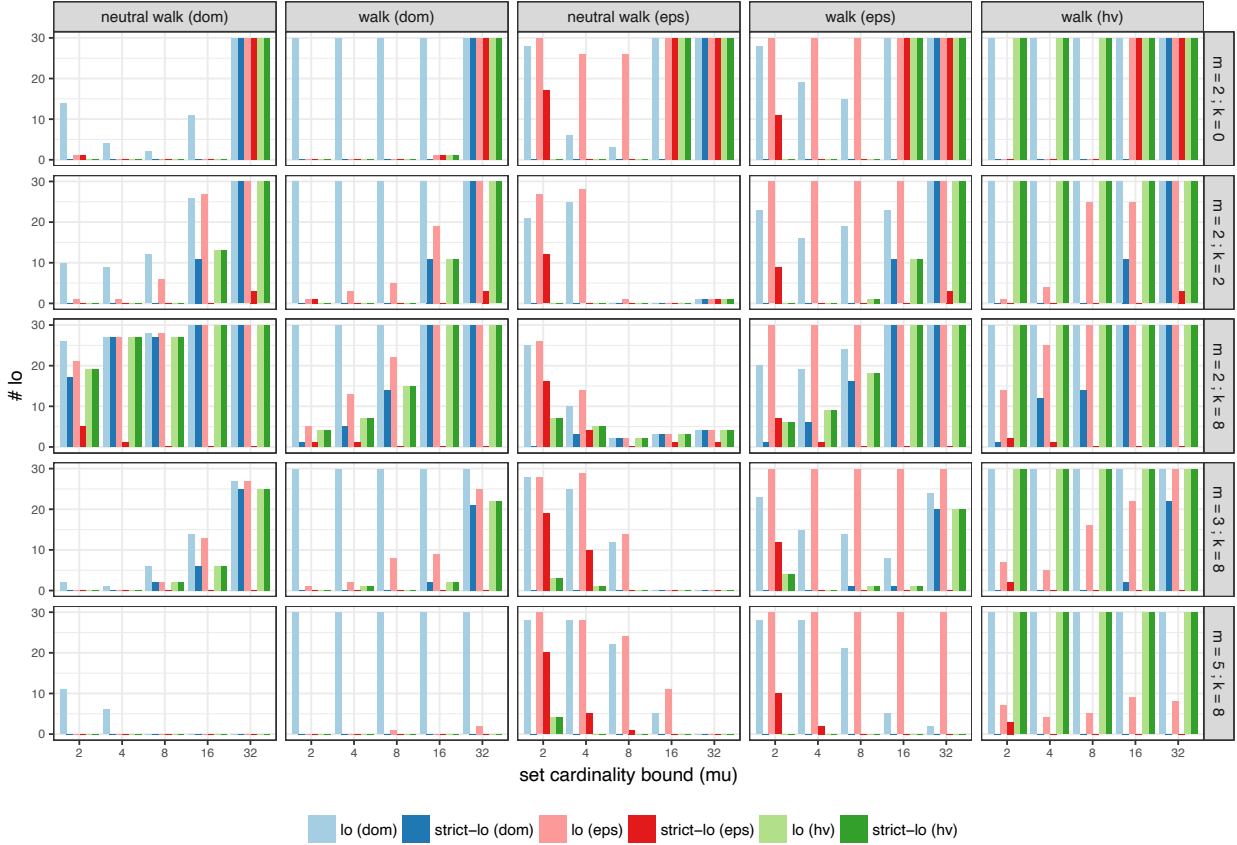


Figure 4.3: Number of each type of LO-set found by each type of LS walk (by column) for different instances (by row), depending on the set cardinality bound (μ). Results for neutral walk (hv) are not reported because they are the same as for walk (hv).

there are few objectives, e.g., for $m = 2$ and $k = 8$. We attribute this to a large non-linearity in the objective values in such cases, with many incomparable neighboring sets, which seem to increase the number of strict LO-sets. By contrast, the neutral-walk using \preceq_{eps} is only able to find a $\text{sLO}_{\preceq_{\text{eps}}}$ when μ is small and/or when the problem is linear ($k = 0$), whereas the neutral-walk using \preceq_{hv} is always able to find a $\text{sLO}_{\preceq_{\text{hv}}}$, as already reported above. The probable reason is that there are more neighboring sets with the same epsilon value and/or that the hypervolume gradient is easier to optimize than that of epsilon. Interestingly, when there are many objectives, and when μ is especially small relative to the size of the exact Pareto set, it appears to be difficult to obtain a $\text{LO}_{\preceq_{\text{hv}}}$ with any method besides a walk based on \preceq_{hv} .

To summarize, by comparing LO-sets under \preceq_{dom} and \preceq_{hv} , we conjecture that:

$$\text{sLO}_{\preceq_{\text{dom}}} \implies \text{sLO}_{\preceq_{\text{hv}}} \implies \text{LO}_{\preceq_{\text{hv}}} \implies \text{LO}_{\preceq_{\text{dom}}}$$

Although we do not obtain any clear trend by comparing LO-sets under \preceq_{dom} and \preceq_{hv} with LO-sets under \preceq_{eps} , we conjecture that:

$$\text{sLO}_{\preceq_{\text{dom}}} \implies \text{LO}_{\preceq_{\text{eps}}}$$

We also suspect that there are slightly more $\text{LO}_{\preceq_{\text{eps}}}$ than $\text{LO}_{\preceq_{\text{hv}}}$, given that the walk based on \preceq_{dom} consistently finds more of those, but the

difference seems to be rather small.

Finally, a general observation is that, whatever the set preference relation, the adaptive walk gets more easily trapped into a LO-set as the problem non-linearity k increases, and as the number of objectives m decreases. As a consequence, there are more local optimal sets for non-linear problems and/or problems with few objectives, whereas there are fewer local optimal sets for linear and/or many-objective problems.

Length of adaptive walks. As in single-objective optimization, the length of adaptive walks provides an estimation of the number of LO-sets. The number of steps performed by the algorithm defines the length of the adaptive walk. This length is an estimator of the diameter of local optima's basins of attraction. Roughly speaking and assuming isotropy in the search space, the longer the walk, the larger the basins size, and the lower the number of local optima (Kauffman, 1993); see also Section 3.2. Figure 4.4 reports the number of steps performed by each type of adaptive walk. In our experiments, the variable space has the same size for all instances; i.e. $|X| = 2^n = 2^{16}$, whatever k and m . However, the number of candidate sets depends on the set cardinality bound μ and on the dominance relations between solutions. For a given μ , the number of candidate sets is bounded by $\sum_{i=1}^{\mu} \binom{|2^X|}{i} = \sum_{i=1}^{\mu} (2^{|2^X|} - 1)$. Therefore, for a given instance, a larger value of μ induces an exponentially larger number of candidate sets. Of course, depending on the dominance relations between solutions, many candidate sets might be equivalent once dominated solutions are discarded.

We observe in Figure 4.4 that the length of adaptive walks typically increases with μ . Therefore, a local search is more easily stuck when μ is small. This means that the absolute number of LO-sets decreases with μ . When relating that to the number of candidate sets, we argue that the proportional number of LO-sets is larger when μ is small, whatever the set preference relation. As expected, the length of adaptive walks decreases with the problem non-linearity k . As for single-objective nk-landscapes, the larger k , the larger the number of local optima (Kauffman, 1993). When considering the number of objectives m , we observe that adaptive walks runs longer as m increases, especially when μ is relatively large. A local search has a larger probability of getting stuck for two-objective landscapes than for three- and five-objective landscapes. This suggests that the number of LO-sets decreases with the number of objectives.

When comparing neutral and non-neutral walks for a given set preference relation, we observe that neutral walks typically run longer. There are two potential explanations for this result: (i) a neutral walk may fail to identify a strict LO-set simply because it does not exist, explaining why the number of steps reaches the overall budget limit for some settings; and (ii) every strict LO-set under a given set preference relation is also an LO-set under the same relation, thus the number of sLO-sets is smaller or equal than the number of LO-sets. This is

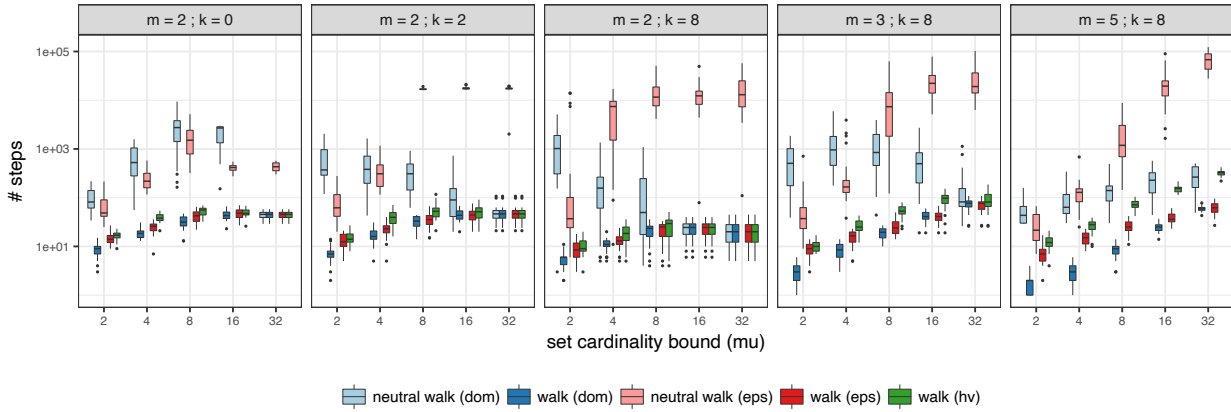


Figure 4.4: Number of steps performed by each walk (colors) for different instances (non-linearity k and number of objectives m , by column), depending on the set cardinality bound.

the case for \preceq_{dom} and \preceq_{eps} . Interestingly, the gap between the number of strict and non-strict LO-sets seems to decrease with μ for \preceq_{dom} , whereas it increases for \preceq_{eps} . In fact, for large sets, almost all $\text{LO}_{\preceq_{\text{dom}}}$ are $\text{sLO}_{\preceq_{\text{dom}}}$, as also noticed in Figure 4.3. We attribute this to the fact that it is more unlikely to come across a neighboring solution that is not dominated by the set when this set is larger. As already mentioned above, in the case of \preceq_{hv} , there is no distinction between neutral and non-neutral walks for the considered instances: an $\text{LO}_{\preceq_{\text{hv}}}$ is always a $\text{sLO}_{\preceq_{\text{hv}}}$.

Let us now compare non-neutral walks and non-strict LO-sets for different set preference relations. For $m = 2$ and $\mu \in \{16, 32\}$, the length of the adaptive walk is roughly the same for all relations. In fact, they are the sole settings where the cardinality of LO-sets is actually smaller than the bound μ , which is explained by the fact that μ is larger than the Pareto set in those cases. This suggests that there is no distinction between LO-sets under the different set preference relations when μ has the same order of magnitude than the Pareto set, as also observed in Figure 4.3. By contrast, for other instances, the length of the adaptive walk for \preceq_{dom} is typically smaller than for \preceq_{eps} , which is itself typically smaller than for \preceq_{hv} . This gives us more evidence that, when μ is smaller than the Pareto set, we have more $\text{LO}_{\preceq_{\text{dom}}}$ than $\text{LO}_{\preceq_{\text{eps}}}$, and more $\text{LO}_{\preceq_{\text{eps}}}$ than $\text{LO}_{\preceq_{\text{hv}}}$. A multi-objective local search with bounded archive is then expected to get more easily trapped when comparing sets in terms of dominance rather than in terms of epsilon or hypervolume. On the contrary, a hypervolume-based local search is expected to perform more steps before getting stuck.

Distance between local optimal sets. At last, we go deeper into the comparison of LO-sets under different set preference relations by investigating their dissimilarity in the space of sets. In particular, we want to know how much different is a $\text{LO}_{\preceq_{\text{eps}}}$ or a $\text{LO}_{\preceq_{\text{hv}}}$ from a $\text{LO}_{\preceq_{\text{dom}}}$. We do not consider strict LO-sets in this section, since they do not necessarily exist for all settings, and since the corresponding neutral walks are not always successful.

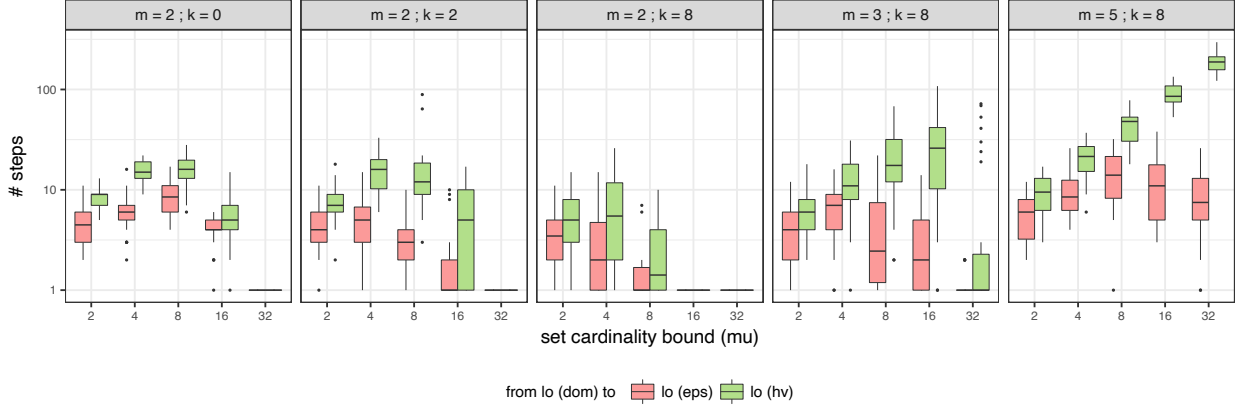


Figure 4.5: Number of steps performed by the walk to go from a $LO_{\preceq_{dom}}$ to a $LO_{\preceq_{eps}}$ and $LO_{\preceq_{hv}}$ for different instances (non-linearity k and number of objectives m , by column), depending on the set cardinality bound.

Let us define the *distance* between a $LO_{\preceq_{dom}}$ and a $LO_{\preceq_{eps}}$ as the length (number of steps) required by a walk based on \preceq_{eps} to reach a $LO_{\preceq_{eps}}$ while starting from a $LO_{\preceq_{dom}}$ as an initial set. To do so, (i) we simply start by running a walk under \preceq_{dom} until it falls into a $LO_{\preceq_{dom}}$, and then (ii) we run a walk under \preceq_{eps} starting from the obtained $LO_{\preceq_{dom}}$. Only the steps performed in the second phase are taken into account to measure the distance. The distance between a $LO_{\preceq_{dom}}$ and a $LO_{\preceq_{hv}}$ follows the same reasoning, but using a walk under \preceq_{hv} . This notion of distance gives how many *1-bit-flips*, performed on any solution from the initial set, separates a set A from a set B . Thus, if $dist(A, B) = d$, then A may differ from B in d solutions, all connected at Hamming distance 1, or they may differ in a single solution with Hamming distance d .

The obtained distances are reported in Figure 4.5. When compared against the walks that start from a random set, as reported in Figure 4.4, the number of steps performed from a $LO_{\preceq_{dom}}$ is lower by an order of magnitude. This means that a $LO_{\preceq_{dom}}$ is much closer to a $LO_{\preceq_{eps}}$ or a $LO_{\preceq_{hv}}$ than a random set is with any of the three. The distance between a $LO_{\preceq_{dom}}$ and a $LO_{\preceq_{eps}}$ is often larger for medium μ values than for small and large values. When considering the hypervolume, the distances from a $LO_{\preceq_{dom}}$ to a $LO_{\preceq_{hv}}$ are always larger than to a $LO_{\preceq_{eps}}$. When $m = 2$, these distances roughly follow the same trend as for $LO_{\preceq_{eps}}$. However, when $m \geq 3$, they seem to increase with μ , with the exception of $\mu = 32$ and $m = 3$ where the distance is close to 1. For $m = 5$, the gap relative to the distances corresponding to $LO_{\preceq_{eps}}$ increases by several orders of magnitude.

Summary

In this section, we empirically studied the properties of various types of LO-sets. We observed that the number of LO-sets of any type increases with the increase in the number of variable interactions, and with the decrease in the number of objectives and the cardinality bound. When comparing LO-sets under different set preference re-

lations, we advance several hypotheses based on our experimental results. In particular, we conjecture the following implications:

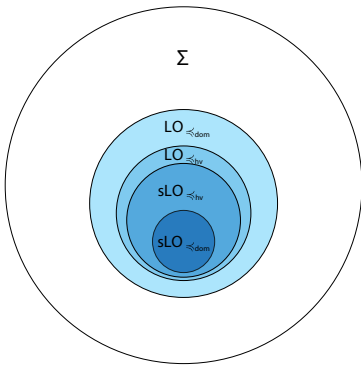
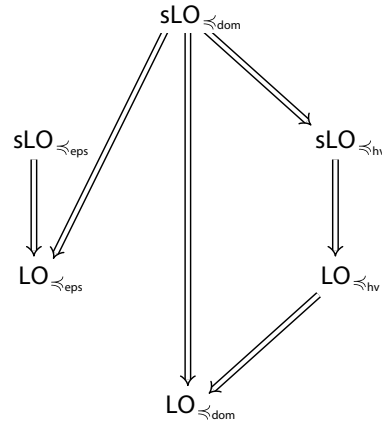


Figure 4.6: Relation among LO-sets under set dominance and hypervolume.

This means that there are more $LO_{\preceq_{dom}}$ than $LO_{\preceq_{hv}}$, but more $sLO_{\preceq_{hv}}$ than $sLO_{\preceq_{dom}}$, as illustrated in Figure 4.6. In addition to the implications above, we also observed that there are many more $LO_{\preceq_{dom}}$ than $LO_{\preceq_{eps}}$, and slightly more $LO_{\preceq_{eps}}$ than $LO_{\preceq_{hv}}$, the latter being perhaps the most surprising conclusion. Notice that additional results on the quality of LO-sets and the convergence profile of adaptive walks are reported in [Liefvooghe et al. \(2018b\)](#). They confirm previous findings and observations in multi-objective optimization, such as the fact that algorithms relying solely on dominance tend to perform worse for more than three objectives.

4.4 A Glimpse on Further Contributions

In this section, we summarize additional contributions related to indicator-based search for multi-objective optimization.

Hypervolume-based Anytime Algorithm Performance

As pointed out in Section 3.6, anytime algorithms allow a practitioner to trade-off runtime for approximation quality ([Dean and Boddy, 1988](#); [Dubois-Lacoste et al., 2015](#); [Zilberstein, 1996](#)). This is of particular interest in multi-objective optimization, since it might be infeasible to identify all Pareto optimal solutions in a reasonable amount of time. In [Jesus et al. \(2021b\)](#), we present a theoretical model that, under some mild assumptions, characterizes the “optimal” trade-off between runtime and relative hypervolume of anytime algorithms for bi-objective optimization. In particular, we assume that Pareto optimal solutions are collected sequentially, such that the collected solution at each iteration maximizes the hypervolume indicator. In addition, we propose a hypervolume-based ε -constraint algorithm, and show that our model can be used to drive the search and improve its anytime behavior.

Our theoretical model is based on a piecewise linear approximation of the Pareto front with ℓ segments. We assume that the objective values of the lexicographic optimal solutions are known, and that the Pareto front can be well approximated by the quadrant of a super-ellipse. This is illustrated in Figure 4.7 for $\ell = 2$ segments. Although it is not expected that the Pareto front matches the quadrant of a super-ellipse exactly, our findings suggest that this gives a good approximation in practice for many multi-objective combinatorial optimization problems. We also define an oracle that, at each call, returns a point of the piecewise linear approximation that maximizes the hypervolume contribution. This oracle is expected to reproduce the behavior of an anytime algorithm which sequentially collects Pareto optimal solutions that maximize the hypervolume contribution. We present two formulations of the model: (i) an analytical formulation for the particular case where the piecewise linear approximation consists of $\ell = 2$ segments and is convex, and (ii) a more general algorithm that works for both the convex and non-convex cases for a piecewise linear approximation defined by any number of linear segments.

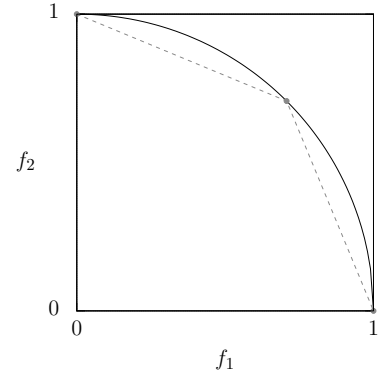


Figure 4.7: Example of a super-ellipse curve (continuous line) and the corresponding piecewise linear approximation (dashed line) for $\ell = 2$ segments.

We compare the relative hypervolume obtained from our model with an “optimal” model that has complete knowledge of the Pareto front and selects at each iteration a point such that the hypervolume is maximized. We also show the anytime behavior of two variants of an ϵ -constraint approach that collect Pareto optimal solutions by solving a sequence of constrained single-objective problems, where one of the variants is guided by our theoretical model, based on hypervolume. We perform this study on different unconstrained bi-objective knapsack problem variants. In Figure 4.8, we report the relative hypervolume deviation over runtime on exemplary problem instances, where runtime corresponds to the number of calls to the oracle for the theoretical and optimal models, and to the number of iterations for the ϵ -constraint algorithms. We consider $\ell = 2$ and $\ell = 10$ linear segments of the piecewise linear approximation of the Pareto front.

Our empirical results suggest that the theoretical model approximates the behavior of the optimal model quite well. In particular, for $\ell = 10$ linear segments, the theoretical model is very close to the optimal model. For $\ell = 2$ linear segments, there is a larger difference during the first few steps due to the difference between the maximal hypervolume of the Pareto front and of the piecewise linear approximation.

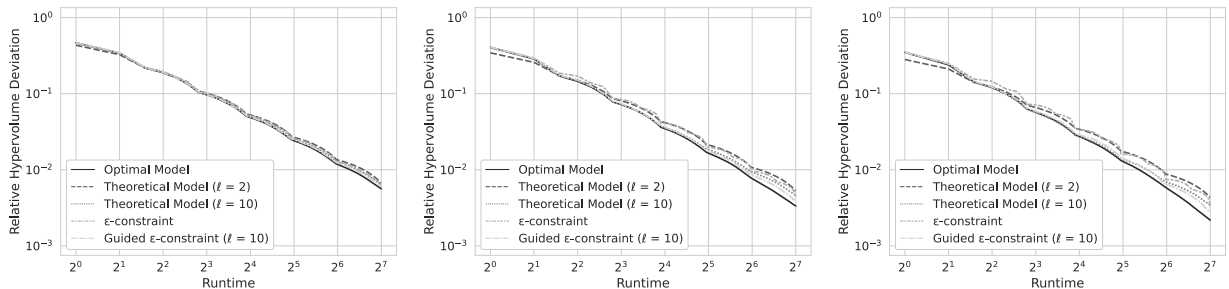


Figure 4.8: Results for the unconstrained bi-objective knapsack problem with a problem size $n = 100$, a reference point $r = (0, 0)$, and a varying correlation among the objectives $\rho \in \{-0.8, 0.0, 0.8\}$, from left to right.

Furthermore, the results show that the basic ε -constraint algorithm has a good anytime behavior for some instances, but that its behavior deteriorates for other instances. By contrast, the variant guided by our theoretical model shows very good anytime behavior on all the experiments. We note that these results are coherent for instances with different characteristics, and we expect the results to generalize for other problems where the Pareto front can be approximated by the quadrant of a super-ellipse.

Indicator-based Branch and Bound

In Jesus et al. (2021a), we propose a branch and bound approach for multi-objective combinatorial optimization whose selection is driven by a quality indicator. Multi-objective branch and bound recursively divides the solution space into several sub-problems at each branching step (Przybylski and Gandibleux, 2017; Visée et al., 1998). For each node of the search tree, we maintain (i) an upper bound set of objective vectors that weakly dominates all solutions to the sub-problem, and (ii) a lower bound set of mutually non-dominated solutions to the sub-problem (Ehrgott and Gandibleux, 2001; Przybylski and Gandibleux, 2017). We additionally keep an archive of the best non-dominated solutions from the lower bound sets of each node. The proposed indicator-based branch and bound uses a binary indicator in order to select the next node to be processed, by measuring the quality of its upper bound with respect to the archive. Although the branch and bound algorithm will eventually identify the whole Pareto set, we are particularly interested in analyzing its anytime behavior.

We compare two indicator-based branch and bound variants against the conventional depth-first (DFS) and breadth-first (BFS) node selection strategies (Visée et al., 1998). The best-first selection (BeFS) variant selects the most promising node with respect to the quality indicator, whereas the best-depth-first selection (BeDFS) variant only evaluates the nodes that are at the deepest level of the search tree. For BeFS and BeDFS, we consider the hypervolume and the epsilon indicator. We conduct experiments on a multi-objective knapsack problem with 2 to 7 objectives, by considering a random branching order (“default”) and a problem-specific branching order — the rank sum order from Bazgan et al. (2009). Figure 4.9 shows the anytime performance profile of the considered approaches with respect to the normalized hypervolume and CPU time. The results suggest that the indicator-based branch and bound outperforms the naive depth-first and breadth-first search strategies. The improvement is especially significant when the branching order is random, which suggests that the approach is particularly relevant when *no* favorable problem-specific branching order is available (Cerqueus et al., 2017). Moreover, the results indicate that the hypervolume-based branch and bound has a good anytime performance for a small number of objectives, whereas the branch and bound based on the epsilon indicator shows better performance for a larger number of objectives.

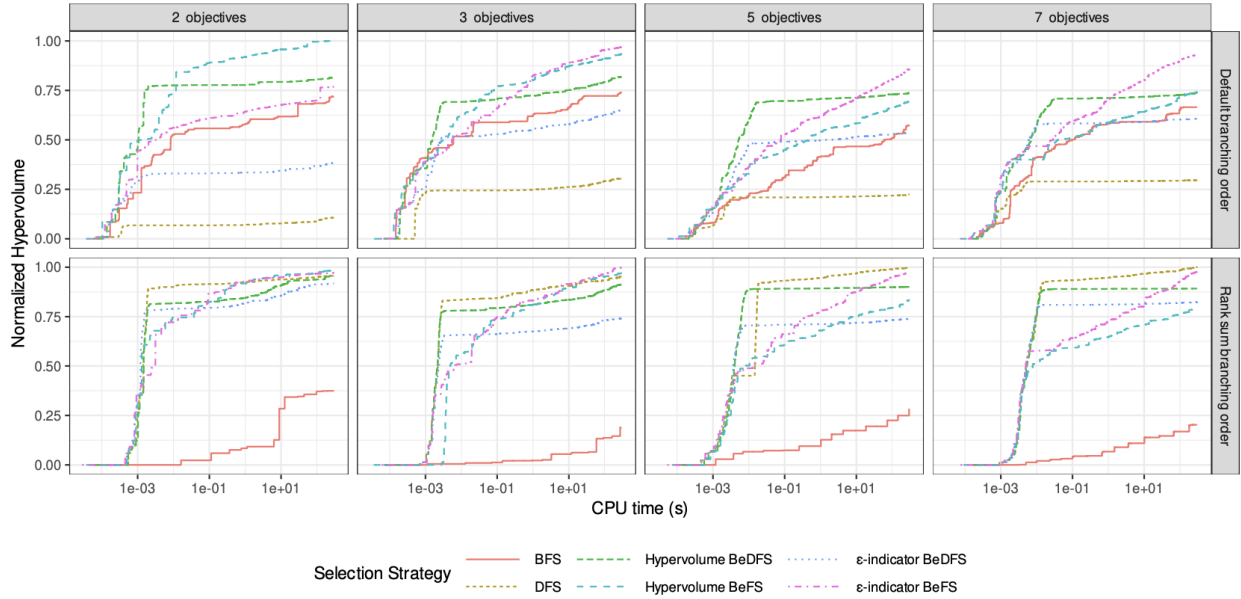


Figure 4.9: Anytime performance profile of the branch and bound approaches for multi-objective knapsack problem instances with a varying number of objectives and a problem size $n = 100$.

Hypervolume Subset Selection

Subset selection is an important stage of any (evolutionary) multi-objective optimization algorithm when truncating the current approximation set for the next iteration (Coello Coello et al., 2007; Deb, 2001). In particular, indicator-based selection has been intensively used in recent years for that purpose (Beume et al., 2007; Zitzler and Künzli, 2004). Subset selection is also relevant for post-processing all solutions returned by a multi-objective optimizer and presenting a representative subset to the decision maker (Bringmann et al., 2014a). This problem appears to be particularly challenging when the number of solutions to be removed is large, and when the approximation set contains many mutually non-dominating solutions. Most approaches for the indicator-based subset selection problem are based on a simple greedy backward elimination (GBE) strategy, that removes the worst solution with respect to the considered indicator, and iterates until the considered set shrinks to the target size (Beume et al., 2007; Zitzler et al., 2010). The subset returned by such a heuristic is generally *not* optimal, and alternative heuristics (Bradstreet et al., 2007), exact methods (Bringmann et al., 2014b; Kuhn et al., 2016; Vaz et al., 2015) and approximation algorithms with performance guarantee (Guerreiro et al., 2015) have also been proposed.

In Basseur et al. (2016), we investigate additional heuristics that include a greedy sequential insertion (GSI) and a greedy forward selection (GFS) policy, a first-improvement hill-climbing local search (LS), together with combinations of those. We evaluate their performance when maximizing the hypervolume of candidate subsets for different numbers of objectives, numbers of solutions, and subset sizes. We consider both *random* instances (i.e. randomly-generated sets of mutually non-dominated solutions) and *structured* instances (i.e. known Pareto

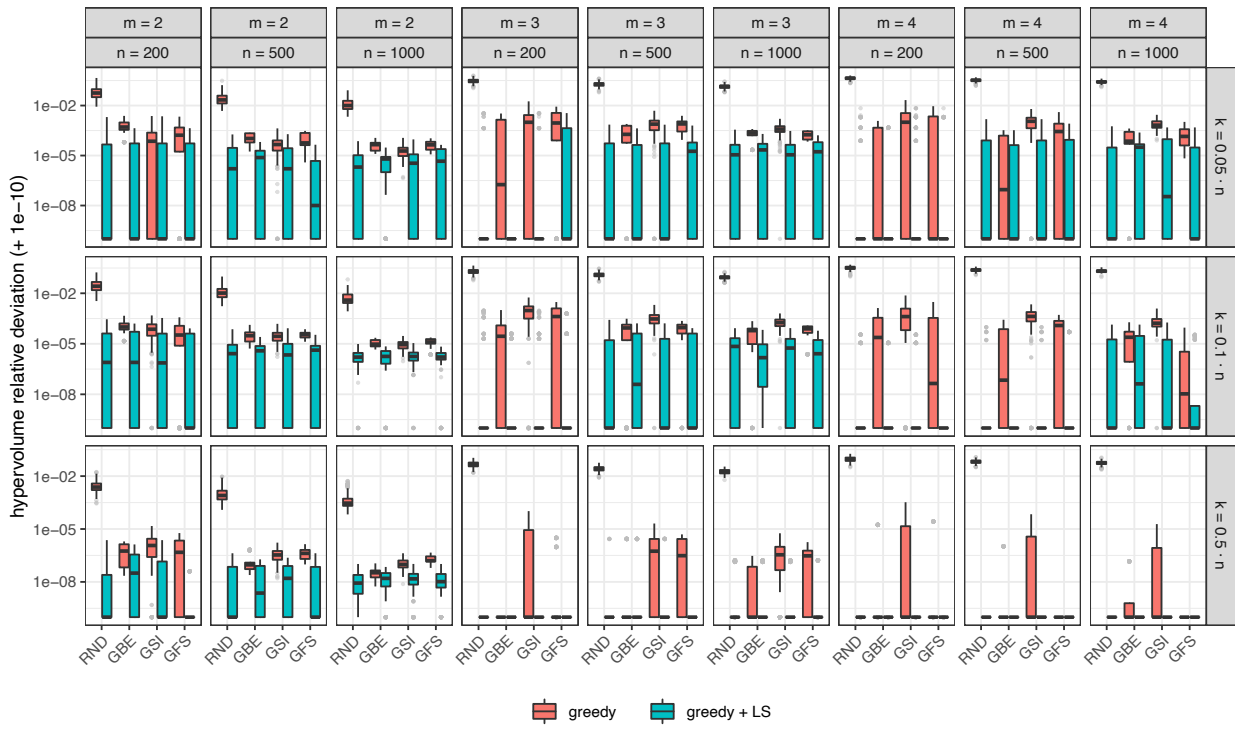


Figure 4.10: Approximation quality of hypervolume subset selection heuristics for random instances.

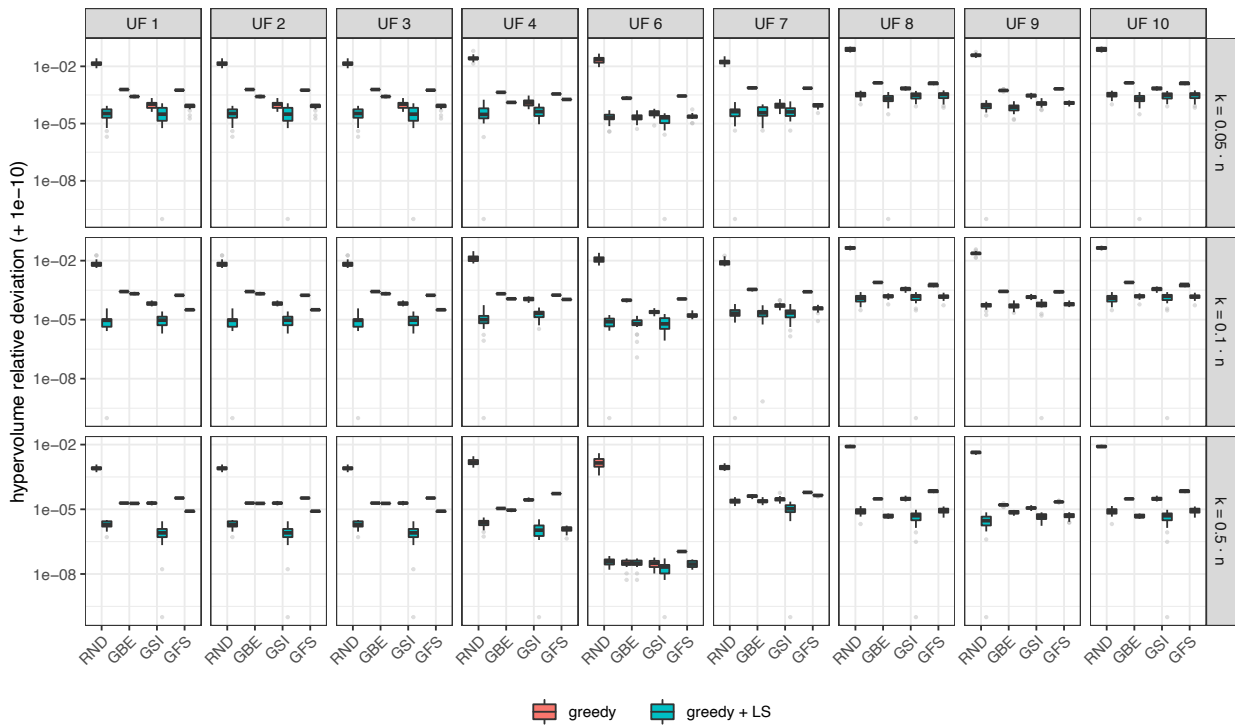


Figure 4.11: Approximation quality of hypervolume subset selection heuristics for structured instances.

fronts from Zhang et al. (2009)). Figure 4.10 reports the hypervolume relative deviation to the best-found subset for each heuristic on random instances. Figure 4.11 is the counterpart for structured instances. Our experimental findings reveal that the default greedy backward elimination (GBE) heuristic is outperformed by a local search, that does not necessarily require more computational resources. Similarly, GBE is outperformed by a greedy sequential insertion (GSI) heuristic with immediate truncation, especially on two-objective structured instances. This has important implications for hypervolume-based algorithms: instead of merging the parent and offspring populations and then iteratively removing the less contributing solution, it might actually be more efficient to iteratively insert offspring solutions to the population, and to immediately remove the worst-contributing solution after each insertion. Moreover, running a simple local search from the solution obtained by a greedy heuristic constantly leads to an improvement in terms of hypervolume, independently of the greedy heuristic, and with a minor computational overhead. This suggests that more efficient indicator-based algorithms could be designed, for which it would be worth measuring the cumulative improvement as the search progresses.

An Ensemble Indicator-based Selection Approach

Indicator-based (evolutionary) multi-objective optimization algorithms make use of an indicator in order to guide the selection of their search process (Beume et al., 2007; Zitzler and Künzli, 2004). As such, it is well understood that an indicator-based approach targets an approximation set exhibiting the preferences of its underlying indicator, and that different algorithms yield different results in terms of the distribution of solutions in the approximation set, due to the underlying properties of the indicator being used. Moreover, a specific indicator-based algorithm might perform well on some problems, but there might exist others on which it does not. As a consequence, it remains unclear which indicator to consider beforehand, and an open question is whether a set of indicator-based selection mechanisms can reach a consensus that outperforms existing ones.

In machine learning, ensemble methods aim at improving predictive performance by combining multiple stand-alone models (Zhou, 2021). In Falcón-Cardona et al. (2020), we consider an *ensemble* of indicators, and we propose an ensemble indicator-based selection approach for multi-objective optimization. We rely on adaptive boosting (Freund and Schapire, 1997) to combine the features of multiple indicators, and we experiment with the hypervolume hv (Zitzler and Thiele, 1998), the $r2$ (Hansen and Jaszkiewicz, 1998), the inverted generational distance plus $igd+$ (Ishibuchi et al., 2015), the additive epsilon eps^+ (Zitzler et al., 2003), and the averaged Hausdorff distance Δ_p (Schütze et al., 2012) indicators. At each iteration, the ensemble mechanism adaptively selects solutions by using a learning process that takes the preferences of the underlying indicators into account. We employ a number of multi-objective continuous benchmark functions, and we compare

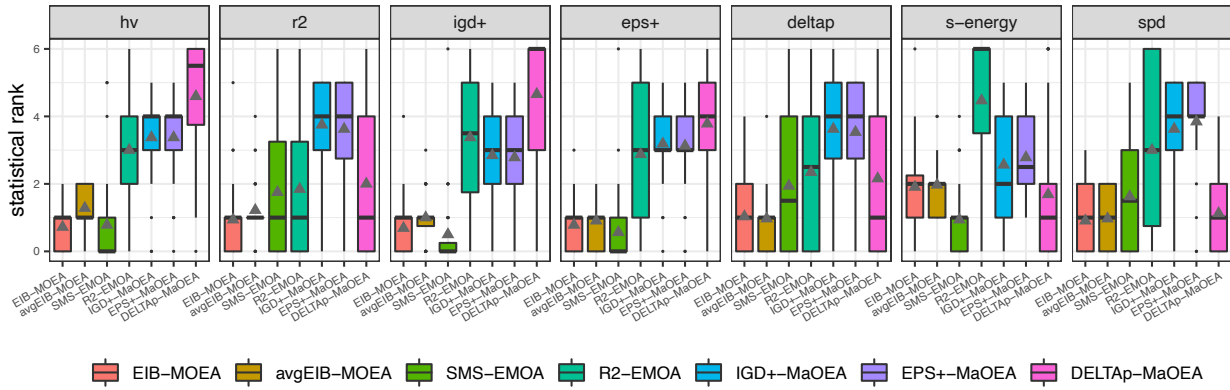


Figure 4.12: Statistical ranks obtained by each algorithm over all benchmark functions with respect to each considered indicator.

the ensemble approach against SMS-EMOA (Beume et al., 2007), R2-EMOA (Brockhoff et al., 2015), and three other variants which use the hv , $r2$, igd^+ , eps^+ , and Δ_p indicators, respectively. Figure 4.12 shows the statistical ranks obtained by each algorithm over all considered benchmark functions with respect to each indicator. The rank corresponds to the number of algorithms that significantly outperform the one under consideration (a lower rank is better). Our experimental results show that an adaptive ensemble indicator-based algorithm (EIB-MOEA) outperforms an average-ranking ensemble approach (avgEIB-MOEA) that sets all indicators as equally important throughout the search process. When comparing the ensemble approach against existing algorithms, we see that it maintains a strong performance with respect to the different indicators. Therefore, we argue that such a strategy positively influences the robustness of selection in multi-objective search.

4.5 Outlook and Current Investigations

This chapter focused on a recent paradigm for multi-objective optimization: set- and indicator-based search. We first reviewed selected quality indicators, and experimentally investigated the degree of agreement and contradiction in the order they induce among approximation sets. Our analysis highlights important insights regarding performance evaluation, preference interpretation, and algorithm design. First, there is no single indicator that captures all facets of approximation quality, although none of them vigorously conflict with others. Second, the correlation of the epsilon indicator and of the inverted generational distance with other considered indicators is particularly low, which suggests that they seem to focus on complementary aspects of approximation quality. For this reason, it seems relevant to revisit some benchmarking analyses, such as those of the CEC 2009 competition where the inverted generational distance was the only performance measure considered, in order to enhance our knowledge and understanding of competing algorithms by means of complementary indicators. Moreover, the hypervolume shows a strong correlation with the R-metrics for sub-optimal approximation sets. This suggests that the

latter could be used as a substitute of the former to speed up the computations while providing a satisfying accuracy in those cases. This seems particularly appropriate for procedures that require many calls to the indicator, such as hypervolume-based search approaches (Bader and Zitzler, 2011; Beume et al., 2007).

Next, we extended the idea of set-based multi-objective optimization to set-based landscapes and local search by introducing a neighborhood relation among sets. Such a framework formalizes and shares similarities with state-of-the-art multi-objective approaches, but also enables the design and analysis of original methodologies. Different definitions of a set-based multi-objective landscape, in terms of set space, set neighborhood relation, or set preference relation, lead to different dynamics of the search process. Of particular interest is the characterization of the local optimal sets in which set-based search algorithms fall into for different set preference relations, such as set dominance, hypervolume or epsilon indicator. Our experimental results reveal that, whatever the set preference relation, the number of local optimal sets typically increases with the number of variable interactions, and decreases with the number of objectives. We observe that strict local optimal sets of bounded cardinality under set dominance are local optimal sets under both epsilon and hypervolume, and that local optimal sets under hypervolume are local optimal sets under set dominance. Nonetheless, local optimal sets under set dominance are more similar to those under epsilon than hypervolume. These findings have important implications for multi-objective local search, and shed some light about a number of open research questions. In particular, set-based multi-objective landscapes with fewer objectives and/or fewer solutions appear to be more multimodal, suggesting that it might be harder for multi-objective local search to find a good approximation set in these cases. In addition, set-based multi-objective landscapes under dominance seem to be more multimodal — although they are more *strictly* multimodal under indicators. As such, we argue that a dominance-based approach with bounded archive gets more easily trapped and might experience difficulty to identify a local optimal set under epsilon or hypervolume. On the contrary, a hypervolume-based approach is expected to perform more steps before converging to better approximations.

On the same line, we introduced a theoretical model of performance for bi-objective optimization approaches that collect, at each iteration, a solution that maximizes the hypervolume. We show that such an analytical model is able to finely predict the anytime performance of ε -constraint algorithms for previously unseen instances, and that it can be further generalized to an algorithmic model that requires less assumptions. After that, we proposed three different approaches for improving the selection of indicator-based search. The first one is an indicator-based branch-and-bound framework that makes use of a quality indicator for selecting the best node to be explored at each iteration. By thoroughly investigating its components, our empirical results show promising results in terms of anytime performance, especially when no favorable (problem-dependent) branching order is available, although we might consider alternative branching strategies

from Cerqueus et al. (2017). The second one looks into the subset selection problem to identify the subset of solutions that jointly optimize a quality indicator, such as hypervolume. We report that the conventional heuristic is in most cases outperformed, and can be further improved by means of a simple local search, without a significant computational overhead. We argue that such a strategy could improve the selection mechanism of multi-objective search. The last contribution is based on the observation that different indicators are not necessarily consistent about the solutions to retain during indicator-based selection. For this, we adopt an ensemble method which scores the solutions on the basis of multiple indicators, so as to offer a good compromise among them. Following a recent survey on quality indicators in multi-objective optimization (Li and Yao, 2019), our contributions are concerned with the following open challenges: (i) indicator selection, especially with respect to the use of an ensemble of indicators; (ii) connection between indicators, for which we measure the degree of agreement; (iii) optimal distribution, that directly relates to subset selection; and (iv) indicator-based search. Another survey on indicator-based multi-objective evolutionary algorithms by Falcón-Cardona and Coello Coello (2020) lists the design of multi-indicator-based (i.e. ensemble) approaches, and the theoretical speed of convergence and distribution of solutions as important future research directions.

Apart from considering complementary benchmarks with additional, potentially evolving samplings of sets, we plan to expand our correlation analysis with further quality indicators. They include the modified inverted generational distance (Ishibuchi et al., 2015), a variant which in turn is order-preserving with respect to set dominance, and also the average Hausdorff distance (Schütze et al., 2012), which generalizes several indicators. For the latter, we anticipate that different parameter settings will exhibit different degrees of correlation with other indicators. In addition, it also seems relevant to visually inspect the optimal (or close to optimal) distribution of solutions in the objective space with respect to different indicators, as well as the potential compromises among them. All together, we expect this will strengthen our knowledge on the relation among indicators and on the underlying mechanisms that explain their differences. Not only this, this might eventually lead to a more efficient indicator-based search.

Regarding local optimal sets, we conjecture that our findings regarding the epsilon indicator (respectively, the hypervolume) generalize to any order-preserving (respectively, strictly order-preserving) quality indicator. In fact, we are currently polishing the formal proof of some of our conjectures, thus increasing our theoretical understanding of set-based multi-objective landscapes. Furthermore, we are extending our experimental analysis to additional problems, neighborhoods and order-preserving indicators, in order to corroborate that our conjectures indeed generalize as expected. Of particular interest is the extension of our work to local optimal sets for continuous problems (Grimme et al., 2021; Kerschke et al., 2016). In terms of set neighborhood, the relations introduced in our work can be extended with higher degrees of reduction or expansion over the set cardinality, and with more general operators over populations like recombination. However, one of the

main computational issue within such very-large set neighborhood structures deals with their efficient exploration (Basseur et al., 2013). To analyze set-based landscapes induced by the continuous domain and by large set neighborhoods, we plan to explore sampling techniques by discretization of the solution space (Liefvooghe et al., 2021), as well as landscape techniques for stochastic search operators (Aboutaib et al., 2020). Moreover, a similar research plan will allow us to make a stronger connection with the previous chapter, by designing set-based landscape features through sampling, thus promoting the uptake of feature-based prediction tasks for the set-based view of multi-objective search. We believe our findings could also allow deriving practical recommendations to support the design of new multi-objective search heuristics. For instance, an epsilon-based local search does not necessarily converge to better epsilon values than a hypervolume-based local search, unless it explicitly handles sets that are equivalent in terms of epsilon.

In terms of problem solving, the main challenge facing set-based (local) search is the large cardinality induced by some set neighborhoods, as already pointed out for landscape analysis. We see two complementary ways of approaching this. First, at the solution level, it deals with benefiting from any fast (incremental) neighborhood or recombination exploration, although this breaks the black-box nature of the problem, at least partially; see, e.g., Whitley et al. (2016) or Abdelkafi et al. (2020). Second, at the set level, we see subset selection as a way to identify or approximate the best neighboring set. In fact, indicator-based subset selection can be formulated as a pseudo-boolean (single-objective) optimization problem, for which an efficient surrogate meta-model has recently been proposed (Verel et al., 2018), as also discussed in the next chapter. In a complementary way, we aim at exploring indicators for which multiple solutions could be identified independently, either to parallelize the search process or to evaluate the set quality (i.e. the indicator value) incrementally. For instance, we believe the r_2 and r_3 indicators could be good candidates for this, given their satisfying properties (not strictly order-preserving, but close to the hypervolume) and their conceptual connection with other multi-objective search paradigms such as the decomposition approaches considered in the next chapter.

Efficient Multi-objective Search

In this last chapter before concluding, we describe our research work on efficient multi-objective search. By relying on the core concept of decomposition, we precisely aim at improving the anytime behavior of algorithms, that is at identifying better solutions in a shorter amount of time. The main challenge deals with the design and analysis of advanced algorithm components in order to improve the search ability. To do so, we first make the most of the high-level parallelism enabled by decomposition-based methods. This allows us to propose three distributed approaches by revisiting the design of sequential decomposition-based search, so as to provide a good balance between parallel runtime and approximation quality (Derbel et al., 2014b, 2015; Shi et al., 2017b). Secondly, we consider the integration of surrogate meta-models to accelerate multi-objective search. This is relevant in the context where the evaluation of the objectives turns out to be particularly expensive. We present a taxonomy and a comparative study of existing methods for expensive multi-objective continuous optimization (Bervoglieri et al., 2019, 2022), and we propose a surrogate-assisted approach for expensive multi-objective combinatorial optimization (Pruvost et al., 2020b, 2021). We conclude the chapter by summarizing contributions related to population, selection and variation, and by discussing further considerations.

5.1 Motivations	77
5.2 Decomposition	80
5.3 Parallel Search	81
5.4 Surrogate-assisted Search	89
5.5 Further Contributions	96
5.6 Outlook	102

5.1 Motivations

In this chapter, we are interested in the foundations of computationally efficient multi-objective search, with a particular focus on *decomposition* (Zhang and Li, 2007). Decomposition-based multi-objective search consists in transforming the original multi-objective optimization problem into a number of scalar sub-problems that are being solved simultaneously and cooperatively. Due to its simplicity and flexibility, the decomposition framework has recently gained in popularity and received a renewed interest from the community. In particular, the so-called multi-objective evolutionary algorithm based on decomposition (MOEA/D) from Zhang and Li (2007) and its many variants is now considered as a reference in the field. On top of offering a high efficiency compared against existing algorithms, our interest in MOEA/D stems from its ability to leverage existing single-objective search procedures and components, while also enabling parallelism in a rather natural way. The origins of MOEA/D can be traced back to Murata et al. (2001) in evolutionary computation, and even earlier than this in multi-criteria decision analysis and traditional optimization, where similar principles date from the late 1970s (Ehrgott, 2005; Miettinen, 1999). The main differences with MOEA/D lie in the fact that (i) the search process benefits from solving the sub-problems concurrently and cooperatively, the solution from one sub-problem contributing to the solution from neighboring sub-problems and vice versa, and (ii) it

emphasizes in a comprehensive way the importance of the scalarizing functions being used to define the sub-problems and to drive the search accordingly. Based on these two aspects, and motivated by the success and effectiveness of this framework, a number of other investigations followed and continue to be actively developed today (Li, 2021; Trivedi et al., 2017). In order to accelerate the convergence of (black-box) multi-objective search by identifying high-quality solutions earlier, and thus improve its anytime behavior, we address three main challenges in our work, which are summarized below.

Given that its search process is in essence distributed among the different sub-problems, the decomposition also implies the distribution of the underlying computations over a possibly large-scale computing environment. Two interrelated issues are typically addressed in *parallel* multi-objective search: (i) how to reduce the computational complexity of a specific multi-objective algorithm, and (ii) how to make the parallel processes cooperate in order to improve the quality of the approximation set (Coello Coello et al., 2007; Talbi et al., 2008). When decomposing the search process into parallel sub-processes targeting different regions of the Pareto front, the challenge is to maintain the search ability of the algorithm while attempting to break down the dependencies in the computational flow of its original sequential implementation (Nebro and Durillo, 2010). On top of that, it remains unclear how to deal with the fine-grained parallelism encountered when effectively deploying the parallel search in a large-scale distributed environment. In this chapter, we attempt to address these challenges while gaining more insight into the best practices one has to follow when adopting a parallel decomposition approach. We derive parallel approaches that exhibit a good balance between approximation quality and acceleration in the context of large-scale parallelism, where a fine-grained workload may drastically prevent good performance, especially in the scenario where the communication cost is non-negligible compared to the cost of evaluating the objectives. First, we deal with the adaptive and distributed setting of the weight vectors used by the decomposition (Derbel et al., 2014b). We show that using highly local distributed rules to define the search directions in an adaptive way leads to a high approximation quality and parallel efficiency. Second, we design a fine-grained message-passing parallel MOEA/D approach, which is to our knowledge the first to achieve parallel scalability while offering a good balance between approximation and acceleration (Derbel et al., 2015). Third, we propose a decomposition-driven parallel strategy for Pareto local search (Shi et al., 2018, 2017b). We show that decomposition allows the parallel variant to significantly speed up the search process while maintaining about the same approximation quality.

The second challenge that we address in this chapter deals with the use of *meta-models* to assist the search process when creating improving candidate solutions, based on an efficient estimation of their objective values. This turns out to be of particular interest when evaluating the objectives has a high computational cost; e.g., when optimizing expensive black-box simulations. This also relates to multi-fidelity optimization, where the evaluation of objective(s) involves several

models with different levels of fidelity, and therefore more or less expensive (Forrester et al., 2007). The issue is thus to find the right balance for the calls of the fast but unreliable model, or of the slow but more accurate one. We here consider that the meta-model acts as a surrogate for the expensive objectives. As such, we argue that meta-models could be wisely coupled with the search process to boost its performance, and thus to ease the solving of expensive multi-objective optimization problems (Bartz-Beielstein and Zaefferer, 2017; Jin, 2011). However, no systematic investigation has been carried out on how the cooperation and synchronization between search and learning could be managed, while carefully accounting for the additional phases of training and predicting. Moreover, the effective integration of a meta-model as a substitute for the expensive evaluation function can only be the result of a proper choice of its inputs-outputs, and of its parameters. In this chapter, we conduct a systematic investigation on the integration of meta-models within multi-objective search (Berveglieri et al., 2019, 2022). We focus on the choice of the dataset from which the meta-model is trained, on the output to be predicted by the meta-model, and on how the meta-model is used to generate candidate solutions among which a selected subset will undergo the real, expensive evaluation. This allows us not only to highlight the main components to consider for successfully solving expensive optimization problems, but also to provide a taxonomy and to define a common ground for existing surrogate-assisted multi-objective search approaches. On the same line, we point out the lack of suitable meta-models for discrete problems, even when dealing with expensive single-objective optimization. By relying on Walsh functions as a surrogate for pseudo-boolean functions (Verel et al., 2018), we design a surrogate-assisted modular approach based on decomposition for expensive multi-objective combinatorial optimization (Pruvost et al., 2020b, 2021). To the best of our knowledge, this constitutes the first study on the subject, and our experiments show the clear benefit of the method to accelerate the convergence of decomposition-based search.

Finally, and in complement to the aforementioned considerations, setting up an effective multi-objective search approach relies on the design and the integration of algorithm components that can be configured in different ways. As with any general-purpose search heuristic, the specification of these components and their combination is of crucial importance. In our case, they deal with the decomposition of the initial problem into sub-problems and the specification of regions where promising solutions are to be identified, together with the solving of these sub-problems and the guidance of the search process within the so-defined regions. We investigate the coordination of these components, their joint setting, and the interaction between the computations from the different regions defined by decomposition, with the aim of reducing the computational cost of the underlying global search procedure while improving its efficiency. More specifically, we are interested in the specification of the population, in how the selection mechanism and the scalarizing functions approach the regions of interest, and in the intelligent design of effective variation operators for producing good-quality candidate solutions.

The chapter is organized as follows. In Section 5.2, we provide a brief overview of decomposition-based multi-objective search, while stating our general scientific interest in the MOEA/D framework. In Section 5.3, we present our contributions on the design of distributed search methods, which provide a high level of parallel scalability while revealing non-trivial trade-offs in terms of approximation quality and acceleration. In Section 5.4, we discuss the integration of meta-models within multi-objective approaches in order to speed-up the search and thus to address computationally expensive problems from continuous and combinatorial multi-objective optimization. In Section 5.5, we summarize complementary contributions for efficient multi-objective search in terms of population, selection and variation. In the last section, we conclude the chapter and we identify a number of open issues on these research topics.

5.2 Decomposition

Most of the contributions covered in this chapter rely on the concept of *decomposition*, briefly introduced in Section 2.3. We start by giving more details on its principles below. Let us remind that we aim at finding a Pareto set approximation for a multi-objective optimization problem $f: X \mapsto Z$, such that X is the variable space and Z is the objective space, and such that each objective $f_i, i \in \{1, \dots, m\}$, is to be maximized.

MOEA/D (*multi-objective evolutionary algorithm based on decomposition*) from Zhang and Li (2007) has become one of the most popular algorithm among decomposition-based heuristics. It decomposes the original multi-objective optimization problem into a number of scalar (single-objective) sub-problems that target different regions of the Pareto front. Each sub-problem is defined by a particular weight vector for the considered scalarizing function. Different scalarizing functions can be used (Miettinen, 1999), and popular examples are the weighted sum (g_{ws}) and the weighted Chebyshev (g_{cheb}) functions:

$$g_{ws}(x | \lambda) := \sum_{i=1}^m \lambda_i \cdot f_i(x) \quad ; \quad g_{cheb}(x | \lambda) := \max_{i \in \{1, \dots, m\}} \lambda_i \cdot |z_i^* - f_i(x)|$$

where $x \in X$ is a solution, $\lambda = (\lambda_1, \dots, \lambda_m)$ is a weighting coefficient vector such that $\lambda_i \geq 0$ for all $i \in \{1, \dots, m\}$, and $z^* = (z_1^*, \dots, z_m^*)$ is a reference point such that $z_i^* \geq f_i(x)$ for all $x \in X$ and $i \in \{1, \dots, m\}$.

A set of uniformly-generated weight vectors $(\lambda^1, \dots, \lambda^\mu)$ defines the scalar sub-problems, for which one solution is maintained and evolved in the population. This principle is illustrated in Figure 5.1. More particularly, given a scalarizing function $g: X \mapsto \mathbb{R}$, MOEA/D seeks a solution $x \in X$ with the best scalarizing function value $g(x | \lambda^i)$ for each sub-problem $i \in \{1, \dots, \mu\}$. To this end, it maintains a population $P = (x^1, \dots, x^\mu)$ such that each individual is the current solution for the corresponding sub-problem. Therefore, the population size exactly matches the number of weight vectors $\mu \in \mathbb{N}^+$. Moreover, for each

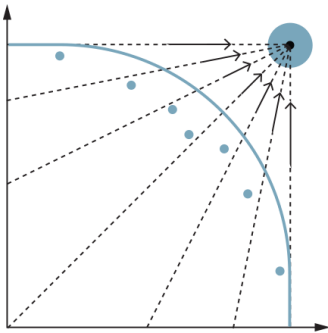


Figure 5.1: Schematic view of the decomposition principles from MOEA/D.

sub-problem $i \in \{1, \dots, \mu\}$, a set of *neighboring sub-problems* $B(i)$ is defined with the T closest weight vectors, itself included. The population evolves such that sub-problems are optimized iteratively and cooperatively based on this neighborhood relation. At a given iteration, and for a given sub-problem $i \in \{1, \dots, \mu\}$, some solutions are selected from $B(i)$ and an offspring y is created by means of variation operators. Next, for every sub-problem $j \in B(i)$, the offspring y replaces the current solution x^j if there is an improvement with respect to the scalarizing function; i.e. if $g(y | \lambda^j)$ is better than $g(x^j | \lambda^j)$. The algorithm loops over sub-problems, attempting to improve them one after the other, until a stopping condition is satisfied.

The MOEA/D algorithm is now considered as the state of the art in decomposition-based multi-objective search approaches. Numerous versions and extensions have been proposed⁶ covering a wide range of optimization problems and scenarios (Li, 2021; Trivedi et al., 2017). Our contributions are concerned with different research lines, some of which some are presented in the rest of this chapter. We give a particular focus to algorithm efficiency by making the most of the underlying parallelism offered by decomposition, and by exploring the integration of meta-models to assist the search process.

6: The MOEA/D website provides a repository of the state-of-the-art developments on decomposition-based multi-objective optimization: <https://sites.google.com/view/moead/>.

5.3 Parallel Multi-objective Search

Quite naturally, the decentralized nature induced by decomposition enables *parallelism*, in the sense that the computations inherent to the different sub-problems could be intuitively distributed over a parallel computing environment. As such, the distributed nature of decomposition-based algorithms offers new opportunities for designing parallel approaches that can be deployed over a massively parallel platform when dealing with large-scale or computationally-intensive problems. Nevertheless, considering a computational platform simply as a physical medium to be used in a straightforward manner can unavoidably result in incompatibility issues when deploying the designed algorithm in a parallel environment. A more appropriate approach is thus to design the distribution of the computational flow till the beginning, at the time the different algorithm components are specified. This constitutes the challenge we address below. We begin with the design of novel adaptive and local rules that can dynamically adjust the weight vectors (Derbel et al., 2014b). We continue with the design of a parallel MOEA/D approach and we analyze its potential in solving large-size optimization problems (Derbel et al., 2015). We finish with the design of a parallel Pareto local search driven by decomposition (Shi et al., 2018, 2017b).

Adaptive Distributed Localized Search

We start by presenting an adaptive distributed mechanism for setting the weight vectors used in decomposition (Derbel et al., 2014b). We remark that this is in line with a prior contribution where weight

Algorithm 2: DLBS — High-level algorithm to be executed by every computing node $i \in \{1, \dots, \mu\}$.

```

1  $x^i \leftarrow$  initial solution corresponding to node  $i$ ;
2 repeat
   // communicate positions
3    $z^i \leftarrow (z_1^i, z_2^i)$  the position of solution  $x^i$  in the bi-objective space
   (i.e.  $z^i = f(x^i)$ );
4   send  $z^i$  to neighboring nodes;
5    $(z^{i-1}, z^{i+1}) \leftarrow$  receive neighboring positions;
   // variation
6    $x \leftarrow$  mutation( $x^i$ );
   // replacement
7   if  $f_\ell(x)$  better than  $f_\ell(x^i)$  then
8      $x^i \leftarrow x$ ;
9 until a stopping condition is satisfied;

```

vectors are dynamically set with respect to the position of the whole population in the objective space (Derbel et al., 2013). A singularity of our proposal is the design of a cooperative and distributed approach which is inherently *local*, meaning that it is thought to be independent of any global knowledge, thus making it particularly suitable for a large-scale distributed environment. The proposed distributed localized bi-objective search (DLBS) algorithm is specifically designed for two-objective optimization problems, and is given in Algorithm 2.

Given a number of distributed computing nodes, our aim is to self-coordinate them locally, in order to cooperatively and adaptively cover different regions of the Pareto front. For clarity, we consider the setting where each processing unit evolves a single solution; i.e. there is a one-to-one mapping between the processing units and the solutions from the population. As illustrated in Figure 5.2, communication is organized following a line graph, where each node exactly has two neighbors, except those being at the extremes of the line. This is somehow similar to an island model (Candan et al., 2013) with a particular structure and inherently different migration mechanism. Based on this line graph, we design *local* rules relying solely on the relative position of neighboring solutions in the objective space. Each node thus simply communicates the objective values of its incumbent solution to its neighboring nodes, and reciprocally receives their positions, at each round. The rules are based on a *localized (scalar) fitness function* f_ℓ to be optimized locally, based on the position of its neighbors, such that every node focuses on a different region of the objective space.

We consider two alternatives for the localized fitness function, such that the search process focuses on the sub-region being orthogonal to the current position of the node's neighbors. This is with the exception of the two extreme nodes, that are respectively guided by one objective, independently of the other. The first localized fitness function is based on a weighted sum, and is inspired by the orthogonal direction of dichotomic search (Aneja and Nair, 1979). Given a candidate



Figure 5.2: Communication graph and expected evolution of nodes guided by the localized fitness function.

solution $x \in X$ and a distributed node $i \in \{1, \dots, \mu\}$, the score is:

$$f_\ell^{od}(x) := (z_2^{i-1} - z_2^{i+1}) \cdot f_1(x) + (z_1^{i+1} - z_1^{i-1}) \cdot f_2(x)$$

where (z^{i-1}, z^{i+1}) is the current objective vectors of the two neighbors. The second localized fitness function is based on hypervolume:

$$f_\ell^{hv}(x) := \begin{cases} (f_1(x) - z_1^{i-1}) \cdot (f_2(x) - z_2^{i+1}) & \text{if } f_1(x) \geq z_1^{i-1} \wedge f_2(x) \geq z_2^{i+1} \\ 0 & \text{otherwise} \end{cases}$$

With this second function, we argue that selecting candidate solutions maximizing the local hypervolume contribution at each node will improve the global hypervolume of the whole population. The two previously-defined localized fitness functions are illustrated in Figure 5.3. They both aim at distributing each solution at best in-between its two neighbors. They are to be maximized locally at each node using a standard evolutionary algorithm.

We experiment the proposed DBLS approach with both localized fitness functions on two-objective ρmnk -landscapes. We also investigate its scalability when using a different number of processing units, corresponding to the population size μ , on a computer cluster of hundreds of cores. The dynamics of the search process, as captured by the trajectory of solutions in the objective space, is reported in Figure 5.4. Our results reveal that the variant based on weighted sum obtains better results than the one based on hypervolume. We attribute this to their underlying contour lines, something we later investigated for static scalarizing functions (Derbel et al., 2014a); see Section 5.5. In addition, we report that both variants perform significantly better than a distributed approach that uses a static set of uniformly-defined weight vectors. This suggests that the local information exchanged in our cooperative strategies is relevant for adapting the weight vectors, and leads to an effective and scalable decomposition-based parallel search.

In Figure 5.5, we report the parallel performance of DLBS using a message passing parallel implementation. Given that only the position

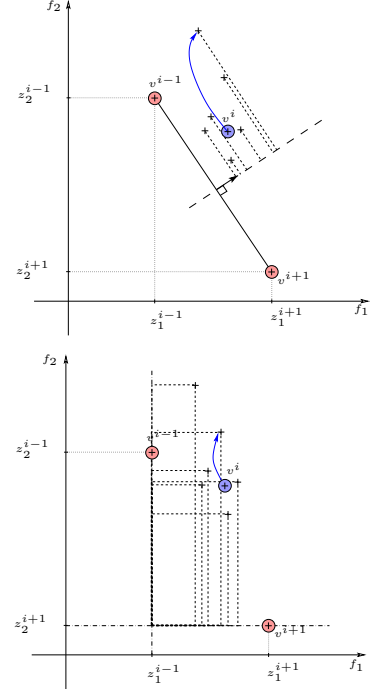


Figure 5.3: Illustration of the selection mechanism using the localized fitness function f_ℓ^{od} (top) and f_ℓ^{hv} (bottom). The crosses without circle correspond to candidate solutions, and the arrow shows the selected candidate solution which is to replace the current one.

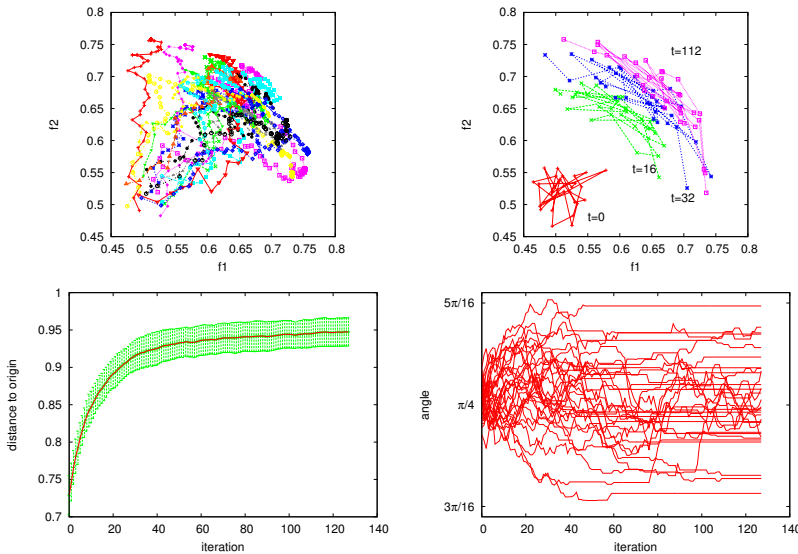
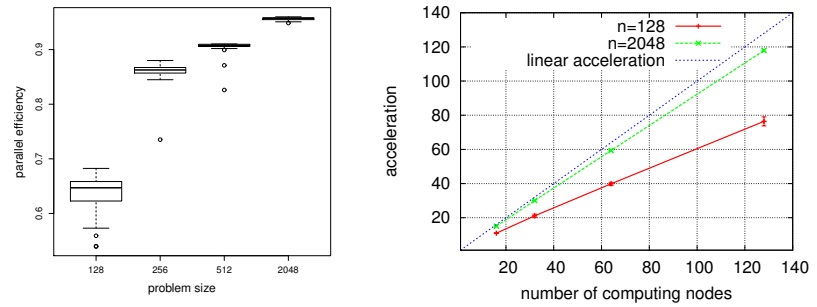


Figure 5.4: Dynamics of DLBS with f_ℓ^{od} on an exemplary instance: evolution of the nodes trajectory (top-left), evolution of the neighborhood graph (top-right), evolution of the distance between the node positions and the origin in the objective space (bottom-left), and evolution of nodes polar angle (bottom-right).

Figure 5.5: Parallel efficiency and scalability of DLBS: ratio of computational time over execution time with respect to the problem size (left), and acceleration ratio of DLBS with respect to the number of processing units (right).



of solutions in the objective space is communicated among neighboring nodes, the size of messages used to effectively implement DLBS does not depend on the solution encoding, nor on the variable space dimension. This is an advantageous feature of our approach to scale particularly efficiently and to obtain a substantial speed-up depending on the problem size, which is tightly related to the cost of the evaluation function. The parallel efficiency, that is the computation vs. communication cost, is around 90% for instances of size $n = 512$, and reaches more than 95% for large instances with $n = 2048$. We also obtain a linear acceleration factor when comparing our message passing implementation to a sequential implementation of DLBS using a single processing unit. From a parallel computation perspective, this suggests that DLBS is able to scale efficiently, both as a function of the problem dimension and of computational resources.

A Parallel Message-Passing Decomposition Approach

In contrast with the previous approach, the selection and replacement steps of MOEA/D are performed using solutions from neighboring sub-problems. This creates a dependency when evolving the population in parallel. Although it would offer an obvious parallelism, breaking this dependency by optimizing each sub-problem independently in one processing unit is not accurate for multi-objective search. Consequently, some works investigate the intuitive idea that non-overlapping sub-problems could be processed in parallel (Durillo et al., 2011; Nebro and Durillo, 2010). In particular, Nebro and Durillo (2010) show that good speed-ups can be obtained at the price of significantly deteriorating the approximation quality of the sequential version of MOEA/D. Mambrini and Izzo (2014) propose a parallel variant of MOEA/D based on the island model (Candan et al., 2013). Every island evolves a sub-population of solutions corresponding to a subset of sub-problems, and selected solutions are communicated to other islands during a migration phase. However, their approach relies on a specific thread-based implementation, and is experimented on an 8-core shared-memory computer. As such, its scalability is questionable, given the increase of communications and of shared-memory read and write operations. Our work departs from previous studies in several aspects, but also retain insightful lessons learnt from those. Following Nebro and Durillo (2010), we believe that handling overlapping neighboring sub-problems is a key issue to achieve a high

scalability and approximation quality. Moreover, as in Mambrini and Izzo (2014), we get inspiration from the island model, although we do not explicitly rely on the conventional concept of migration, thus finely optimizing our approach when facing a fine-grained parallelism.

More particularly, we propose a parallel message-passing version of MOEA/D (MP-MOEA/D), which is summarized in Algorithm 3 (Derbel et al., 2015). In fact, MP-MOEA/D is based on a (sequential) generational variant of MOEA/D, as detailed in a prior study (Marquet et al., 2014). The algorithm is to be executed in parallel by *every* processing unit; i.e. the memory is local and not shared in any way. We consider a one-to-one mapping between sub-problems and processing units, while assuming seemingly the standard neighborhood relation among sub-problems. We notice this is a particularly harsh assumption, but this allows us to fairly analyze the scalability of our scheme with a fine-grained level of parallelism. The parallel process is divided into two stages. The first one is performed locally without any communication, whereas the second one requires sub-problems, and thus processing units, to communicate with their neighbors. Each processing unit aims at identifying an improving solution for its own sub-problem, while also sending improving solutions to neighboring sub-problems and receiving improving solutions from neighboring sub-problems. As such, each processing unit maintains a *local copy* of the solution from its neighbors. Based on this, it performs the selection and variation mechanisms from vanilla MOEA/D with essentially three main modifications: (i) the current solution of the considered sub-problem is always selected for reproduction, as in the selfish selection from Marquet et al. (2014), (ii) offspring solutions are generated for t_{\max} iterations, and (iii) since the current remote solutions from neighbors are not available locally, the algorithm simply checks whether each newly-generated offspring improves over any of the local copies maintained for every neighbor. Improving offspring solutions are temporarily saved locally in order to be sent to neighbors at the communication stage, as explained below, with the aim of updating and synchronizing solutions within the population.

The second stage deals with the update of local copies by means of communication. In case an improving solution with respect to the processing unit's own sub-problem is identified, the processing unit notifies its neighbors so that they can update their local copies with the improving offspring. In case an improving solution for one or more neighboring sub-problems is identified, the processing unit also notifies the corresponding neighbors so that they can update their own solution with the improving offspring. Symmetrically, a processing unit checks whether these situations occur by receiving information from its neighbors before resuming a new round of local computations. This communication phase needs to be handled with care, since this fine-grained parallelism can prevent scalability. Indeed, on one hand, the fastest the solutions of a processing unit are updated with the latest information from its neighbors, the better the improvement in terms of approximation quality. On the other hand, synchronizing processing units often implies a non-negligible communication cost. This is precisely the reason for introducing the t_{\max} parameter, which allows

Algorithm 3: MP-MOEA/D — High-level algorithm to be executed by every computing node $i \in \{1, \dots, \mu\}$.

```

1 initialize  $x^j$  for all  $j \in B(i)$ ;
2 repeat
3   flag  $\leftarrow 0$ ;
4   // stage #1: local computations
5   for  $j \in B(i) \setminus \{i\}$  do
6      $y^j \leftarrow x^j$ ;
7   repeat  $t_{\max}$  times:
8     // selection and variation
9      $\ell \leftarrow \text{rand}(B(i) \setminus \{i\})$ ;
10     $y \leftarrow \text{variation}(x^i, x^\ell)$ ;
11    // local replacement
12    if  $g(y, \lambda^i)$  better than  $g(x^i, \lambda^i)$  then
13       $x^i \leftarrow y$ ; flag  $\leftarrow 1$ ;
14    // check for neighbors' improvements
15    for  $j \in B(i) \setminus \{i\}$  do
16      if  $g(y, \lambda^j)$  better than  $g(y^j, \lambda^j)$  then
17         $y^j \leftarrow y$ 
18    // stage #2: communication
19    communicate_local_copies (flag);
20 until a stopping condition is satisfied;

```

us to control the relative cost of local computations by setting the frequency of communication. Moreover, we consider both a synchronous and an asynchronous message passing implementation, where the information of local copies is aggregated into a single message in order to reduce the number of transmissions over the network. In the synchronous MP-MOEA/D, each processing unit sends a message with its current information to its neighbors, and then waits to receive the message from its neighbors. This ensures that the most up-to-date information is used by all processing units, although an acknowledgment has to be issued even when there is no improvement, in order to avoid deadlocks. By contrast, the asynchronous MP-MOEA/D reduces the idle time by removing the remote synchronization cost. However, we lose the guarantee that local copies are up-to-date. This might thus lead to the situation where the search process is resumed for several rounds with outdated information, which constitutes a penalty in terms of approximation quality.

We deploy a message passing implementation of MP-MOEA/D on Grid'5000⁷ using 128 computing cores. Our experimental results reveal that MP-MOEA/D achieves a non-trivial trade-off in terms of approximation quality and parallel efficiency, as reported in Figure 5.6. Acceleration is here measured as the ratio between the runtime of the sequential MOEA/D and the parallel runtime of MP-MOEA/D. Each point corresponds to one update frequency parameter setting t_{\max} , and the horizontal line corresponds to the approximation quality obtained by the sequential MOEA/D. As such, the points below this line indicate that MP-MOEA/D outperforms MOEA/D in terms of approximation quality, and the points further on the right indicate a

7: The French national experimental grid:
<https://www.grid5000.fr/>.

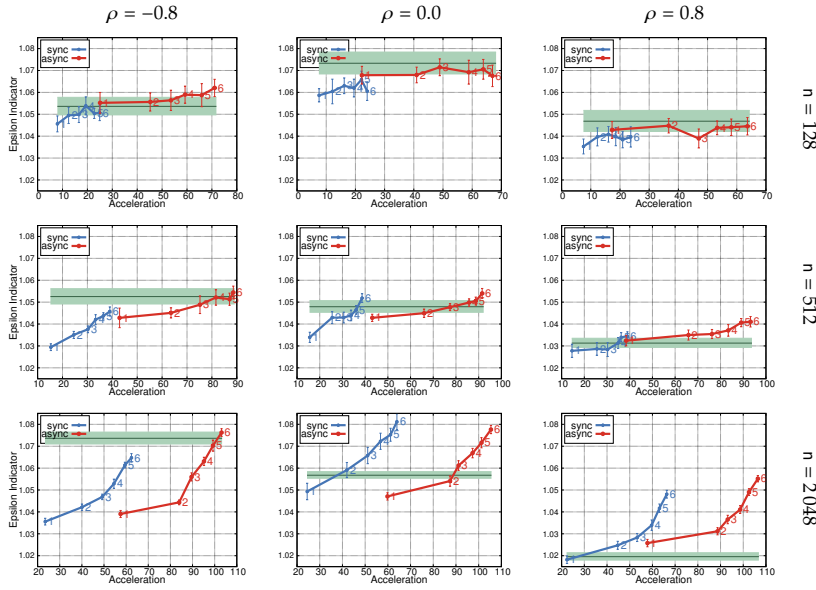


Figure 5.6: Acceleration vs. approximation quality with respect to the update frequency t_{\max} for different ρ_{mnk} -landscapes with $k = 4$ and $m = 2$.

good parallel efficiency. Overall, the approximation quality seems to drop with higher t_{\max} values, while acceleration gets better. Interestingly, there are communication frequencies for which approximation quality is very competitive compared against the sequential MOEA/D, while achieving a substantial acceleration. For smaller instances, where the evaluation cost is lower, the impact of t_{\max} on quality is less pronounced, and larger values offer a significant acceleration without a substantial drop in terms of quality. For larger instances, we observe a similar trend for conflicting objectives, whereas it seems harder to obtain a high acceleration without a significant drop in quality for correlated objectives. We attribute this to the probability that improving solutions for one sub-problem is more likely to improve neighboring sub-problems as the objective correlation increases, such that communicating right away becomes more critical. This relates to the impact of replacement in MOEA/D, as analyzed in [Marquet et al. \(2014\)](#). This is less likely to happen with conflicting objectives, where the Pareto front is larger, and where the underlying diversity among sub-problems balances this side-effect. Finally, we remark that the gap between the synchronous and asynchronous MP-MOEA/D increases in favor of the asynchronous implementation as the problem size or the degree of conflict among the objectives increases. Interestingly, for large-size instances with many non-dominated solutions, the asynchronous MP-MOEA/D obtains a close to linear acceleration, while being as good as MOEA/D in terms of approximation quality.

A Parallel Pareto Local Search based on Decomposition

As presented in Chapter 2, Pareto Local Search (PLS) is a multi-objective local search based on dominance, that maintains an unbounded archive of non-dominated solutions found so far. At each iteration, it selects a solution from the archive, explores its neighborhood, and updates the archive with neighboring solutions. This process is iterated until the neighborhood of all solutions from the archive has been examined,

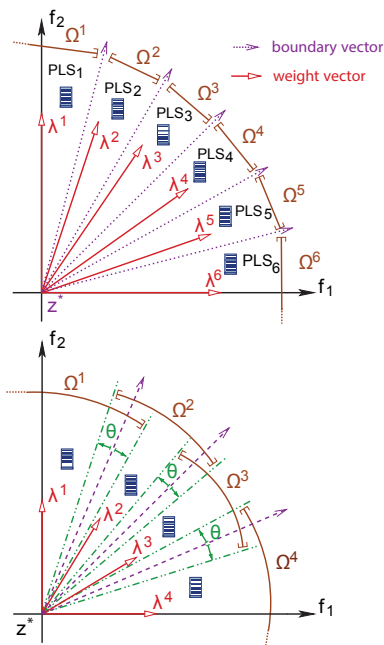


Figure 5.7: Illustration of search regions decomposition in Parallel PLS. On top, six processes are executed independently in parallel. At the bottom, four processes are executed independently in parallel, with overlapping sub-regions based on an opening angle θ .

and PLS naturally stops on a Pareto local optimal set (Paquete et al., 2007), as analyzed in Chapter 4. Although the basic PLS algorithm is able to achieve a high-quality approximation, it is well known that its convergence is slow, and several strategies have been proposed to overcome this issue; see, e.g., Drugan and Thierens (2012), Dubois-Lacoste et al. (2015), and Geiger (2011). PLS has three main components that are crucially important for its performance (Liefvooghe et al., 2012): (i) the selection of the solution to choose next from the archive, (ii) the neighborhood exploration strategy, and (iii) the update of the archive with respect to newly-explored neighboring solutions. We remark that, with the exception of the neighborhood exploration, the two other components require the full knowledge of the archive, which makes it rather challenging to derive a high-level parallel version of PLS. In Shi et al. (2018, 2017b), we get inspiration from decomposition-based techniques in order to design a parallel PLS for bi-objective optimization. It extends a preliminary work on parallel local search for single-objective optimization (Shi et al., 2017a).

More particularly, we decompose the objective space evenly into multiple regions based on a number of weight vectors, similar to Liu et al. (2014). This consists in delimiting a sub-region of the objective space by means of a reference point and two consecutive lines, as illustrated in Figure 5.7. Based on this, multiple PLS processes are executed in parallel, each one operating in one of the sub-regions. In case a solution is located outside the boundaries, it is simply ignored, unless there is currently no solution inside the boundaries in the archive. In addition, the selection and update phases are revised accordingly. In particular, as illustrated in Figure 5.7, each parallel process is assigned a weight vector corresponding to the region where it operates, and uses a weighted sum in order to select solutions from the archive and to guide the neighborhood exploration. This does not only allow us to coordinate the parallel processes locally by using different weight vectors, but also to drastically reduce the archive size. In Shi et al. (2017b), we experiment such a parallel PLS on a number of bi-objective unconstrained binary quadratic programming instances, for which the sequential PLS is known as the state of the art (Liefvooghe et al., 2015b). We report that the parallel PLS significantly accelerates the sequential PLS while obtaining a similar level of approximation quality.

In Shi et al. (2018), we further refine the parallel PLS approach by addressing the load imbalance and scalability issues caused by the definition of sub-regions. In particular, we improve the setting of the reference point for unknown Pareto front shapes, and we slightly enlarge the partitions obtained by decomposition by a small factor θ , hence allowing two neighboring regions to overlap, as illustrated in Figure 5.7 (bottom). The trajectory of solutions from the parallel PLS is illustrated in Figure 5.8 for exemplary executions, which provides visual evidence on the accuracy of decomposition. In Figure 5.9, we report the trade-offs obtained by parallel PLS for different settings of θ . Our results suggest that, when scaling the number of parallel processes, there exists an optimal setting of the opening angle θ such that parallel PLS performs at best, in terms of anytime behavior. This shows that bringing decomposition into PLS is beneficial to improve convergence,

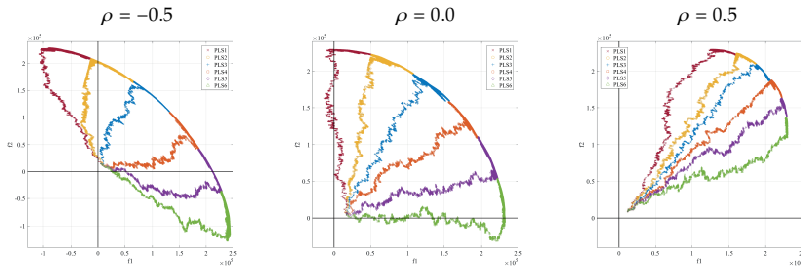


Figure 5.8: Parallel PLS trajectory in the objective space on bi-objective unconstrained binary quadratic programming instances with $n = 100$ variables, and a varying objective correlation (ρ).

while enabling a very efficient high-level parallel design that was unexplored until now. This opens the door to further investigations on improving the anytime behavior of PLS and on deploying PLS into large-scale parallel environments.

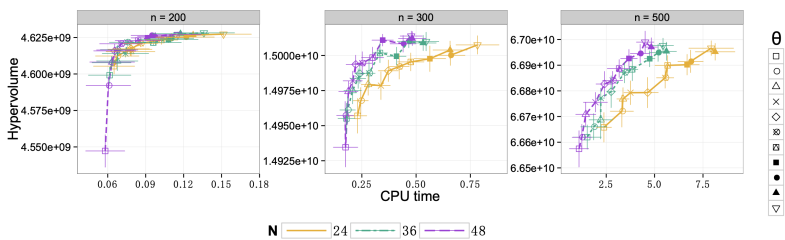


Figure 5.9: Runtime vs. approximation quality of parallel PLS with respect to the opening angle θ for different number of parallel processes (N) on bi-objective unconstrained binary quadratic programming instances with no objective correlation, and a varying number of variables (n).

5.4 Surrogate-assisted Multi-objective Search

In this section, we are interested in the use of *meta-models* to speed up the search process, and thus improve algorithm performance when the search budget is particularly tight. More precisely, we consider *expensive* multi-objective optimization problems, for which evaluating the quality of each and every solution is particularly demanding in terms of computational time. This is typically the case in different application domains that require heavy and costly simulation efforts. In this context, one has to deal with the multi-objective nature of the problem by identifying a high-quality approximation set, while minimizing as much as possible the computational effort in terms of calls to the expensive evaluation function. *Surrogate-assisted* search approaches have received intensive research in this regard over the last decade; see, e.g., Bartz-Beielstein and Zaefferer (2017) and Jin (2011).

Surrogate-assisted techniques rely on a meta-model as a substitute for the expensive function. The meta-model is used to efficiently sample or pre-screen candidate solutions, from which a subset is then evaluated for real, using the real evaluation function. Leveraging surrogates for multi-objective optimization is facing difficult challenges from different perspectives (Chugh et al., 2019), such as the nature of the meta-model being used as a surrogate, or the multi-objective selection paradigm being used at the core of the underlying search process. Interestingly, there exist a number of well-established meta-models for continuous functions, so that the main effort to deal with expensive continuous multi-objective optimization is on articulating the meta-model with the multi-objective search process. Unfortunately, such an issue has not been addressed in the context of combinatorial

optimization. Of particular interest is the design of meta-models for discrete functions, that would make it possible to deal with expensive combinatorial multi-objective optimization problems. This constitutes the two complementary goals that we address below: (i) a taxonomy and comparative study of surrogate-assisted algorithms for continuous multi-objective optimization, and (ii) the design of a surrogate-assisted approach for combinatorial multi-objective optimization.

A Comparative Analysis of Surrogate-assisted Multi-objective Evolutionary Algorithms

A substantial number of surrogate-assisted search approaches have been developed to solve expensive problems from *continuous* multi-objective optimization. The extent of techniques and their combination makes it particularly difficult to assess the effectiveness, implication, drawback and robustness of their internal components, and therefore makes it a tedious task to make a thorough design choice. In [Berveglieri et al. \(2019, 2022\)](#), we propose a refined and fine-grained classification covering existing approaches from the literature, ranging from decomposition- to dominance- and indicator-based selection. The resulting taxonomy of surrogate-assisted multi-objective search algorithms is presented in Figure 5.10. The search process is divided into five components discussed below:

- I. Initialization.** As with any search heuristic, the *initialization* phase aims at generating the first batch of solutions, with the difference that these solutions will not only be used by the search process, but they will also take part in the construction of the training set for fitting the considered meta-model(s). In order to cover a large part of the variable space, and thus expect to improve the model accuracy, a latin hypercube sampling ([Carnell, 2020](#)) of the variable space is typically used.
- II. Model training set.** We thus enter the outer loop of the algorithm with the construction of the *training set*. This step is usually performed at each iteration. The aim here is thus to find a good trade-off between fitting accuracy and training time. Rather than selecting the whole set of solutions evaluated so far, existing strategies tend to opt for (i) either a single meta-model trained with a smaller training set using subset selection, or (ii) a combination of target-specific meta-models constructed by means of clustering. In the first case, this ranges from a random subset, to the most recent solutions, or even the best solutions according to the selection paradigm under consideration. In the second case, the idea is to build a different meta-model per cluster of solutions, the clustering being applied in the variable space, in the objective space, or even in the weight space for decomposition-based search.
- III. Model fitting and response.** On the basis of the training set constructed at the previous step, one or multiple meta-models are trained, sometimes even ensemble meta-models that could be trained in parallel ([Berveglieri et al., 2020](#)). In addition to selecting a statistical or machine learning model type and its

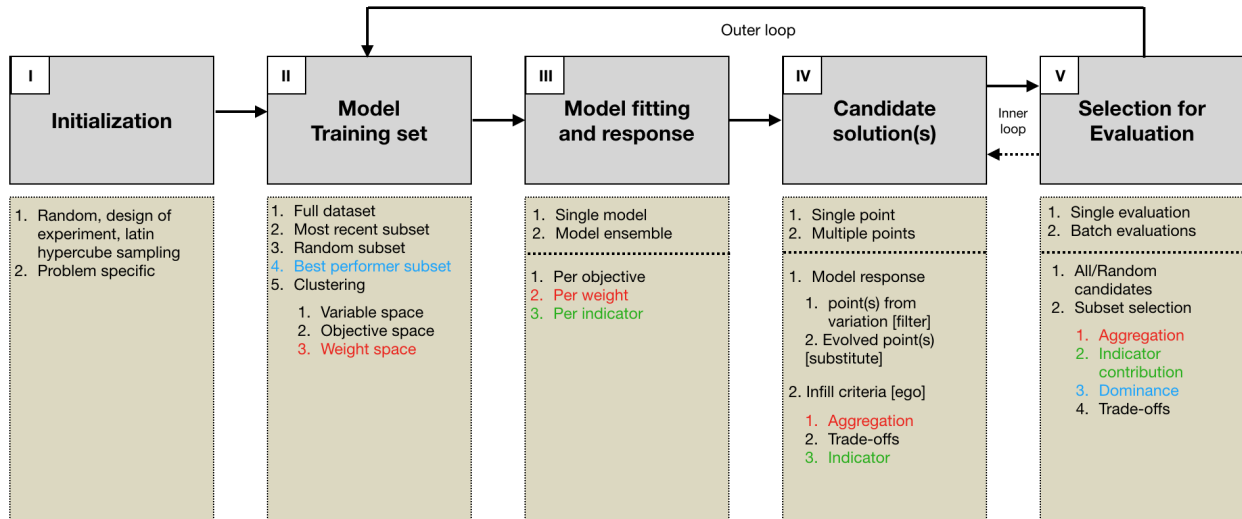


Figure 5.10: A taxonomy of surrogate-assisted multi-objective search algorithms. In blue are strategies tailored to dominance-based selection, in red to decomposition-based selection, and in green to indicator-based selection.

parameters, one must also decide on the *model response*. A usual approach is to predict the objective values; that is, to construct a meta-model independently for each objective function. However, some alternatives depend on the peculiarities of the considered selection paradigm. For instance, some indicator-based approaches choose to predict the indicator contributions, while some decomposition-based approaches choose to fit a meta-model per sub-problem's scalarizing function.

VI. Candidate solution(s). Based on the trained meta-models, several options open up in order to generate candidate solutions. Perhaps the simplest one, denoted as *filter*, consists in using the meta-models to pre-screen new candidate solutions. These solutions are, for instance, generated by random variation. Based on their predicted values, only a subset of solutions will undergo an expensive evaluation in the next step. Alternatively, *substitute* approaches temporarily rely on the meta-models in order to evolve solutions that are predicted to be of higher quality. The idea is to use an internal multi-objective search procedure as an inner optimizer of the meta-models in order to identify improving solutions according to the surrogates, all this without using the real evaluation function. Thus, unlike *filter*, several rounds of candidate solution generations are performed before resorting to the expensive objectives. At last, bayesian and *efficient global optimization* (EGO) approaches are tailored to gaussian processes-based meta-models, whose response is used to build an acquisition function, such as expected improvement or probability of improvement (Bartz-Beielstein and Zaefferer, 2017; Jones et al., 1998). This allows the algorithm, not to optimize the objective function directly, but rather the acquisition function, thus balancing the exploration-vs-exploitation trade-off when selecting which solution to sample next. In the multi-objective case, the EGO meta-models are either based on indicator contributions or on scalarizing functions.

V. Selection for Evaluation. Based on the pool of solutions generated at the previous step, one or multiple solutions are to be selected to undergo the true evaluation function. Once again, the meta-models are used to rank candidate solutions based on the algorithm's internal selection process, being based on dominance, decomposition, or indicator. Selecting multiple solutions at this stage is particularly relevant for processing batch evaluation in parallel (Berveglieri et al., 2020). Selected solutions are evaluated using the expensive objectives, and the algorithm iterates to Step II. Newly evaluated solutions can thus take part in the pool of solutions from which the training set is built at the next round.

In light of the taxonomy discussed above, we can not only instantiate existing algorithms from the literature, but also design new ones by simply selecting a different strategy for each component; see Berveglieri et al. (2019, 2022) for a more detailed discussion.

We further provide a comprehensive comparative analysis of selected approaches and their components under a common ground, each one being represented by state-of-the-art and/or baseline cleaned-up surrogate-assisted algorithms. State-of-the-art algorithms include ParEGO (Knowles, 2006), SMS-EGO (Ponweiser et al., 2008), MOEA/D-EGO (Zhang et al., 2010), MOEA/D-RBF (Zapotecas-Martínez and Coello Coello, 2013), M-EGO (Hussein and Deb, 2016) and K-RVEA (Chugh et al., 2018), among others. This gives us the chance to improve our understanding of surrogate approaches as well as their impact on the different classes of multi-objective search paradigms. Besides comparing and evaluating the influence of different strategies for constructing the training set (II), for model fitting and response (III), for the generation of candidate solutions (IV), as well as their selection for an expensive evaluation (V), our experiments reveal that the best-performing approach strongly depends on the budget allocated to the search process in terms of calls to the expensive evaluation function. From our analysis, the most impactful component appears to be the way new candidate solutions are generated by means of the meta-models. This is illustrated in Figure 5.11 for a subset of algorithms on selected problems from the *bbob-biobj* bi-objective black-box continuous benchmark functions test suite (Brockhoff et al., 2019). Three selection paradigms are considered: MOEA/D for decomposition-based search, SMS-EMOA for indicator-based search, and NSGA-II for dominance-based search. They are combined with three surrogate strategies for the generation of candidate solutions (IV): filter, substitute and EGO — by design there is no obvious way to configure an EGO-like NSGA-II. Filter and substitute approaches are based on support vector regression, whereas EGO approaches are based on Gaussian processes. Although our results suggest that decomposition-based approaches often lead to better approximations than other multi-objective selection paradigms, the gain is significantly less substantial than using a suitable surrogate strategy. Indeed, EGO approaches such as SMS-EGO (Ponweiser et al., 2008) and MOEA/D-EGO (Zhang et al., 2010) quickly converge to a reasonable approximation quality, but do not seem to be able to improve after reaching a plateau. By contrast,

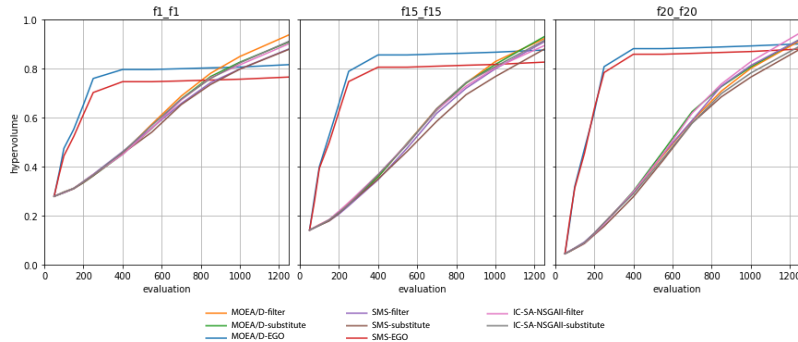


Figure 5.11: Performance of surrogate-assisted algorithms for selected problems from the bbo-biobj test suite with 20 variables: double Sphere (left), double Rastrigin (middle), and double Schwefel (right) functions.

filter and substitute approaches converge less abruptly, but end up outperforming the former when the budget gets large enough. There does not seem to be a significant difference between filter and substitute approaches, regardless of the selection paradigm. Subsequently, these findings allowed us to develop a simple adaptive approach to switch from one strategy to another at the right time (Berveglieri et al., 2022). As illustrated in Figure 5.12, the proposed approach improves the anytime behavior of surrogate-assisted multi-objective search and outperforms approaches from the literature on the considered problems under most optimization scenarios in terms of search budget.

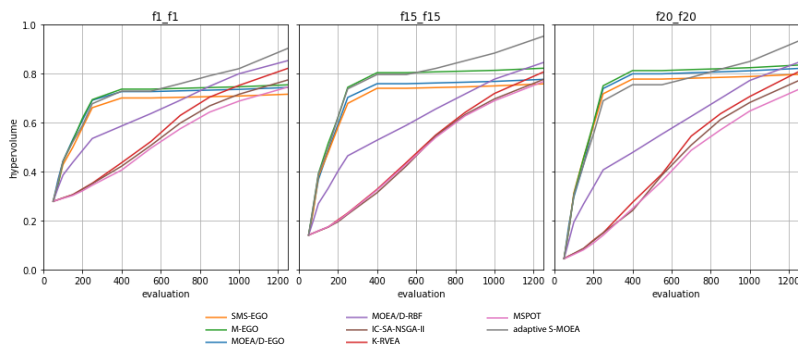


Figure 5.12: Adaptive vs. state-of-the-art surrogate-assisted algorithms for selected problems from the bbo-biobj test suite with 20 variables: double Sphere (left), double Rastrigin (middle), and double Schwefel (right) functions.

A Surrogate-assisted Approach for Multi-objective Combinatorial Optimization

Let us now take a step towards the development of a surrogate-assisted methodology for expensive optimization problems having both a multi-objective and a *combinatorial* nature. The first challenge concerns the meta-model to be used for discrete functions. Looking at the specialized literature, one can find a handful of recent studies on surrogates for single-objective black-box combinatorial optimization (Bartz-Beielstein and Zaefferer, 2017). In fact, there are three general-purpose meta-models for discrete functions, namely bayesian models, radial basis functions, and kriging (Baptista and Poloczek, 2018; Moraglio and Kattan, 2011; Zaefferer et al., 2014). All of them are adaptations of their well-established counterparts from the continuous case. By contrast, in Verel et al. (2018), we rely on the mathematical foundations of discrete Walsh functions to derive a meta-model for discrete problems. Unlike existing distance- and similarity-based discrete surrogates, the

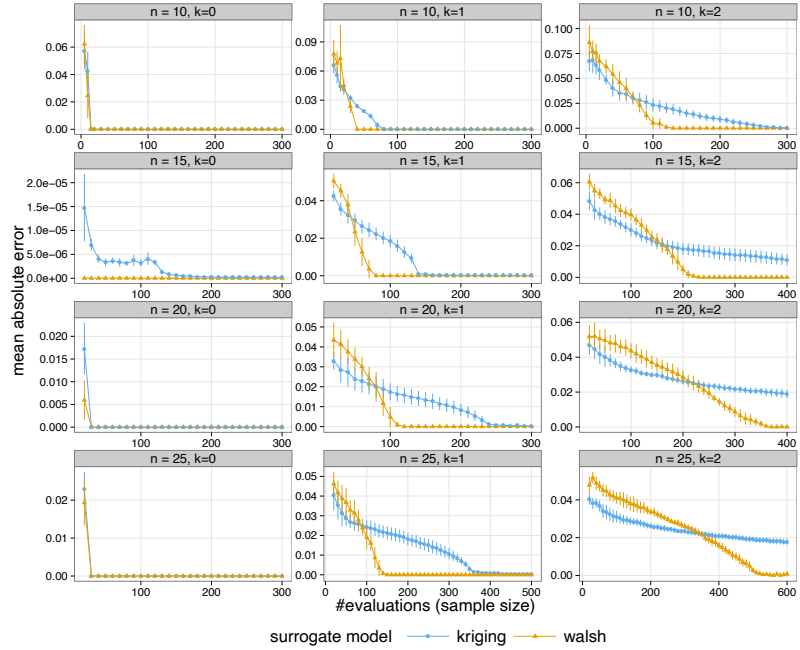


Figure 5.13: Mean absolute error of function values on a test set of 1000 solutions with respect to the size of the training set for (single-objective) nk -landscapes with $n \in \{10, 15, 20, 25\}$ variables (by rows), $k \in \{0, 1, 2\}$ variable interactions (by columns), and an order of $k + 1$ for the Walsh expansion.

Walsh model is based on a deterministic approximation. More specifically, we consider pseudo-boolean functions for dealing with combinatorial optimization problems with binary variables. Walsh functions form a complete orthogonal set of functions, and can be considered as a discrete counterpart of the trigonometric Fourier series. We model our surrogate by means of a Walsh expansion, and we propose to represent any pseudo-boolean function as a discrete Walsh decomposition. The model coefficients can thus be approximated by means of sparse linear regression. Figure 5.13 reports experimental results comparing the prediction accuracy of the proposed Walsh surrogate against discrete kriging on a comprehensive set of (single-objective) nk -landscapes. We observe that the Walsh meta-model provides a highly accurate approximation, substantially outperforming kriging on non-linear instances. On top of that, training the Walsh surrogate turns out to be extremely fast compared against kriging.

In Pruvost et al. (2020b, 2021), we hence focus our investigations on such an optimization domain, and we consider to leverage Walsh surrogates to deal with multi-objective pseudo-boolean problems. To the best of our knowledge, this is the first proposal in this line, aiming at dealing with expensive multi-objective combinatorial optimization problems. The proposed decomposition-based modular framework is presented in Algorithm 4. At each iteration, a Walsh meta-model is trained independently for each objective. A set of candidate solutions is thus generated on the basis of the Walsh surrogates, considered as substitutes of the real objectives, following the substitute approach discussed above. Among those, one solution is finally selected in order to be evaluated using the real, expensive objectives. The framework integrates three main configurable components: (i) the inner optimizer used for producing promising candidate solutions by (temporarily) relying on the previously-trained Walsh surrogates, (ii) the selection strategy to decide which solution is to be evaluated by the expensive

Algorithm 4: S-MCO — Surrogate-assisted framework for multi-objective combinatorial optimization.

```

1  $P \leftarrow \{x^1, \dots, x^\mu\}$  : initial population of size  $\mu$ ;
2  $D \leftarrow \{(x^1, f(x^1)), \dots, (x^\mu, f(x^\mu))\}$ : training data;
3  $A \leftarrow$  initialize external archive (optional);
4  $z^* \leftarrow$  initialize reference point;
5 while global budget is not exhausted do
6   for  $i \in \{1, \dots, \mu\}$  do
7     // choose Walsh order
8      $o \leftarrow$  choose_walsh_order (history,  $d$ );
9     // train Walsh models
10     $\tilde{f} := (\tilde{f}_1, \dots, \tilde{f}_m) \leftarrow$  train_walsh_models ( $D, o$ );
11    // copy reference point for optimizer and selection
12     $z^{**} \leftarrow z^*$ ;
13    // run optimizer for surrogate models
14     $S \leftarrow$  optimizer ( $P, \tilde{f}, z^{**}$ );
15    // select solution for true evaluation
16     $x' \leftarrow$  select_for_evaluation ( $S, \lambda^i, \tilde{f}, z^{**}$ );
17     $f(x') \leftarrow$  evaluate ( $x'$ );
18     $A \leftarrow$  update external archive using  $x'$  (optional);
19     $z^* \leftarrow$  update reference point using  $f(x')$ ;
20    // replacement in the population
21    for  $j \in \{1, \dots, \mu\}$  do
22      if  $g(x', \lambda^j)$  better than  $g(x^j, \lambda^j)$  then
23         $x^j \leftarrow x'$ ;
24    // update training data
25     $D \leftarrow D \cup \{(x', f(x'))\}$ ;

```

objectives, and (iii) the strategy used to setup the order of the Walsh expansion, that is, the hyper-parameter of Walsh meta-models. Based on a thorough empirical analysis on bi-objective ρ mnk-landscapes and unconstrained binary quadratic programming problems, we show that local search provides an accurate inner optimizer for Walsh surrogates, while a selection strategy based on the predicted improvement of candidate solutions with respect to decomposition is highly effective. We also highlight the importance of using a proper Walsh order by means of a simple dynamic strategy. Additionally, our experiments reveal the effectiveness of the proposed approach with respect to the available budget in terms of calls to the evaluation function. This is illustrated in Figure 5.14 on exemplary instances, by comparing the surrogate-assisted method against a surrogate-less variant of three multi-objective search strategies, namely MOEA/D, multi-start local search (MLS) and Pareto local search (PLS). We clearly see that surrogate-assisted approaches obtain substantially better approximations, independently of the considered optimizer, instance and budget. More importantly, our empirical findings shed more lights on the combined effects of the investigated components on search performance, thus providing a better understanding of the key challenges for designing a successful surrogate-assisted search process for expensive multi-objective combinatorial optimization.

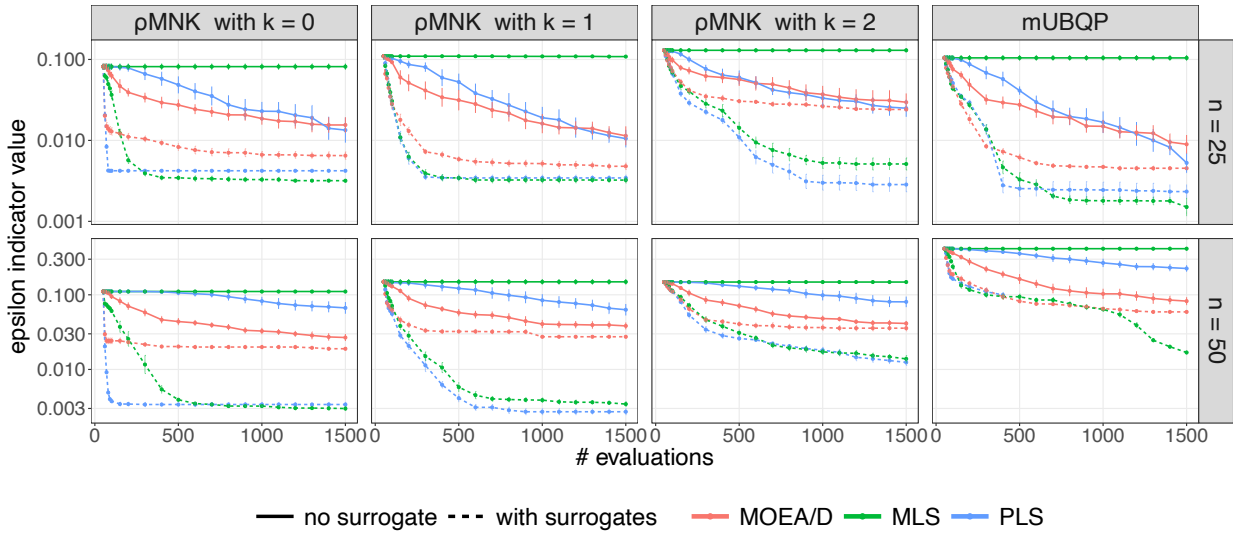


Figure 5.14: Surrogate-less vs. surrogate-assisted approaches based on the Walsh meta-model for two-objective ρmnk -landscapes with a varying number of variables (n) and variable interactions (k), and for bi-objective unconstrained binary quadratic programming (UBQP) instances with a varying number of variables (n).

5.5 A Glimpse on Further Contributions

We give below a brief overview of contributions related to other algorithm components which aim at improving the anytime behavior of multi-objective search. In particular, we study components related to population, selection and variation, with a particular emphasis on decomposition-based approaches.

Population

A critical aspect of any evolutionary algorithm and multi-objective search approach deals with the configuration of its population, that is the set of solutions maintained and evolved by the algorithm and ultimately returned as the resulting approximation set. Setting the population size is a well-known issue in evolutionary computation to prevent from premature convergence (Leung et al., 1997). On top of that, the population size in multi-objective optimization directly influences the cardinality, and therefore the quality of the resulting approximation set (Coello Coello et al., 2007; Deb, 2001). In Aguirre et al. (2013), we investigate the impact of the population size on the dynamics of state-of-the-art multi-objective evolutionary algorithms. Our experiments reveal that Pareto optimal solutions might be discarded from the population in favor of sub-optimal solutions that are temporarily non-dominated. This selection lapse also affects the discovery of new Pareto optimal solutions. We show that selection makes fewer mistakes with larger populations, and when the distribution of solutions is better controlled by the search process. As such, we argue that not only the size of the population, but also its structure strongly influences the search process. Compared against other algorithm classes, this is something that can be handled more explicitly in decomposition-based search, since each individual maps to a given

scalar sub-problem. For instance, in [Derbel et al. \(2018\)](#), we propose to re-design the working principles of MOEA/D by adopting a many-to-one mapping between sub-problems and solutions, thus allowing the structured population to have a varying, but bounded size. This leads to a significant improvement in the approximation quality, especially when dealing with a large number of objectives.

More recently, we investigated the impact of the population in decomposition-based search more thoroughly ([Pruvost et al., 2020a](#)). Firstly, our results confirm that a larger population tends to be better when a sufficient budget is allocated to the search process for the algorithm to converge, especially when there are many objectives to optimize. Not only this, we also carefully considered both the number of solutions that survive, and the number of new candidate solutions generated, at each iteration. In its default setting, both parameters are the same in vanilla MOEA/D — roughly speaking one solution is maintained and generated per-sub problem at each iteration. As such, we revise the design of MOEA/D by explicitly dissociating three components: (i) the number of solutions maintained in the population (μ), (ii) the number of new candidate solutions generated at each iteration (λ) — generally denoted as *offspring* in evolutionary computation — and (iii) the strategy adopted to select the solutions involved in the creation of new candidate solutions — the *parents*. In fact, in decomposition-based search, selecting solutions amounts to selecting sub-problems. This is the reason why we denote the latter as *sps*, for sub-problem selection. Although some strategies to distribute the computational effort allocated to sub-problems were integrated in decomposition-based search ([Lavinias et al., 2019](#); [Wang et al., 2019](#); [Zhou and Zhang, 2016](#)), to the best of our knowledge the individual impact of such components were loosely studied in the past. We denote the revised algorithm as MOEA/D-(μ, λ, sps), and we conduct a comprehensive analysis about the impact of those three components on the convergence profile of the search process on a broad range of ρmnk -landscapes. Surprisingly, we find that generating one single candidate solution (i.e., $\lambda = 1$ offspring) per iteration seems to be a reasonable setting for the considered approaches. Notice, however, that additionally selecting the boundary sub-problems was found to be beneficial, as this directly impacts the coordinates of the reference point required by the scalarizing functions ([Wang et al., 2019](#)). Moreover, we analyze whether the sub-problem selection strategy has any impact on search performance. In addition to the conventional MOEA/D setting with $\lambda = \mu$, we consider a state-of-the-art approach when dealing with the distribution of the computational effort over sub-problems, known as dynamical resource allocation (DRA) and proposed by [Zhou and Zhang \(2016\)](#). In DRA, a utility function is defined for each sub-problem relative to its current state in terms of progress over several iterations. We also consider a simple baseline sub-problem selection strategy, which is to select sub-problems uniformly at random. Results are reported in [Figure 5.15](#) for a selection of instances. There are two settings for DRA: $\lambda = 1$, but also $\lambda = \mu/5$ to follow the recommendations from [Zhou and Zhang \(2016\)](#). Interestingly, we observe that a simple random strategy outperforms existing, sophisticated ones from the literature, even

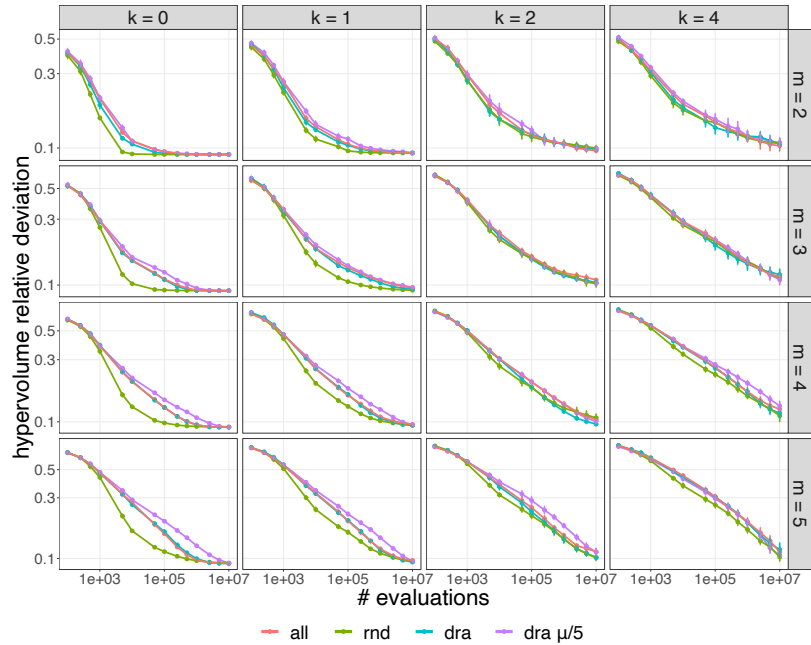


Figure 5.15: Convergence profile of MOEA/D- $(\mu, \lambda, \text{sps})$ with respect to the sub-problem selection strategy: $\mu = 500$, and $\lambda = 500$ for $\text{sps} = \text{all}$, $\lambda \in \{1, \mu/5\}$ for $\text{sps} = \text{dra}$, and $\lambda = 1$ for $\text{sps} = \text{rnd}$.

when properly configured. Our results also suggest that the number of sub-problems selected at each iteration plays a more important role than the way sub-problems are actually selected.

In Figure 5.16, we complement our analysis by studying the sensitivity of this simple strategy with respect to the population size μ . We observe that its anytime behavior is more stable than that of default MOEA/D. We also observe that when using small μ -values, convergence occurs much faster for linear instances compared against non-linear ones. This means that a larger population size μ , combined with a small value of λ , shall be preferred. This observation suggests that, by increasing the number of weight vectors in decomposition, one can obtain a high-level structure of the population, possibly of very large size. Notice also that such a data structure can be maintained very efficiently in terms of computational complexity, given the scalar nature of decomposition. This is to contrast with dominance- and indicator-based search, where maintaining a large population may be computationally intensive, particularly for many-objective problems. Having such an efficient structure, the issue turns out to select the sub-problems from which the population is updated. A random strategy for sub-problem selection with a small λ value is found to work arguably well. This observation was later confirmed for continuous multi-objective optimization as well (Lavinias et al., 2020).

Selection (Scalarizing Function)

Not only the setting of the population, but also the survival selection mechanism has a direct impact on the distribution of solutions in the approximation set identified by multi-objective search. As pointed out in Section 5.2, in the decomposition framework there are different ways of decomposing the original problem by means of a scalarizing function, such as the weighted sum and Chebyshev functions (Trivedi

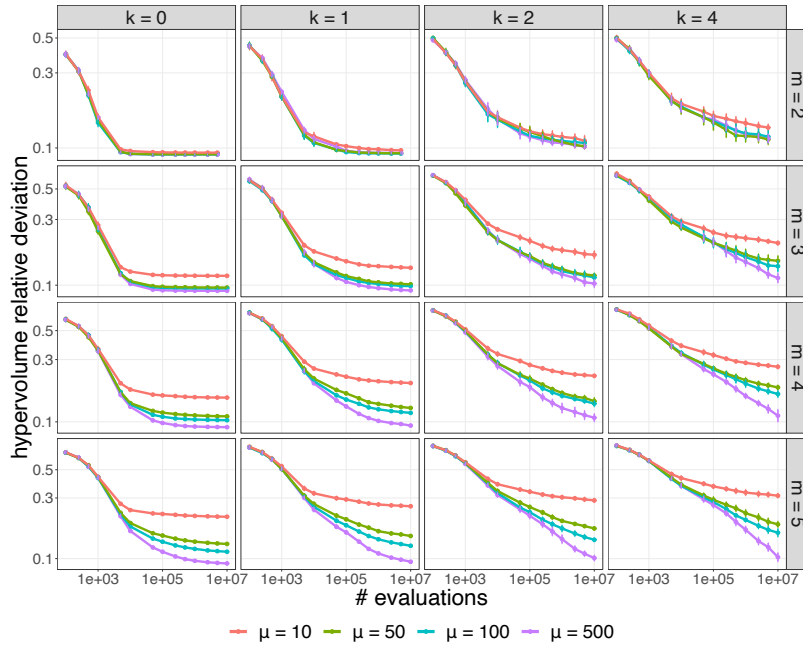


Figure 5.16: Convergence profile of MOEA/D-(μ , $\lambda = 1$, sps = rnd) with respect to the population size (μ).

et al., 2017; Zhang and Li, 2007). These scalarizing functions exhibit different properties with respect to the optimal solution they target, and its position on the Pareto front (Miettinen, 1999). However, their properties in terms of evolutionary and local search are not well understood. In the context of MOEA/D, scalarizing functions were mostly compared with respect to their ability to provide a good approximation set (Ishibuchi et al., 2013; Sato, 2015). In Derbel et al. (2014a), we show that the search dynamics depends on the underlying contour lines induced by the function, rather than the actual choice of a particular scalarizing function or its setting.

We restrict ourselves to the two-objective case, and we define a generic scalarizing function that covers and generalizes existing, commonly-used functions. A crucially important property turns out to be the shape of its contour lines, i.e. the curve in the objective space where all points on the curve share the same scalar value for the considered setting. These lines are given by two straight lines characterized by the opening angles they form with each objective axis, as illustrated in Figure 5.17. We observe that, for a given scalarizing function, the search trajectory in the objective space is a joint effect of the distribution of neighboring solutions obtained by means of variation, and of the gradient direction of the scalarizing function, which is orthogonal to its contour lines. The resulting solution corresponds to the intersection of the ‘highest’ contour line in the gradient direction and the feasible region of the objective space. Interestingly, although the search direction is different, the position of the final solution is similar in the middle and in the right-hand side of of Figure 5.17. Additional results suggest that, independently of the scalarizing function, the final solution is strongly correlated with the opening angles of its contour lines, and not to a particular scalarizing function. As such, the opening angles explicitly drive the search towards a specific region of the objective space.

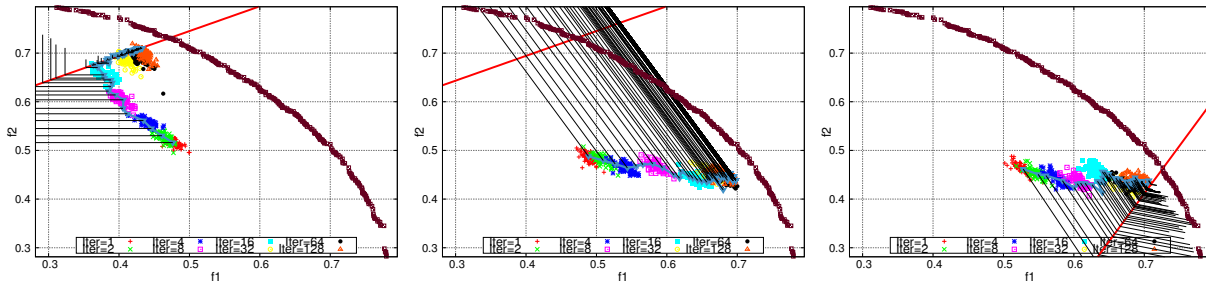


Figure 5.17: Exemplary run of a $(1 + \lambda)$ evolutionary algorithm on a ρ_{mnk} -landscape with an objective correlation $\rho = -0.7$ for different settings of the generalized scalarizing function. Shown are the best-known Pareto front approximation, the solutions generated at different iterations, the search direction and the contour lines for Chebyshev (left), weighted sum (middle), and augmented Chebyshev (right).

While these results considering simple search procedures seem natural and intuitive, they constitute a fundamental step towards a better understanding of the properties and dynamics of decomposition-based search. By raising considerations that were until now hidden by the complex design of well-established algorithms, they certainly make it possible to reconsider some algorithm design choices, in particular with regard to the way in which the initial problem is decomposed into scalar sub-problems.

Variation

As revealed in the previous section, a critical aspect of any search heuristic is the creation of new candidate solutions at each iteration. In a more conventional way than with surrogate models, this is typically done by means of variation operators. Within decomposition-based search in particular, there is in this regard a number of studies aimed both at incorporating widely-used operators and techniques from local search, differential evolution or particle swarm optimization, but also at designing improved variants that benefit from the cooperative process of solving neighboring sub-problems (Trivedi et al., 2017). The important observation here is that the design of intelligent variation operators and their combination is essential in order to solve problems with different characteristics or from different domains. In connection with other algorithm components, they all together aim at effectively improving the population, and thus the quality of the resulting approximation set.

Our contributions in this line first deal with decomposition-based *local* search, and the influence of a number of strategies for the local search moves that are concurrently applied to the different sub-problems on the overall anytime performance (Derbel et al., 2016). We investigate the efficiency of a number of algorithm variants for multi-objective traveling salesperson problems (TSP) with tunable objective correlations. This work can be seen as an alternative approach to decomposition-based local search for the multi-objective TSP proposed by Cornu et al. (2017). Secondly, in Zapotecas-Martínez et al. (2015b), we took advantage of the opportunities offered by the flexibility of the decomposition framework to incorporate the well-established

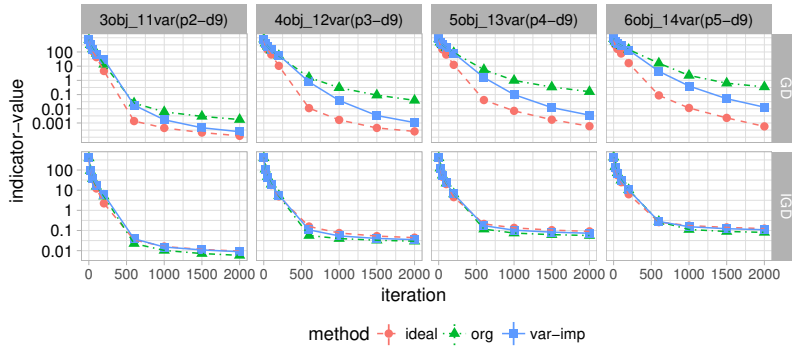


Figure 5.18: Convergence profile of the algorithm with and without machine learning-enhanced variation on DTLZ3 problems with respect to generational distance (top) and inverted generational distance (bottom).

covariance matrix adaptation evolution strategy (CMA-ES), a state-of-the-art method for black-box single-objective continuous optimization (Hansen and Ostermeier, 2001). Besides being able to obtain competitive results compared against the multi-objective indicator-based CMA-ES (Igel et al., 2007), our investigations highlight novel promising alternatives for leveraging existing single-objective CMA-ES ingredients by means of *injection* (Hansen, 2011). Considerations on differential evolution (Das and Suganthan, 2011) for continuous and combinatorial multi-objective search are further presented in Drozdik et al. (2014) and Zapotecas-Martínez et al. (2015a).

An alternative way to CMA-ES for exploiting the information collected so far by the search process in order to adapt and control the generation of new candidate solutions is presented in Sagawa et al. (2016). More particularly, we consider the idea of learning, in an online fashion based on statistical modeling, which problem variables affect the convergence of the population towards the Pareto front. The rationale is that different variables might influence convergence while some might influence diversity — and others might have no influence at all. At each iteration, we use the ranking obtained by the considered multi-objective selection paradigm in order to score the quality of solutions with respect to convergence. We then bias standard variation operators accordingly, in order to help finding high-quality solutions as early as possible, and thus improving algorithm convergence. To do so, we train a random forest regression model (Breiman, 2001) in order to predict the rank of solutions from the current population based on the value of their variables. After training, we extract the importance of variables from the obtained model to select which variables will undergo variation. Selected experimental results are reported in Figure 5.18 on four problems from continuous multi-objective optimization with a varying number of objectives as well as convergence and diversity variables — DTLZ3 from Deb et al. (2005). Three approaches are considered: a conventional evolutionary approach (org), the proposed approach based on variable importance (var-imp), and a cheating approach with perfect knowledge of convergence and diversity variables (ideal). Generational distance (top) is used to account for convergence, while inverted generational distance (bottom) accounts for diversity. Besides showing that the proposed approach achieves a significantly better convergence on some well-established continuous benchmark functions, our investigations suggest that the design

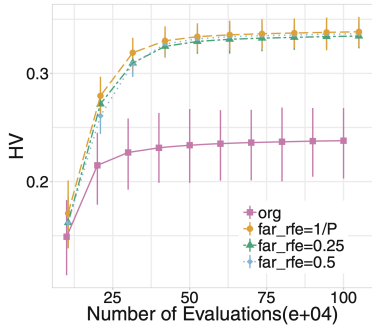


Figure 5.19: Convergence profile of the algorithm with and without enhanced variation on the vibrating beam problem with respect to hypervolume.

of machine learning-enhanced variation operators is a promising research direction that might help catching the underlying difficulty of multi-objective optimization problems.

In Ito et al. (2019), we further improve the proposed methodology by classifying the variables as influential or non-influential, this by mutating one variable at a time on random solutions among the best solutions found so far. The method estimates that influential variables affect the convergence of the population, and increases their recombination rate. We also experiment the effectiveness of the proposed approach on a real-world bi-objective vibrating beam problem. The experimental results reported in Figure 5.19 show that different settings of the method contribute to achieve faster and better convergence.

5.6 Outlook and Current Investigations

This chapter provided an overview of our research contributions on efficient multi-objective optimization, with a particular focus on the decomposition-based search paradigm. They can be considered along three interconnected axes. Firstly, by considering the decomposition in a very local way, we paid a particular attention to the design of cooperative rules in order to improve approximation quality while enabling parallelism to take full benefit from large-scale distributed computational resources. In this regard, we argue that the high level of parallelism offered by the decomposition framework is a strong feature for designing more powerful search procedures, and for solving increasingly complex optimization problems. Secondly, we were interested in the design of search methods that integrate meta-models for solving expensive multi-objective optimization problems. A fine-grained taxonomy allowed us to instantiate existing approaches and new alternatives under a common framework, and to systematically highlight their main design choices and components, thus revealing a number of important considerations for surrogate-assisted multi-objective search. They include the construction of the training set and of an ensemble of local meta-models clustered in different regions of the objective space, together with the computationally-intensive evaluation of a diverse batch of solutions under the real, expensive objectives. This is certainly of high importance in terms of computational complexity: not only does this reduce the number of computationally-demanding model training tasks, but this also enables the costly evaluation of multiple solutions in parallel. Our investigations also pointed out the critical importance of design components with respect to the available search budget. We then presented a surrogate-assisted modular approach for expensive multi-objective combinatorial optimization, based on a Walsh meta-model. We found that there is a non-trivial interaction between the inner optimizer of the surrogates — used as substitutes of the objectives — and the way the next solution to be evaluated is selected. At last, we addressed important complementary algorithm components for efficient (decomposition-based) multi-objective search. By analyzing the impact of the population and of the formulation of scalar sub-problems, by proposing new mechanisms for variation

as well parent and survival selection, and most importantly by conducting extensive empirical investigations to analyze and compare their search dynamics and anytime performance, we were able to make a step towards a more fundamental understanding of what multi-objective search in general, and the decomposition framework in particular, is able to achieve.

Following a recent survey on decomposition-based multi-objective search (Trivedi et al., 2017), our contributions are mainly concerned with the following research lines: (i) the specification of weight vectors, (ii) the mating selection and replacement mechanisms, (iii) the computational effort underlying decomposition-based search, and (iv) the design and integration of variation operators. Even more recently, Li (2021) reviews major developments on decomposition, among which our contributions address all the considered components, namely: (i) the weight vector setting, in particular with adaptation methods, (ii) the scalar sub-problems formulation, with the fundamental study of scalarizing functions' contour lines and improvement regions, (iii) the selection mechanisms, with new strategies for both parent and survival selection, and (iv) variation, with local search and model-based operators. Li (2021) further discusses advanced and emerging topics. Of special interest to our work are surrogate-assisted approaches for expensive optimization, and parallel computing.

Regarding the decomposition framework, we are currently working on making a stronger connection with landscape analysis, as discussed in Chapter 3. For instance, we recently studied the "aggregated" landscape of multiple scalar sub-problems as a whole, showing that it helps explain and predict decomposition-based search (Cosson et al., 2021). We are also considering using tools and features from single-objective landscape analysis to characterize the different scalar sub-problems defined by decomposition. This is certainly of great interest for the study of sub-problems with an heterogeneous level of difficulty, that is, problems with heterogeneous objectives (Santana et al., 2022), for which the scalarizing function corresponding to each weight vector is expected to have a different complexity. Informing about the heterogeneity of sub-problems might lead to a better distribution of the search effort among the sub-problems. Related to this, we have seen that the setting of the scalarizing function, and more importantly its contour lines, directly impact the trajectory, the dynamics and the performance of sub-problem solving. This suggests that some algorithm design choices could be reconsidered, including the formulation of heterogeneous scalarizing functions for the different weight vectors in order to balance the difficulty of sub-problems and the distribution of target solutions. To do so, we plan to analyze the landscape of sub-problems and to investigate the configuration of scalarizing functions' opening angles, thus abstracting away from any particular closed-form scalarizing function. In a complementary way, we believe that some quality indicators from Chapter 4 — see also Hansen and Jaszkiewicz (1998) — have a strong connection with the way in which decomposition is defined, and might actually be relevant to clarify the distribution of solutions within the population from decomposition-based search. Not only this, we anticipate it might also help to gain

a better understanding of the differences and similarities between these two selection paradigms from multi-objective search. We would not be complete without mentioning that the cooperation among sub-problems certainly helps the overall search procedure. Investigating whether cooperation actually implies a better search behavior could allow us to better understand why solving sub-problems cooperatively appears to be fundamentally more efficient than solving them independently. To do so, we would need to capture the evolvability of solutions from one sub-problem to another.

Another obvious perspective to our work is to leverage the decentralized nature of decomposition and the computational power provided by modern parallel platforms. The aim is to achieve a high efficiency and a high scalability without deteriorating search performance. A first step is to identify the different sources of parallelism induced by the decomposition, both in the objective space and in the variable space, independently of its effective parallelization. Our work on the learning of variable importance could serve as a basis for variable space decomposition, in order to study the effect of linkage among variables (Pelikan et al., 2000), especially when variable interactions are intricate. However, scalability often implies unbalanced computations. As pointed out above, different sub-problems might require a varying computational effort. To our knowledge, no distributed approach taking into account the cooperation among multiple parallel processes has been studied so far for solving heterogeneous sub-problems. We argue that decomposition provides a framework in which online adaptation could help balance the search effort among sub-problems, but also the algorithm components and parameters used by the cooperative and distributed search process. Finally, we believe that surrogate-assisted search may prove relevant for multi-fidelity and/or simulation optimization (Branke, 2019; Forrester et al., 2007); e.g., when evaluating the objective(s) involve expensive simulations (Rifki et al., 2020). In this line, we plan to address other discrete optimization domains, such as permutation problems. In principle, the proposed surrogate-assisted multi-objective search approach is compatible with any accurate meta-model. However, studying the combined effects of algorithm components from surrogate-assisted search can only be a function of a target optimization problem. More generally, we advocate for a more systematic benchmarking analysis of such considerations for improving our fundamental understanding on the design of multi-objective search, on the key differences among algorithm classes, and on their success in solving challenging problems from multi-objective optimization. In the next chapter, we conclude the manuscript with general considerations and a research plan for massive optimization.

Optimization is now ubiquitous to countless modern engineering and scientific applications. Problems and algorithms are increasingly large-scale and heterogeneous, requiring to deal with a huge number of variables and conflicting objectives of different nature. From the application point of view, fields of particular interest relate to sustainable systems, complex scheduling or multidisciplinary engineering design and innovation, for which many optimization models are increasingly complex and involve large-size instances, cross-domain formulations and heterogeneous objectives. Such characteristics lead to *massive* optimization problems, and raise new important and difficult scientific challenges for researchers and practitioners, that traditional approaches will hardly succeed when facing them. Those techniques shall be taken to the next level for solving heterogeneous problem classes, with a large number of variables and objectives. We must therefore push the boundaries of existing approaches, in order to go beyond the problem scale investigated so far in the literature, and to design innovative flexible general-purpose computational intelligence methods able to efficiently and effectively tackle such massive optimization problems.

Challenges

Although some research dealing with the aforementioned characteristics can be found, we target a unified integrated approach tackling the issues from today's complex application domains in engineering design and sustainable systems. In particular, we are interested in setting up the foundations and developing cutting-edge *autonomous solvers* able to globally and jointly address the challenges encountered in problems from massive optimization:

1. **Large-scale optimization problems**, which commonly involve hundreds of variables that induce a large increase in the space where the search algorithm operates.
2. **Any-objective optimization problems**, where one, multiple, or many criteria are to be simultaneously optimized, typically leading to a significant increase in the number of optimal trade-offs to be identified.
3. **Cross-domain optimization problems**, dealing with continuous, integer, categorical variables, or even more complex structures such as permutations, strings, trees, or graphs, that may be mixed among themselves.
4. **Expensive optimization problems**, where the propagation of environmental parameters, the requirement of heavy simulations, or simply the black-box nature of the objectives makes it

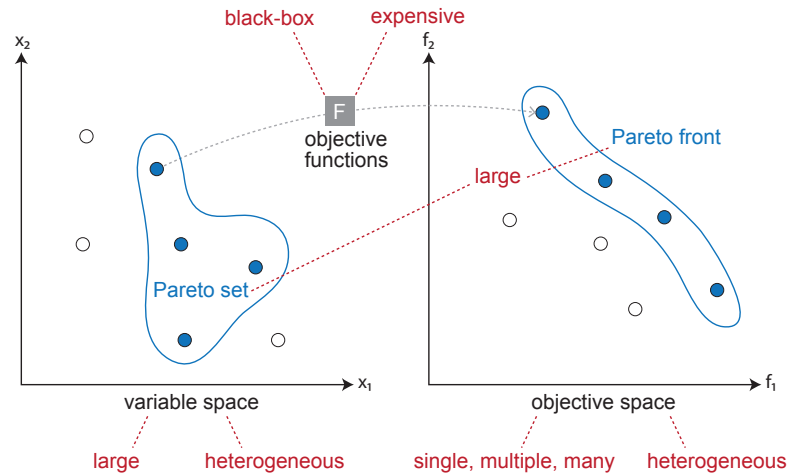


Figure 6.1: Challenges from massive optimization.

already computationally demanding to obtain the quality of one single candidate solution at the evaluation stage.

The general goal is to foster the next generation of search algorithms for solving such problems by precisely investigating the modeling, the resolution and the fundamental and experimental analysis of massive optimization problems, with a clear emphasis on their multi-objective nature. As illustrated in Figure 6.1, such massive optimization problems raise new challenges, in particular because of (a) their *dimensionality* in terms of *variables*, (b) of *objectives*, (c) their *heterogeneity*, and (d) their *black-box* and *expensive* nature, our research program aims at jointly addressing them, and is organized following three interconnected scientific goals described below. Obviously, other important challenges that are not considered below relate to the number of constraints involved in the problem formulation and how to handle them, as well as the different sources of uncertainty that an application can face and how to take them into account during its resolution.

Landscape-aware Search

The class of optimization problems encountered in real-life complex application domains is wide and heterogeneous. This explains the plethora of ad-hoc optimization techniques specialized in solving a particular problem formulation. On the contrary, general-purpose methods such as branch and bound or search heuristics constitute upper-level methodologies that can be used as guiding strategies in designing underlying search algorithms. One of our goal precisely lies in the foundation, analysis and intelligent design of enhanced general-purpose algorithms, search paradigms and their design principles, as well as innovative ways of combining them. However, being effective and efficient in solving the target problem always requires a proper configuration and adaptation. However, most algorithms continue to be designed on the basis of intuition, and require an intensive phase of trials and errors for parameter setting. One way of addressing this in practice is to rely on automated algorithm configuration in order to automatically configure an algorithm by finding the most appropriate

parameter setting, specialized for a given set of problem instances. Complementarily, we aim at avoiding hyper-specialized approaches, and at improving the way we develop algorithms by incorporating a more fundamental approach in their design process. Our goal is to understand the difficulties a given optimization approach has to face, and what makes it efficient, independently of the target application, by deriving high-level and relevant features able to capture problem difficulty by means of tools from landscape analysis, as well as statistics and machine learning data analysis. Such an analytics-driven methodology, based on landscape analysis and extensive benchmarking efforts, would allow us, not only to understand what makes a problem difficult or an optimization approach efficient, but also to predict the algorithm performance, to select the most appropriate configuration from an algorithm portfolio, and to adapt and improve the algorithm design for unknown optimization domain and problem instances. Such a cross-domain autonomous solver would adaptively adjust its internal mechanisms in order to fully take advantage of the opportunities offered by the target massive optimization problem.

Model-assisted Search

In expensive optimization, evaluating the quality of a candidate solution is particularly demanding computationally speaking. This is typically the case when this evaluation step corresponds to the result of a (black-box) complex system simulation, or because of the large number of environmental parameters encountered in multidisciplinary engineering design and innovation, as well as sustainable systems. In this context, existing algorithms from optimization and computational intelligence suffer from slow convergence, and their scalability raises new scientific challenges. To overcome this, we will rely on surrogate models and machine learning algorithms in order to predict the approximation quality without systematically computing their (expensive) objective value(s). The goal here is to accelerate the convergence of the optimization process and to improve the quality of final solutions. More particularly, we will focus on the suitability of advanced statistical and machine learning meta-models for large-scale optimization, the choice of the output to be predicted by these meta-models, their prediction accuracy and their parameter sensibility, the uncertainties and inaccuracies occurring in their responses, the choice of the data set from which the meta-model learns from, and the integration of the learning phase within the search process. Because of the target application context, the computational cost of designed approaches is prohibitive. As a consequence, we will attach a particular attention to distributed approaches for addressing these different issues, with an effective parallelization on high performance computing platforms. Complementarily, we will investigate model-assisted computational intelligence algorithms, that consist in explicitly modeling the key features — such as variable interactions — that impact approximation quality, and to use this model as an algorithm component in order to produce new candidate solutions with an expected improved quality.

Decomposition-based Decentralized Search

Given the large-scale nature of the target applications and the underlying optimization problems, in terms of the number of variables and objectives, a natural answer is to decompose the original massive optimization problem to be solved into several sub-problems for which solutions are computed and aggregated taking inspiration from the “divide and conquer” paradigm. However, setting up an effective decomposition-based approach relies on the design and integration of several components that are to be configured accurately. Firstly, we will address the definition of the set of sub-problems to be solved cooperatively, by decomposing the original problem into a set of sub-problems within a smaller region of the variable space and/or the objective space, so as to increase the efficiency of the search process. Secondly, we will design cooperative computational intelligence algorithms and mechanisms in order to solve each sub-problem, and to specify the local rules of interaction and cooperation governing the global search process. The idea is to view the solving of an optimization problem as a complex system operating at different local parts so that the overall global computational power is eventually larger than the sum of its parts. At last, we will take advantage from the decentralized nature of decomposition-based approaches in order to deploy them efficiently on large-scale distributed and parallel platforms. On the one hand, the power of modern and massively parallel computing platforms is becoming both huge and increasingly available for the community. On the other hand, and following the evolution of modern computational science, the characteristics of massive optimization give rise to difficult challenges, beyond the ability of commonly-used algorithms. In this respect, there is evidence that decentralized computation will play a crucially important role in order to foster the next generation of optimization techniques, and to accelerate their widespread uptake. One main issue we will tackle is to consider the cooperation rules within the different search procedures operating at every sub-problem. This can in fact constitute a bottleneck towards the design of highly scalable parallel decomposition for massive optimization.

All in all, we argue that considering the combined design of landscape-aware, model-assisted, and decentralized decomposition-based search algorithms will allow for an efficient and effective approach for solving upcoming problems from massive optimization.

Bibliography

Self-citations appear in color.

- Abbasi, M., Paquete, L., Liefvooghe, A., Pinheiro, M., and Matias, P. (2013). 'Improvements on Bicriteria Pairwise Sequence Alignment: Algorithms and Applications'. In: *Bioinformatics* 29 (8), pp. 996–1003.
- Abdelkafi, O., Derbel, B., Liefvooghe, A., and Whitley, D. (2020). 'On the Design of a Partition Crossover for the Quadratic Assignment Problem'. In: *Parallel Problem Solving from Nature, PPSN XVI*. Ed. by T. Bäck, M. Preuss, A. H. Deutz, H. Wang, C. Doerr, M. T. M. Emmerich, and H. Trautmann. Vol. 12270. Lecture Notes in Computer Science. Leiden, The Netherlands: Springer, pp. 303–316.
- Aboutaib, B., Verel, S., Fonlupt, C., Derbel, B., Liefvooghe, A., and Ahiod, B. (2020). 'On Stochastic Fitness Landscapes: Local Optimality and Fitness Landscape Analysis for Stochastic Search Operators'. In: *Parallel Problem Solving from Nature, PPSN XVI*. Ed. by T. Bäck, M. Preuss, A. H. Deutz, H. Wang, C. Doerr, M. T. M. Emmerich, and H. Trautmann. Vol. 12270. Lecture Notes in Computer Science. Leiden, The Netherlands: Springer, pp. 97–110.
- Adenso-Diaz, B. and Laguna, M. (2006). 'Fine-Tuning of Algorithms Using Fractional Experimental Design and Local Search'. In: *Operations Research* 54 (1), pp. 99–114.
- Aguirre, H. E., Liefvooghe, A., Verel, S., and Tanaka, K. (2013). 'A Study on Population Size and Selection Lapse in Many-objective Optimization'. In: *Proceedings of the 2013 Congress on Evolutionary Computation (CEC 2013)*. Cancún, Mexico: IEEE Press, pp. 1507–1514.
- (2014). 'An Analysis on Selection for High-Resolution Approximations in Many-Objective Optimization'. In: *Parallel Problem Solving from Nature, PPSN XIII*. Ed. by T. Bartz-Beielstein, J. Branke, B. Filipič, and J. Smith. Vol. 8672. Lecture Notes in Computer Science. Ljubljana, Slovenia: Springer, Heidelberg, Germany, pp. 487–497.
- Aguirre, H. E. and Tanaka, K. (2007). 'Working Principles, Behavior, and Performance of MOEAs on MNK-landscapes'. In: *European Journal of Operational Research* 181 (3), pp. 1670–1690.
- Allmendinger, R., Handl, J., and Knowles, J. (2015). 'Multiobjective Optimization: When Objectives Exhibit Non-uniform Latencies'. In: *European Journal of Operational Research* 243 (2), pp. 497–513.
- Allmendinger, R., Jaszkiwicz, A., Liefvooghe, A., and Tammer, C. (2021). *What if we Increase the Number of Objectives? Theoretical and Empirical Implications for Many-objective Optimization*. Research Report CoRR abs/2106.03275. arXiv.
- Allmendinger, R. and Knowles, J. (2021). *Heterogeneous Objectives: State-of-the-art and Future Research*. Research Report CoRR abs/2103.15546. arXiv.
- Alsouly, H., Kirley, M., and Muñoz, M. A. (2022). *An Instance Space Analysis of Constrained Multi-Objective Optimization Problems*. Research Report CoRR abs/2203.00868. arXiv.
- Aneja, Y. P. and Nair, K. P. K. (1979). 'Bicriteria Transportation Problem'. In: *Management Science* 25 (1), pp. 73–78.
- Auger, A., Bader, J., Brockhoff, D., and Zitzler, E. (2012). 'Hypervolume-based Multiobjective Optimization: Theoretical Foundations and Practical Implications'. In: *Theoretical Computer Science* 425, pp. 75–103.
- Bader, J. and Zitzler, E. (2011). 'HypE: An Algorithm for Fast Hypervolume-Based Many-Objective Optimization'. In: *Evolutionary Computation* 19 (1), pp. 45–76.

- Baptista, R. and Poloczek, M. (2018). ‘Bayesian Optimization of Combinatorial Structures’. In: *Proceedings of the 35th International Conference on Machine Learning (ICML 2018)*. Ed. by J. G. Dy and A. Krause. Vol. 80. Proceedings of Machine Learning Research. Stockholm, Sweden: PMLR, pp. 471–480.
- Bartz-Beielstein, T. (2006). *Experimental Research in Evolutionary Computation: The New Experimentalism*. Berlin, Germany: Springer.
- Bartz-Beielstein, T., Doerr, C., Berg, D. van den, Bossek, J., Chandrasekaran, S., Eftimov, T., Fischbach, A., Kerschke, P., Cava, W. L., Lopez-Ibanez, M., Malan, K. M., Moore, J. H., Naujoks, B., Orzechowski, P., Volz, V., Wagner, M., and Weise, T. (2020). *Benchmarking in Optimization: Best Practice and Open Issues*. Research Report CoRR abs/2007.03488. arXiv.
- Bartz-Beielstein, T. and Zaefferer, M. (2017). ‘Model-based Methods for Continuous and Discrete Global Optimization’. In: *Applied Soft Computing* 55, pp. 154–167.
- Basseur, M., Derbel, B., Goëffon, A., and Liefoghe, A. (2016). ‘Experiments on Greedy and Local Search Heuristics for d -dimensional Hypervolume Subset Selection’. In: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2016*. Ed. by T. Friedrich, F. Neumann, and A. M. Sutton. Denver, Colorado, USA: ACM Press, pp. 541–548.
- Basseur, M., Goëffon, A., Liefoghe, A., and Verel, S. (2013). ‘On Set-based Local Search for Multiobjective Combinatorial Optimization’. In: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2013*. Ed. by C. Blum and E. Alba. New York, NY: ACM Press, pp. 471–478.
- Bazgan, C., Hugot, H., and Vanderpooten, D. (2009). ‘Solving Efficiently the 0–1 Multi-objective Knapsack Problem’. In: *Computers & Operations Research* 36 (1), pp. 260–279.
- Bazgan, C., Jamain, F., and Vanderpooten, D. (2013). ‘On the Number of Non-dominated Points of a Multicriteria Optimization Problem’. In: *Discrete Applied Mathematics* 161 (18), pp. 2841–2850.
- Beham, A., Wagner, S., and Affenzeller, M. (2018). ‘Algorithm Selection on Generalized Quadratic Assignment Problem Landscapes’. In: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2018*. Ed. by H. E. Aguirre and K. Takadama. New York, NY: ACM Press, pp. 253–260.
- Belkhir, N., Dréo, J., Savéant, P., and Schoenauer, M. (2017). ‘Per Instance Algorithm Configuration of CMA-ES with Limited Budget’. In: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2017*. Ed. by P. A. N. Bosman. New York, NY: ACM Press, pp. 681–688.
- Beverglieri, N., Derbel, B., Liefoghe, A., Aguirre, H. E., and Tanaka, K. (2019). ‘Surrogate-assisted Multiobjective Optimization based on Decomposition: A Comprehensive Comparative Analysis’. In: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2019*. Ed. by M. López-Ibañez, A. Auger, and T. Stützle. Prague, Czech Republic: ACM Press, pp. 507–515.
- Beverglieri, N., Derbel, B., Liefoghe, A., Aguirre, H. E., Zhang, Q., and Tanaka, K. (2020). ‘Designing Parallelism in Surrogate-assisted Multiobjective Optimization based on Decomposition’. In: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2020*. Ed. by C. A. Coello Coello. Cancún, Mexico: ACM Press, pp. 462–470.
- (2022). ‘Surrogate-assisted Multiobjective Optimization: Taxonomy and Hybridization’. (under review).
- Beume, N., Fonseca, C. M., López-Ibañez, M., Paquete, L., and Vahrenhold, J. (2009). ‘On the Complexity of Computing the Hypervolume Indicator’. In: *IEEE Transactions on Evolutionary Computation* 13 (5), pp. 1075–1082.
- Beume, N., Naujoks, B., and Emmerich, M. T. M. (2007). ‘SMS-EMOA: Multiobjective Selection Based on Dominated Hypervolume’. In: *European Journal of Operational Research* 181 (3), pp. 1653–1669.
- Birattari, M. (2009). *Tuning Metaheuristics: A Machine Learning Perspective*. Vol. 197. Studies in Computational Intelligence. Berlin, Heidelberg: Springer.

- Birattari, M., Stützle, T., Paquete, L., and Varrentrapp, K. (2002). 'A Racing Algorithm for Configuring Metaheuristics'. In: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2002*. Ed. by W. B. Langdon et al. Morgan Kaufmann Publishers, San Francisco, CA, pp. 11–18.
- Bischi, B., Mersmann, O., Trautmann, H., and Preuss, M. (2012). 'Algorithm Selection Based on Exploratory Landscape Analysis and Cost-sensitive Learning'. In: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2012*. Ed. by T. Soule and J. H. Moore. New York, NY: ACM Press, pp. 313–320.
- Bradstreet, L., While, L., and Barone, L. (2007). 'Incrementally Maximising Hypervolume for Selection in Multi-objective Evolutionary Algorithms'. In: *Proceedings of the 2007 Congress on Evolutionary Computation (CEC 2007)*. Singapore: IEEE, pp. 3203–3210.
- Branke, J., Deb, K., Miettinen, K., and Slowinski, R., eds. (2008). *Multiobjective Optimization – Interactive and Evolutionary Approaches*. Vol. 5252. Lecture Notes in Computer Science. Berlin, Germany: Springer-Verlag.
- Branke, J. (2019). 'Simulation Optimisation: Tutorial'. In: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2019*. Ed. by M. López-Ibáñez, A. Auger, and T. Stützle. Prague, Czech Republic: ACM Press, pp. 862–889.
- Breiman, L. (2001). 'Random Forests'. In: *Machine Learning* 45 (1), pp. 5–32.
- Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and Regression Trees*. Wadsworth & Brooks.
- Bringmann, K., Friedrich, T., and Klitzke, P. (2014a). 'Generic Postprocessing via Subset Selection for Hypervolume and Epsilon-Indicator'. In: *Parallel Problem Solving from Nature, PPSN XIII*. Ed. by T. Bartz-Beielstein, J. Branke, B. Filipič, and J. Smith. Vol. 8672. Lecture Notes in Computer Science. Springer, Heidelberg, Germany, pp. 518–527.
- (2014b). 'Two-dimensional Subset Selection for Hypervolume and Epsilon-indicator'. In: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2014*. Ed. by C. Igel and D. V. Arnold. New York, NY: ACM Press.
- (2015). 'Efficient Computation of Two-dimensional Solution Sets Maximizing the Epsilon-Indicator'. In: *Proceedings of the 2015 Congress on Evolutionary Computation (CEC 2015)*. Sendai, Japan: IEEE Press, pp. 970–977.
- Brockhoff, D., Wagner, T., and Trautmann, H. (2015). 'R2 Indicator Based Multiobjective Search'. In: *Evolutionary Computation* 23 (3), pp. 369–395.
- Brockhoff, D., Tusar, T., Auger, A., and Hansen, N. (2019). *Using Well-Understood Single-Objective Functions in Multiobjective Black-Box Optimization Test Suites*. Research Report CoRR abs/1604.00359. arXiv.
- Cahon, S., Melab, N., and Talbi, E.-G. (2004). 'ParadisEO: A Framework for the Reusable Design of Parallel and Distributed Metaheuristics'. In: *Journal of Heuristics* 10 (3), pp. 357–380.
- Candan, C., Goëffon, A., Lardeux, F., and Saubion, F. (2013). 'Non Stationary Operator Selection with Island Models'. In: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2013*. Ed. by C. Blum and E. Alba. Amsterdam, The Netherlands: ACM Press, pp. 1509–1516.
- Carnell, R. (2020). *lhs: Latin Hypercube Samples*. R package version 1.0.2.
- Cerqueus, A., Gandibleux, X., Przybylski, A., and Saubion, F. (2017). 'On Branching Heuristics for the Bi-objective 0/1 Unidimensional Knapsack Problem'. In: *Journal of Heuristics* 23 (5), pp. 285–319.
- Chan, T. M. (2013). 'Klee's Measure Problem Made Easy'. In: *54th Annual Symposium on Foundations of Computer Science, FOCS 2013*. IEEE Computer Society Press, pp. 410–419.

- Chugh, T., Jin, Y., Miettinen, K., Hakanen, J., and Sindhya, K. (2018). 'A Surrogate-assisted Reference Vector Guided Evolutionary Algorithm for Computationally Expensive Many-Objective Optimization'. In: *IEEE Transactions on Evolutionary Computation* 22 (1), pp. 129–142.
- Chugh, T., Sindhya, K., Hakanen, J., and Miettinen, K. (2019). 'A Survey on Handling Computationally Expensive Multiobjective Optimization Problems with Evolutionary Algorithms'. In: *Soft Computing* 23 (9), pp. 3137–3166.
- Coello Coello, C. A., Lamont, G. B., and Van Veldhuizen, D. A. (2007). *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, New York, NY.
- Coello Coello, C. A. and Reyes-Sierra, M. (2004). 'A Study of the Parallelization of a Coevolutionary Multi-objective Evolutionary Algorithm'. In: *Third Mexican International Conference on Artificial Intelligence, MICAI 2004*. Ed. by R. Monroy, G. Arroyo-Figueroa, L. E. Sucar, and H. Sossa. Vol. 2972. Lecture Notes in Artificial Intelligence. Springer, Heidelberg, Germany, pp. 688–697.
- Conover, W. J. (1999). *Practical Nonparametric Statistics*. Third. John Wiley & Sons, New York, NY.
- Cornu, M., Cazenave, T., and Vanderpooten, D. (2017). 'Perturbed Decomposition Algorithm applied to the Multi-objective Traveling Salesman Problem'. In: *Computers & Operations Research* 79, pp. 314–330.
- Cosson, R., Derbel, B., Liefoghe, A., Aguirre, H. E., Tanaka, K., and Zhang, Q. (2021). 'Decomposition-Based Multi-objective Landscape Features and Automated Algorithm Selection'. In: *Proceedings of EvoCOP 2021 – 21st European Conference on Evolutionary Computation in Combinatorial Optimization*. Ed. by C. Zarges and S. Verel. Vol. 12692. Lecture Notes in Computer Science. Virtual Event: Springer, Berlin, pp. 34–50.
- Cosson, R., Derbel, B., Liefoghe, A., Verel, S., Aguirre, H. E., Tanaka, K., and Zhang, Q. (2022). 'Cost-vs-Accuracy of Sampling in Multi-objective Combinatorial Exploratory Landscape Analysis'. In: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2022*. Boston, MA, USA: ACM Press.
- Daolio, F., Liefoghe, A., Verel, S., Aguirre, H., and Tanaka, K. (2015). 'Global vs Local Search on Multi-objective NK-Landscapes: Contrasting the Impact of Problem Features'. In: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2015*. Ed. by S. Silva and A. I. Esparcia-Alcázar. Madrid, Spain: ACM Press, pp. 369–376.
- (2017). 'Problem Features versus Algorithm Performance on Rugged Multiobjective Combinatorial Fitness Landscapes'. In: *Evolutionary Computation* 25 (4), pp. 555–585.
- Daolio, F., Verel, S., Ochoa, G., and Tomassini, M. (2012). 'Local Optima Networks and the Performance of Iterated Local Search'. In: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2012*. Ed. by T. Soule and J. H. Moore. New York, NY: ACM Press, pp. 369–376.
- Das, S. and Suganthan, P. N. (2011). 'Differential Evolution: A Survey of the State-of-the-art'. In: *IEEE Transactions on Evolutionary Computation* 15 (1).
- Dean, T. and Boddy, M. S. (1988). 'An Analysis of Time-Dependent Planning'. In: *Proceedings of the 7th National Conference on Artificial Intelligence, AAAI 88*. Ed. by H. E. Shrobe, T. M. Mitchell, and R. G. Smith. AAAI Press/MIT Press, Menlo Park, CA, pp. 49–54.
- Deb, K. (2001). *Multi-Objective Optimization Using Evolutionary Algorithms*. Chichester, UK: Wiley.
- Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). 'A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II'. In: *IEEE Transactions on Evolutionary Computation* 6 (2), pp. 182–197.
- Deb, K., Thiele, L., Laumanns, M., and Zitzler, E. (2005). 'Scalable Test Problems for Evolutionary Multi-objective Optimization'. In: *Evolutionary Multiobjective Optimization*. Ed. by A. Abraham, L. Jain, and R. Goldberg. Advanced Information and Knowledge Processing. London, UK: Springer, pp. 105–145.

- Derbel, B., Liefoghe, A., Zhang, Q., Aguirre, H., and Tanaka, K. (2016). 'Multi-objective Local Search based on Decomposition'. In: *Parallel Problem Solving from Nature, PPSN XIV*. Ed. by J. Handl, E. Hart, P. R. Lewis, M. López-Ibáñez, G. Ochoa, and B. Paechter. Vol. 9921. Lecture Notes in Computer Science. Edinburgh, UK: Springer, Heidelberg, Germany, pp. 431–441.
- Derbel, B., Brockhoff, D., and Liefoghe, A. (2013). 'Force-Based Cooperative Search Directions in Evolutionary Multi-objective Optimization'. In: *Evolutionary Multi-criterion Optimization, EMO 2013*. Ed. by R. C. Purshouse, P. J. Fleming, C. M. Fonseca, S. Greco, and J. Shaw. Vol. 7811. Lecture Notes in Computer Science. Sheffield, UK: Springer, Heidelberg, Germany, pp. 383–397.
- Derbel, B., Brockhoff, D., Liefoghe, A., and Verel, S. (2014a). 'On the Impact of Multiobjective Scalarizing Functions'. In: *Parallel Problem Solving from Nature, PPSN XIII*. Ed. by T. Bartz-Beielstein, J. Branke, B. Filipič, and J. Smith. Vol. 8672. Lecture Notes in Computer Science. Ljubljana, Slovenia: Springer, Heidelberg, Germany, pp. 548–558.
- Derbel, B., Humeau, J., Liefoghe, A., and Verel, S. (2014b). 'Distributed Localized Bi-objective Search'. In: *European Journal of Operational Research* 239, pp. 731–743.
- Derbel, B., Liefoghe, A., Marquet, G., and Talbi, E.-G. (2015). 'A Fine-Grained Message Passing MOEA/D'. In: *Proceedings of the 2015 Congress on Evolutionary Computation (CEC 2015)*. Sendai, Japan: IEEE Press, pp. 1837–1844.
- Derbel, B., Liefoghe, A., Zhang, Q., Verel, S., Aguirre, H. E., and Tanaka, K. (2018). 'A Set-oriented MOEA/D'. In: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2018*. Ed. by H. E. Aguirre and K. Takadama. Kyoto, Japan: ACM Press, pp. 617–624.
- Dréo, J., Liefoghe, A., Verel, S., Schoenauer, M., Merelo, J. J., Quemy, A., Bouvier, B., and Gmys, J. (2021). 'Paradiseo: From a Modular Framework for Evolutionary Computation to the Automated Design of Metaheuristics – 22 Years of Paradiseo'. In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO 2021*. Ed. by F. Chicano and K. Krawiec. Lille, France: ACM Press, pp. 1522–1530.
- Drozdzik, M., Tanaka, K., Aguirre, H., Verel, S., Liefoghe, A., and Derbel, B. (2014). 'An Analysis of Differential Evolution Parameters on Rotated Bi-objective Optimization Functions'. In: *Simulated Evolution and Learning (SEAL 2014)*. Ed. by G. Dick, W. N. Browne, P. Whigham, M. Zhang, L. T. Bui, H. Ishibuchi, Y. Jin, X. Li, Y. Shi, P. Singh, K. C. Tan, and K. Tang. Vol. 8886. Lecture Notes in Computer Science. Dunedin, New Zealand: Springer, pp. 143–154.
- Drugan, M. M. and Thierens, D. (2012). 'Stochastic Pareto Local Search: Pareto Neighbourhood Exploration and Perturbation Strategies'. In: *Journal of Heuristics* 18 (5), pp. 727–766.
- Dubois-Lacoste, J., López-Ibáñez, M., and Stützle, T. (2015). 'Anytime Pareto Local Search'. In: *European Journal of Operational Research* 243 (2), pp. 369–385.
- Durillo, J. J. and Nebro, A. J. (2011). 'jMetal: A Java Framework for Multi-objective Optimization'. In: *Advances in Engineering Software* 42, pp. 760–771.
- Durillo, J., Zhang, Q., Nebro, A., and Alba, E. (2011). 'Distribution of Computational Effort in Parallel MOEA/D'. In: *Learning and Intelligent Optimization, LION 5*. Ed. by C. A. Coello Coello. Vol. 6683. Lecture Notes in Computer Science. Springer, Heidelberg, Germany, pp. 488–502.
- Ehrgott, M. and Klamroth, K. (1997). 'Connectedness of Efficient Solutions in Multiple Criteria Combinatorial Optimization'. In: *European Journal of Operational Research* 97 (1), pp. 159–166.
- Ehrgott, M. (2005). *Multicriteria Optimization*. 2nd. Springer, Berlin, Germany.
- Ehrgott, M. and Gandibleux, X. (2001). 'Bounds and Bound Sets for Biobjective Combinatorial Optimization Problems'. In: *Multiple Criteria Decision Making in the New Millennium*. Ed. by M. Köksalan and S. Zionts. Berlin, Heidelberg: Springer, pp. 241–253.

- Eichfelder, G., Gandibleux, X., Geiger, M. J., Jahn, J., Jaszkiwicz, A., Knowles, J. D., Shukla, P. K., Trautmann, H., and Wessing, S. (2015). 'Heterogeneous Functions'. In: *Understanding Complexity in Multiobjective Optimization*. Report from Dagstuhl Seminar 15031. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, pp. 121–129.
- Emelichev, V. A. and Perepelitsa, V. A. (1992). 'On the Cardinality of the Set of Alternatives in Discrete Many-criterion Problems'. In: *Discrete Mathematics and Applications* 2 (5), pp. 461–471.
- Falcón-Cardona, J. G. and Coello Coello, C. A. (2020). 'Indicator-Based Multi-Objective Evolutionary Algorithms: A Comprehensive Survey'. In: *ACM Computing Surveys* 53 (2), p. 29.
- Falcón-Cardona, J. G., Liefooghe, A., and Coello Coello, C. A. (2020). 'An Ensemble Indicator-Based Density Estimator for Evolutionary Multi-objective Optimization'. In: *Parallel Problem Solving from Nature, PPSN XVI*. Ed. by T. Bäck, M. Preuss, A. H. Deutz, H. Wang, C. Doerr, M. T. M. Emmerich, and H. Trautmann. Vol. 12270. Lecture Notes in Computer Science. Leiden, The Netherlands: Springer, pp. 201–214.
- Figueira, J. R., Paquete, L., Simões, M., and Vanderpooten, D. (2013). 'Algorithmic Improvements on Dynamic Programming for the Bi-objective {0,1} Knapsack Problem'. In: *Computational Optimization and Applications* 56 (1), pp. 97–111.
- Forrester, A. I., Sóbester, A., and Keane, A. J. (2007). 'Multi-fidelity Optimization via Surrogate Modelling'. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 463 (2088), pp. 3251–3269.
- Freund, Y. and Schapire, R. E. (1997). 'A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting'. In: *Journal of Computer and System Sciences* 55 (1), pp. 119–139.
- Garrett, D. and Dasgupta, D. (2008). 'Multiobjective Landscape Analysis and the Generalized Assignment Problem'. In: *Learning and Intelligent Optimization, LION 2*. Ed. by V. Maniezzo, R. Battiti, and J.-P. Watson. Vol. 5313. Lecture Notes in Computer Science. Springer, Heidelberg, Germany, pp. 110–124.
- (2009). 'Plateau Connection Structure and Multiobjective Metaheuristic Performance'. In: *Proceedings of the 2009 Congress on Evolutionary Computation (CEC 2009)*. Piscataway, NJ: IEEE Press, pp. 1281–1288.
- Geiger, M. J. (2011). 'Decision Support for Multi-objective Flow Shop Scheduling by the Pareto Iterated Local Search Methodology'. In: *Computers and Industrial Engineering* 61 (3), pp. 805–812.
- Geurts, P., Ernst, D., and Wehenkel, L. (2006). 'Extremely Randomized Trees'. In: *Machine Learning* 63 (1), pp. 3–42.
- Godfrey, K. (1983). *Compartmental Models and their Application*. Academic Press.
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Boston, MA, USA: Addison-Wesley.
- Gorski, J., Klamroth, K., and Ruzika, S. (2011). 'Connectedness of Efficient Solutions in Multiple Objective Combinatorial Optimization'. In: *Journal of Optimization Theory and Applications* 150 (3), pp. 475–497.
- Gorski, J., Paquete, L., and Pedrosa, F. (2012). 'Greedy Algorithms for a Class of Knapsack Problems with Binary Weights'. In: *Computers & Operations Research* 39 (3), pp. 498–511.
- Grimme, C., Kerschke, P., Aspar, P., Trautmann, H., Preuss, M., Deutz, A. H., Wang, H., and Emmerich, M. (2021). 'Peeking Beyond Peaks: Challenges and Research Potentials of Continuous Multimodal Multi-objective Optimization'. In: *Computers & Operations Research* 136, p. 105489.
- Guerreiro, A. P. and Fonseca, C. M. (2018). 'Computing and Updating Hypervolume Contributions in Up to Four Dimensions'. In: *IEEE Transactions on Evolutionary Computation* 22 (3), pp. 449–463.

- Guerreiro, A. P., Fonseca, C. M., and Paquete, L. (2015). 'Greedy Hypervolume Subset Selection in the Three-Objective Case'. In: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2015*. Ed. by S. Silva and A. I. Esparcia-Alcázar. New York, NY: ACM Press, pp. 671–678.
- Hamacher, H. W. and Ruhe, G. (1994). 'On Spanning Tree Problems with Multiple Objectives'. In: *Annals of Operations Research* 52 (4), pp. 209–230.
- Hamadi, Y., Monfroy, E., and Saubion, F., eds. (2012). *Autonomous Search*. Springer.
- Hansen, M. P. and Jazzkiewicz, A. (1998). *Evaluating the Quality of Approximations to the Non-dominated Set*. Tech. rep. IMM-REP-1998-7. Institute of Mathematical Modelling, Technical University of Denmark.
- Hansen, N. (2011). *Injecting External Solutions into CMA-ES*. Research Report CoRR abs/1110.4181. arXiv.
- Hansen, N. and Ostermeier, A. (2001). 'Completely Derandomized Self-Adaptation in Evolution Strategies'. In: *Evolutionary Computation* 9 (2), pp. 159–195.
- Hansen, P. (1979). 'Bicriterion Path Problems'. In: *Multiple Criteria Decision Making: Theory and Applications*. Vol. 177. Lecture Notes in Economics and Mathematical Systems. Berlin, Germany: Springer-Verlag, pp. 109–127.
- Herzel, A., Ruzika, S., and Thielen, C. (2021). 'Approximation Methods for Multiobjective Optimization Problems: A Survey'. In: *INFORMS Journal on Computing* 33 (4), pp. 1284–1299.
- Hoos, H. H. and Stützle, T. (2005). *Stochastic Local Search—Foundations and Applications*. San Francisco, CA: Morgan Kaufmann Publishers.
- Horn, J., Goldberg, D., and Deb, K. (1994). 'Long Path Problems'. In: *Parallel Problem Solving from Nature, PPSN III*. Vol. 866. Lecture Notes in Computer Science. Springer, Heidelberg, Germany, pp. 149–158.
- Huband, S., Hingston, P., Barone, L., and While, L. (2006). 'A Review of Multiobjective Test Problems and a Scalable Test Problem Toolkit'. In: *IEEE Transactions on Evolutionary Computation* 10 (5), pp. 477–506.
- Humeau, J., Liefoghe, A., Talbi, E.-G., and Verel, S. (2013). 'ParadisE0-M0: From Fitness Landscape Analysis to Efficient Local Search Algorithms'. In: *Journal of Heuristics* 19 (6), pp. 881–915.
- Hussein, R. and Deb, K. (2016). 'A Generative Kriging Surrogate Model for Constrained and Unconstrained Multi-Objective Optimization'. In: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2016*. Ed. by T. Friedrich, F. Neumann, and A. M. Sutton. Denver, Colorado, USA: ACM Press, pp. 573–580.
- Hutter, F., Hoos, H. H., and Leyton-Brown, K. (2011). 'Sequential Model-Based Optimization for General Algorithm Configuration'. In: *Learning and Intelligent Optimization, LION 5*. Ed. by C. A. Coello Coello. Vol. 6683. Lecture Notes in Computer Science. Springer, Heidelberg, Germany, pp. 507–523.
- Hutter, F., Hoos, H. H., Leyton-Brown, K., and Stützle, T. (2009). 'ParamILS: An Automatic Algorithm Configuration Framework'. In: *Journal of Artificial Intelligence Research* 36, pp. 267–306.
- Hutter, F., Xu, L., Hoos, H. H., and Leyton-Brown, K. (2014). 'Algorithm Runtime Prediction: Methods & evaluation'. In: *Artificial Intelligence* 206, pp. 79–111.
- Igel, C., Hansen, N., and Roth, S. (2007). 'Covariance Matrix Adaptation for Multi-objective Optimization'. In: *Evolutionary Computation* 15 (1), pp. 1–28.
- Ishibuchi, H., Akedo, N., and Nojima, Y. (2013). 'A Study on the Specification of a Scalarizing Function in MOEA/D for Many-Objective Knapsack Problems'. In: *Learning and Intelligent Optimization, LION 7*. Ed. by P. M. Pardalos and G. Nicosia. Vol. 7997. Lecture Notes in Computer Science. Catania, Italy: Springer, Heidelberg, Germany, pp. 231–246.
- Ishibuchi, H., Imada, R., Setoguchi, Y., and Nojima, Y. (2017). 'Reference Point Specification in Hypervolume Calculation for Fair Comparison and Efficient Search'. In: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2017*. Ed. by P. A. N. Bosman. Berlin, Germany: ACM Press, pp. 585–592.

- Ishibuchi, H., Masuda, H., Tanigaki, Y., and Nojima, Y. (2015). 'Modified Distance Calculation in Generational Distance and Inverted Generational Distance'. In: *Evolutionary Multi-criterion Optimization, EMO 2015 Part I*. Ed. by A. Gaspar-Cunha, C. H. Antunes, and C. A. Coello Coello. Vol. 9018. Lecture Notes in Computer Science. Springer, Heidelberg, Germany, pp. 110–125.
- Ito, T., Aguirre, H. E., Tanaka, K., Liefoghe, A., Derbel, B., and Verel, S. (2019). 'Estimating Relevance of Variables for Effective Recombination'. In: *Evolutionary Multi-criterion Optimization, EMO 2019*. Ed. by K. Deb, E. D. Goodman, C. A. Coello Coello, K. Klamroth, K. Miettinen, S. Mostaghim, and P. Reed. Vol. 11411. Lecture Notes in Computer Science. East Lansing, MI, USA: Springer International Publishing, pp. 411–423.
- Jesus, A. D., Liefoghe, A., Derbel, B., and Paquete, L. (2020). 'Algorithm Selection of Anytime Algorithms'. In: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2020*. Ed. by C. A. Coello Coello. Cancún, Mexico: ACM Press, pp. 850–858.
- Jesus, A. D., Paquete, L., Derbel, B., and Liefoghe, A. (2021a). 'On the Design and Anytime Performance of Indicator-based Branch and Bound for Multi-objective Combinatorial Optimization'. In: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2021*. Ed. by F. Chicano and K. Krawiec. Lille, France: ACM Press, pp. 234–242.
- Jesus, A. D., Paquete, L., and Liefoghe, A. (2021b). 'A Model of Anytime Algorithm Performance for Bi-objective Optimization'. In: *Journal of Global Optimization* 79 (2), pp. 329–350.
- Jiang, S., Ong, Y. S., Zhang, J., and Feng, L. (2014). 'Consistencies and Contradictions of Performance Metrics in Multiobjective Optimization'. In: *IEEE Transactions on Cybernetics* 44 (12), pp. 2391–2404.
- Jin, Y. (2011). 'Surrogate-assisted Evolutionary Computation: Recent Advances and Future Challenges'. In: *Swarm and Evolutionary Computation* 1 (2), pp. 61–70.
- Jones, D. R., Schonlau, M., and Welch, W. J. (1998). 'Efficient Global Optimization of Expensive Black-Box Functions'. In: *Journal of Global Optimization* 13 (4), pp. 455–492.
- Jones, T. and Forrest, S. (1995). 'Fitness Distance Correlation as a Measure of Problem Difficulty for Genetic Algorithms'. In: *6th International Conference on Genetic Algorithms, ICGA*. Ed. by L. J. Eshelman. Morgan Kaufmann Publishers, San Francisco, CA, pp. 184–192.
- Kadioglu, S., Malitsky, Y., Sellmann, M., and Tierney, K. (2010). 'ISAC: Instance-Specific Algorithm Configuration'. In: *Proceedings of the 19th European Conference on Artificial Intelligence, ECAI 2010*. Ed. by H. Coelho, R. Studer, and M. Wooldridge. IOS Press, pp. 751–756.
- Kauffman, S. A. (1993). *The Origins of Order*. Oxford University Press.
- Keijzer, M., Merelo, J.-J., Romero, G., and Schoenauer, M. (2001). 'Evolving Objects: A General Purpose Evolutionary Computation Library'. In: *5th International Conference on Artificial Evolution (EA 2001)*. Le Creusot, France, pp. 231–244.
- Kendall, M. G. (1948). *Rank Correlation Methods*. London: Griffin.
- Kerschke, P., Hoos, H. H., Neumann, F., and Trautmann, H. (2019). 'Automated Algorithm Selection: Survey and Perspectives'. In: *Evolutionary Computation* 27 (1), pp. 3–45.
- Kerschke, P. and Trautmann, H. (2019). 'Automated Algorithm Selection on Continuous Black-Box Problems by Combining Exploratory Landscape Analysis and Machine Learning'. In: *Evolutionary Computation* 27 (1), pp. 99–127.
- Kerschke, P., Wang, H., Preuss, M., Grimme, C., Deutz, A. H., Trautmann, H., and Emmerich, M. T. M. (2016). 'Towards Analyzing Multimodality of Continuous Multiobjective Landscapes'. In: *Parallel Problem Solving from Nature, PPSN XIV*. Ed. by J. Handl, E. Hart, P. R. Lewis, M. López-Ibáñez, G. Ochoa, and B. Paechter. Vol. 9921. Lecture Notes in Computer Science. Springer, Heidelberg, Germany, pp. 962–972.

- Knowles, J. D. (2002). 'Local-Search and Hybrid Evolutionary Algorithms for Pareto Optimization'. PhD thesis. University of Reading, UK.
- (2006). 'ParEGO: A Hybrid Algorithm with On-line Landscape Approximation for Expensive Multiobjective Optimization Problems'. In: *IEEE Transactions on Evolutionary Computation* 10 (1), pp. 50–66.
- Knowles, J. D. and Corne, D. (2002). 'On Metrics for Comparing Non-dominated Sets'. In: *Proceedings of the 2002 Congress on Evolutionary Computation (CEC'02)*. Piscataway, NJ: IEEE Press, pp. 711–716.
- (2003a). 'Instance Generators and Test Suites for the Multiobjective Quadratic Assignment Problem'. In: *Evolutionary Multi-criterion Optimization, EMO 2003*. Ed. by C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, and L. Thiele. Vol. 2632. Lecture Notes in Computer Science. Springer, Heidelberg, Germany, pp. 295–310.
 - (2003b). 'Properties of an Adaptive Archiving Algorithm for Storing Nondominated Vectors'. In: *IEEE Transactions on Evolutionary Computation* 7 (2), pp. 100–116.
 - (2007). 'Quantifying the Effects of Objective Space Dimension in Evolutionary Multiobjective Optimization'. In: *Evolutionary Multi-criterion Optimization, EMO 2007*. Ed. by S. Obayashi et al. Vol. 4403. Lecture Notes in Computer Science. Matsushima, Japan: Springer, Heidelberg, Germany, pp. 757–771.
- Knowles, J. D., Thiele, L., and Zitzler, E. (2006). *A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers*. TIK-Report 214. Revised version. Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zürich, Switzerland.
- Kuhn, T., M.Fonseca, C., Paquete, L., S.Ruzika, Duarte, M., and Figueira, J. R. (2016). 'Hypervolume Subset Selection in Two Dimensions: Formulations and Algorithms'. In: *Evolutionary Computation* 24 (3), pp. 411–425.
- Laumanns, M., Thiele, L., and Zitzler, E. (2004a). 'Running Time Analysis of Evolutionary Algorithms on a Simplified Multiobjective Knapsack Problem'. In: *Natural Computing* 3 (1), pp. 37–51.
- (2004b). 'Running Time Analysis of Multiobjective Evolutionary Algorithms on Pseudo-Boolean Functions'. In: *IEEE Transactions on Evolutionary Computation* 8 (2), pp. 170–182.
- Lavinas, Y. C., Aranha, C., and Ladeira, M. (2019). 'Improving Resource Allocation in MOEA/D with Decision-Space Diversity Metrics'. In: *8th International Conference on Theory and Practice of Natural Computing (TPNC 2019)*. Ed. by C. Martín-Vide, G. T. Pond, and M. A. Vega-Rodríguez. Vol. 11934. Lecture Notes in Computer Science. Kingston, ON, Canada: Springer, pp. 134–146.
- Lavinas, Y., Aranha, C., Ladeira, M., and Campelo, F. (2020). 'MOEA/D with Random Partial Update Strategy'. In: *Proceedings of the 2020 Congress on Evolutionary Computation (CEC 2020)*. Glasgow, United Kingdom: IEEE Press, pp. 1–8.
- Leung, Y., Gao, Y., and Xu, Z.-B. (1997). 'Degree of Population Diversity – A Perspective on Premature Convergence in Genetic Algorithms and its Markov Chain Analysis'. In: *IEEE Transactions on Neural Networks* 8 (5), pp. 1165–1176.
- Li, K. (2021). *Decomposition Multi-Objective Evolutionary Optimization: From State-of-the-Art to Future Opportunities*. Research Report CoRR abs/2108.09588. arXiv.
- Li, M. and Yao, X. (2019). 'Quality Evaluation of Solution Sets in Multiobjective Optimisation: A Survey'. In: *ACM Computing Surveys* 52 (2), p. 26.
- Liaw, A. and Wiener, M. (2002). 'Classification and Regression by randomForest'. In: *R News* 2 (3), pp. 18–22.
- Liefoghe, A., Daolio, F., Verel, S., Derbel, B., Aguirre, H., and Tanaka, K. (2020a). 'Landscape-Aware Performance Prediction for Evolutionary Multi-objective Optimization'. In: *IEEE Transactions on Evolutionary Computation* 24 (6), pp. 1063–1077.

- Liefooghe, A. and Derbel, B. (2016). 'A Correlation Analysis of Set Quality Indicator Values in Multiobjective Optimization'. In: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2016*. Ed. by T. Friedrich, F. Neumann, and A. M. Sutton. Denver, Colorado, USA: ACM Press, pp. 581–588.
- Liefooghe, A., Derbel, B., Verel, S., Aguirre, H., and Tanaka, K. (2017a). 'A Fitness Landscape Analysis of Pareto Local Search on Bi-objective Permutation Flowshop Scheduling Problems'. In: *Evolutionary Multi-criterion Optimization, EMO 2017*. Ed. by H. Trautmann, G. Rudolph, K. Klamroth, O. Schütze, M. M. Wiecek, Y. Jin, and C. Grimme. Lecture Notes in Computer Science. Münster, Germany: Springer International Publishing, pp. 422–437.
- (2017b). 'Towards Landscape-Aware Automatic Algorithm Configuration: Preliminary Experiments on Neutral and Rugged Landscapes'. In: *Proceedings of EvoCOP 2017 – 17th European Conference on Evolutionary Computation in Combinatorial Optimization*. Ed. by B. Hu and M. López-Ibáñez. Vol. 10197. Lecture Notes in Computer Science. Amsterdam, The Netherlands: Springer, Heidelberg, Germany, pp. 215–232.
- Liefooghe, A., Derbel, B., Verel, S., López-Ibáñez, M., Aguirre, H. E., and Tanaka, K. (2018a). 'On Pareto Local Optimal Solutions Networks'. In: *Parallel Problem Solving from Nature - PPSN XV*. Ed. by A. Auger, C. M. Fonseca, N. Lourenço, P. Machado, L. Paquete, and D. Whitley. Vol. 11102. Lecture Notes in Computer Science. Springer, Cham, pp. 232–244.
- Liefooghe, A., Humeau, J., Mesmoudi, S., Jourdan, L., and Talbi, E.-G. (2012). 'On Dominance-based Multiobjective Local Search: Design, Implementation and Experimental Analysis on Scheduling and Traveling Salesman Problems'. In: *Journal of Heuristics* 18 (2), pp. 317–352.
- Liefooghe, A., Jourdan, L., and Talbi, E.-G. (2011). 'A Software Framework Based on a Conceptual Unified Model for Evolutionary Multiobjective Optimization: Paradise0-M0E0'. In: *European Journal of Operational Research* 209 (2), pp. 104–112.
- Liefooghe, A., López-Ibáñez, M., Paquete, L., and Verel, S. (2018b). 'Dominance, Epsilon, and Hypervolume Local Optimal Sets in Multi-objective Optimization, and How to Tell the Difference'. In: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2018*. Ed. by H. E. Aguirre and K. Takadama. New York, NY: ACM Press, pp. 324–331.
- Liefooghe, A., Paquete, L., and Figueira, J. R. (2013a). 'On Local Search for Bi-objective Knapsack Problems'. In: *Evolutionary Computation* 21 (1), pp. 179–196.
- Liefooghe, A., Verel, S., Aguirre, H., and Tanaka, K. (2013b). 'What Makes an Instance Difficult for Black-box 0–1 Evolutionary Multiobjective Optimizers?' In: *Artificial Evolution (EA 2013)*. Vol. 8752. Lecture Notes in Computer Science. Bordeaux, France, pp. 3–15.
- Liefooghe, A., Verel, S., Daolio, F., Aguirre, H., and Tanaka, K. (2015a). 'A Feature-Based Performance Analysis in Evolutionary Multiobjective Optimization'. In: *Evolutionary Multi-criterion Optimization, EMO 2015 Part II*. Ed. by A. Gaspar-Cunha, C. H. Antunes, and C. A. Coello Coello. Vol. 9019. Lecture Notes in Computer Science. Guimarães, Portugal: Springer, Heidelberg, Germany, pp. 95–109.
- Liefooghe, A., Verel, S., Derbel, B., Aguirre, H., and Tanaka, K. (2020b). 'Dominance, Indicator and Decomposition Based Search for Multi-objective QAP: Landscape Analysis and Automated Algorithm Selection'. In: *Parallel Problem Solving from Nature - PPSN XVI*. Ed. by T. Bäck, M. Preuss, A. Deutz, H. Wang, C. Doerr, M. Emmerich, and H. Trautmann. Lecture Notes in Computer Science. Leiden, The Netherlands: Springer, Cham, pp. 33–47.
- Liefooghe, A., Verel, S., and Hao, J.-K. (2014). 'A Hybrid Metaheuristic for Multiobjective Unconstrained Binary Quadratic Programming'. In: *Applied Soft Computing* 16, pp. 10–19.
- Liefooghe, A., Verel, S., Lacroix, B., Zavoianu, A., and McCall, J. A. W. (2021). 'Landscape Features and Automated Algorithm Selection for Multi-objective Interpolated Continuous Optimisation Problems'.

- In: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2021*. Ed. by F. Chicano and K. Krawiec. Lille, France: ACM, pp. 421–429.
- Liefooghe, A., Verel, S., Paquete, L., and Hao, J.-K. (2015b). ‘Experiments on Local Search for Bi-objective Unconstrained Binary Quadratic Programming’. In: *Evolutionary Multi-criterion Optimization, EMO 2015 Part I*. Ed. by A. Gaspar-Cunha, C. H. Antunes, and C. A. Coello Coello. Vol. 9018. Lecture Notes in Computer Science. Guimarães, Portugal: Springer, Heidelberg, Germany, pp. 171–186.
- Liu, H. L., Gu, F., and Zhang, Q. (2014). ‘Decomposition of a Multiobjective Optimization Problem Into a Number of Simple Multiobjective Subproblems’. In: *IEEE Transactions on Evolutionary Computation* 18 (3), pp. 450–455.
- López-Ibáñez, M., Dubois-Lacoste, J., Pérez Cáceres, L., Stützle, T., and Birattari, M. (2016). ‘The i race Package: Iterated Racing for Automatic Algorithm Configuration’. In: *Operations Research Perspectives* 3, pp. 43–58.
- López-Ibáñez, M., Liefooghe, A., and Verel, S. (2014). ‘Local Optimal Sets and Bounded Archiving on Multi-objective NK-Landscapes with Correlated Objectives’. In: *Parallel Problem Solving from Nature, PPSN XIII*. Ed. by T. Bartz-Beielstein, J. Branke, B. Filipič, and J. Smith. Vol. 8672. Lecture Notes in Computer Science. Springer, Heidelberg, Germany, pp. 621–630.
- Louppe, G., Wehenkel, L., Sutera, A., and Geurts, P. (2013). ‘Understanding Variable Importances in Forests of Randomized Trees’. In: *Advances in Neural Information Processing Systems (NIPS 26)*. Curran Associates, Red Hook, NY, pp. 431–439.
- Malan, K. M. and Engelbrecht, A. P. (2014). ‘Fitness Landscape Analysis for Metaheuristic Performance Prediction’. In: *Recent Advances in the Theory and Application of Fitness Landscapes*. Springer, pp. 103–132.
- Malan, K. M. and Moser, I. (2019). ‘Constraint Handling Guided by Landscape Analysis in Combinatorial and Continuous Search Spaces’. In: *Evolutionary Computation* 27 (2), pp. 267–289.
- Mambrini, A. and Izzo, D. (2014). ‘PaDe: A Parallel Algorithm Based on the MOEA/D Framework and the Island Model’. In: *Parallel Problem Solving from Nature, PPSN XIII*. Ed. by T. Bartz-Beielstein, J. Branke, B. Filipič, and J. Smith. Vol. 8672. Lecture Notes in Computer Science. Springer, Heidelberg, Germany, pp. 711–720.
- Marquet, G., Derbel, B., Liefooghe, A., and Talbi, E. (2014). ‘Shake Them All! - Rethinking Selection and Replacement in MOEA/D’. In: *Parallel Problem Solving from Nature, PPSN XIII*. Ed. by T. Bartz-Beielstein, J. Branke, B. Filipič, and J. Smith. Vol. 8672. Lecture Notes in Computer Science. Ljubljana, Slovenia: Springer, Heidelberg, Germany, pp. 641–651.
- Mersmann, O., Bischl, B., Trautmann, H., Wagner, M., Bossek, J., and Neumann, F. (2013). ‘A Novel Feature-based Approach to Characterize Algorithm Performance for the Traveling Salesperson Problem’. In: *Annals of Mathematics and Artificial Intelligence* 69, pp. 151–182.
- Merz, P. (2004). ‘Advanced Fitness Landscape Analysis and the Performance of Memetic Algorithms’. In: *Evolutionary Computation* 12 (3), pp. 303–325.
- Miettinen, K. (1999). *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, p. 298.
- Monzón, H., Aguirre, H., Verel, S., Liefooghe, A., Derbel, B., and Tanaka, K. (2017). ‘Closed State Model for Understanding the Dynamics of MOEAs’. In: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2017*. Ed. by P. A. N. Bosman. Berlin, Germany: ACM Press, pp. 609–616.
- (2020). ‘Dynamic Compartmental Models for Large Multi-objective Landscapes and Performance Estimation’. In: *Proceedings of EvoCOP 2020 – 20th European Conference on Evolutionary Computation in Combinatorial Optimization*. Ed. by L. Paquete and C. Zarges. Vol. 12102. Lecture Notes in Computer Science. Seville, Spain: Springer, Berlin, pp. 99–113.

- Moraglio, A. and Kattan, A. (2011). 'Geometric Generalisation of Surrogate Model Based Optimization to Combinatorial Spaces'. In: *Proceedings of EvoCOP 2011 – 11th European Conference on Evolutionary Computation in Combinatorial Optimization*. Ed. by P. Merz and J.-K. Hao. Vol. 6622. Lecture Notes in Computer Science. Springer, Heidelberg, Germany, pp. 142–154.
- Moser, I., Gheorghita, M., and Aleti, A. (2017). 'Identifying Features of Fitness Landscapes and Relating them to Problem Difficulty'. In: *Evolutionary Computation 25 (3)*, pp. 407–437.
- Murata, T., Ishibuchi, H., and Gen, M. (2001). 'Specification of Genetic Search Directions in Cellular Multi-objective Genetic Algorithms'. In: *Evolutionary Multi-criterion Optimization, EMO 2001*. Ed. by E. Zitzler, K. Deb, L. Thiele, C. A. Coello Coello, and D. Corne. Vol. 1993. Lecture Notes in Computer Science. Zurich, Switzerland: Springer, Heidelberg, Germany, pp. 82–95.
- Murtagh, F. and Legendre, P. (2014). 'Ward's Hierarchical Agglomerative Clustering Method: Which Algorithms Implement Ward's Criterion?' In: *Journal of Classification 31 (3)*, pp. 274–295.
- Nebro, A. J. and Durillo, J. J. (2010). 'A Study of the Parallelization of the Multi-Objective Metaheuristic MOEA/D'. In: *Learning and Intelligent Optimization, LION 4*. Ed. by C. Blum and R. Battiti. Vol. 6073. Lecture Notes in Computer Science. Springer, Heidelberg, Germany, pp. 303–317.
- Ochoa, G., Tomassini, M., Verel, S., and Darabos, C. (2008). 'A Study of NK Landscapes' Basins and Local Optima Networks'. In: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2008*. Ed. by C. Ryan. New York, NY: ACM Press, pp. 555–562.
- Ochoa, G., Veerapen, N., Daolio, F., and Tomassini, M. (2017). 'Understanding Phase Transitions with Local Optima Networks: Number Partitioning as a Case Study'. In: *Proceedings of EvoCOP 2017 – 17th European Conference on Evolutionary Computation in Combinatorial Optimization*. Ed. by B. Hu and M. López-Ibáñez. Vol. 10197. Lecture Notes in Computer Science. Amsterdam, The Netherlands: Springer, Heidelberg, Germany, pp. 233–248.
- Ochoa, G., Verel, S., Daolio, F., and Tomassini, M. (2014). 'Local Optima Networks: A New Model of Combinatorial Fitness Landscapes'. In: *Recent Advances in the Theory and Application of Fitness Landscapes*. Springer, pp. 233–262.
- Paquete, L., Camacho, C., and Figueira, J. R. (2008). *A Two-phase Heuristic for the Biobjective 0/1 Knapsack Problem*. Tech. rep. Instituto Superior Técnico, Technical University of Lisbon, Portugal.
- Paquete, L., Schiavinotto, T., and Stützle, T. (2007). 'On Local Optima in Multiobjective Combinatorial Optimization Problems'. In: *Annals of Operations Research 156*, pp. 83–97.
- Paquete, L. and Stützle, T. (2009). 'Clusters of Non-dominated Solutions in Multiobjective Combinatorial Optimization: An Experimental Analysis'. In: *Multiobjective Programming and Goal Programming: Theoretical Results and Practical Applications*. Ed. by V. Barichard, M. Ehrgott, X. Gandibleux, and V. T'Kindt. Vol. 618. Lecture Notes in Economics and Mathematical Systems. Springer, Berlin, pp. 69–77.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). 'Scikit-learn: Machine Learning in Python'. In: *Journal of Machine Learning Research 12*, pp. 2825–2830.
- Pelikan, M., Goldberg, D. E., and Cantú-Paz, E. (2000). 'Linkage Problem, Distribution Estimation, and Bayesian Networks'. In: *Evolutionary Computation 8 (3)*, pp. 311–340.
- Ponweiser, W., Wagner, T., Biermann, D., and Vincze, M. (2008). 'Multiobjective Optimization on a Limited Budget of Evaluations Using Model-assisted S-Metric Selection'. In: *Parallel Problem Solving from Nature, PPSN X*. Ed. by G. Rudolph et al. Vol. 5199. Lecture Notes in Computer Science. Dortmund, Germany: Springer, Heidelberg, Germany, pp. 784–794.
- Preuss, M. (2015). *Multimodal Optimization by Means of Evolutionary Algorithms*. Natural Computing Series. Springer.

- Pruvost, G., Derbel, B., Liefoghe, A., Li, K., and Zhang, Q. (2020a). 'On the Combined Impact of Population Size and Sub-problem Selection in MOEA/D'. In: *Proceedings of EvoCOP 2020 – 20th European Conference on Evolutionary Computation in Combinatorial Optimization*. Ed. by L. Paquete and C. Zarges. Vol. 12102. Lecture Notes in Computer Science. Seville, Spain: Springer, Berlin, pp. 131–147.
- Pruvost, G., Derbel, B., Liefoghe, A., Verel, S., and Zhang, Q. (2020b). 'Surrogate-assisted Multi-objective Combinatorial Optimization based on Decomposition and Walsh Basis'. In: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2020*. Ed. by C. A. Coello Coello. Cancún, Mexico: ACM Press, pp. 542–550.
- (2021). *A Modular Surrogate-assisted Framework for Expensive Multiobjective Combinatorial Optimization*. Research Report hal-03380316. Inria.
- Przybylski, A. and Gandibleux, X. (2017). 'Multi-objective Branch and Bound'. In: *European Journal of Operational Research* 260 (3), pp. 856–872.
- R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria.
- Rice, J. R. (1976). 'The Algorithm Selection Problem'. In: *Advances in Computers* 15, pp. 65–118.
- Richter, H. and Engelbrecht, A. P., eds. (2014). *Recent Advances in the Theory and Application of Fitness Landscapes*. Emergence, Complexity and Computation. Springer.
- Rifki, O., Chiabaut, N., and Solnon, C. (2020). 'On the Impact of Spatio-temporal Granularity of Traffic Conditions on the Quality of Pickup and Delivery Optimal Tours'. In: *Transportation Research Part E: Logistics and Transportation Review* 142, p. 102085.
- Sagawa, M., Aguirre, H. E., Daolio, F., Liefoghe, A., Derbel, B., Verel, S., and Tanaka, K. (2016). 'Learning Variable Importance to Guide Recombination'. In: *Symposium Series on Computational Intelligence (SSCI 2016)*. Athens, Greece: IEEE, pp. 1–7.
- Santana, R., Cosson, R., Liefoghe, A., and Derbel, B. (2022). 'Multi-objective NK Landscapes with Heterogeneous Objectives'. In: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2022*. Boston, MA, USA: ACM Press.
- Sato, H. (2015). 'Analysis of Inverted PBI and Comparison with other Scalarizing Functions in Decomposition based MOEAs'. In: *Journal of Heuristics* 21 (6), pp. 819–849.
- Schütze, O., Esquivel, X., Lara, A., and Coello Coello, C. A. (2012). 'Using the Averaged Hausdorff Distance as a Performance Measure in Evolutionary Multiobjective Optimization'. In: *IEEE Transactions on Evolutionary Computation* 16 (4), pp. 504–522.
- Shang, K., Ishibuchi, H., and Ni, X. (2020). 'R2-Based Hypervolume Contribution Approximation'. In: *IEEE Transactions on Evolutionary Computation* 24 (1), pp. 185–192.
- Shi, J., Zhang, Q., Derbel, B., and Liefoghe, A. (2017a). 'A Parallel Tabu Search for the Unconstrained Binary Quadratic Programming Problem'. In: *Proceedings of the 2017 Congress on Evolutionary Computation (CEC 2017)*. San Sebastián, Spain: IEEE Press, pp. 557–564.
- Shi, J., Zhang, Q., Derbel, B., Liefoghe, A., and Sun, J. (2018). 'Parallel Pareto Local Search Revisited: First Experimental Results on Bi-objective UBQP'. In: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2018*. Ed. by H. E. Aguirre and K. Takadama. Kyoto, Japan: ACM Press, pp. 753–760.
- Shi, J., Zhang, Q., Derbel, B., Liefoghe, A., and Verel, S. (2017b). 'Using Parallel Strategies to Speed up Pareto Local Search'. In: *11th International Conference on Simulated Evolution and Learning (SEAL 2017)*. Ed. by Y. Shi, K. C. Tan, M. Zhang, K. Tang, X. Li, Q. Zhang, Y. Tan, M. Middendorf, and Y. Jin. Vol. 10593. Lecture Notes in Computer Science. Shenzhen, China: Springer, Heidelberg, Germany, pp. 62–74.
- Smith, T., Husbands, P., and O'Shea, M. (2002). 'Fitness Landscapes and Evolvability'. In: *Evolutionary Computation* 10 (1), pp. 1–34.

- Smith-Miles, K. (2008). 'Cross-disciplinary Perspectives on Meta-learning for Algorithm Selection'. In: *ACM Computing Surveys* 41 (1), pp. 1–25.
- Smith-Miles, K. A., Baatar, D., Wreford, B., and Lewis, R. (2014). 'Towards Objective Measures of Algorithm Performance Across Instance Space'. In: *Computers & Operations Research* 45, pp. 12–24.
- Talbi, E.-G. (2009). *Metaheuristics: From Design to Implementation*. Wiley.
- Talbi, E.-G., Mostaghim, S., Okabe, T., Ishibuchi, H., Rudolph, G., and Coello Coello, C. A. (2008). 'Parallel Approaches for Multiobjective Optimization'. In: *Multiobjective Optimization – Interactive and Evolutionary Approaches*. Ed. by J. Branke, K. Deb, K. Miettinen, and R. Slowinski. Vol. 5252. Lecture Notes in Computer Science. Springer, pp. 349–372.
- Therneau, T. and Atkinson, B. (2018). *rpart: Recursive Partitioning and Regression Trees*. R package version 4.1-13.
- Trivedi, A., Srinivasan, D., Sanyal, K., and Ghosh, A. (2017). 'A Survey of Multiobjective Evolutionary Algorithms Based on Decomposition'. In: *IEEE Transactions on Evolutionary Computation* 21 (3), pp. 440–462.
- Van Veldhuizen, D. A. and Lamont, G. B. (1998). 'Evolutionary Computation and Convergence to a Pareto Front'. In: *Genetic Programming 1998: Proceedings of the Third Annual Conference, Late Breaking Papers*. Ed. by J. R. Koza. Stanford University, California: Stanford University Bookstore, pp. 221–228.
- Vaz, D., Paquete, L., Fonseca, C. M., Klamroth, K., and Stiglmayr, M. (2015). 'Representation of the non-dominated set in biobjective discrete optimization'. In: *Computers & Operations Research* 63, pp. 172–186.
- Verel, S., Derbel, B., Liefooghe, A., Aguirre, H. E., and Tanaka, K. (2018). 'A Surrogate Model based on Walsh Decomposition for Pseudo-Boolean Functions'. In: *Parallel Problem Solving from Nature - PPSN XV*. Ed. by A. Auger, C. M. Fonseca, N. Lourenço, P. Machado, L. Paquete, and D. Whitley. Vol. 11101. Lecture Notes in Computer Science. Coimbra, Portugal: Springer, Cham, pp. 181–193.
- Verel, S., Liefooghe, A., and Dhaenens, C. (2011a). 'Set-based Multiobjective Fitness Landscapes: A Preliminary Study'. In: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2011*. Ed. by N. Krasnogor and P. L. Lanzi. New York, NY: ACM Press, pp. 769–776.
- Verel, S., Liefooghe, A., Humeau, J., Jourdan, L., and Dhaenens, C. (2011b). 'On the Effect of Connectedness for Biobjective Multiple and Long Path Problems'. In: *Learning and Intelligent Optimization, LION 5*. Ed. by C. A. Coello Coello. Vol. 6683. Lecture Notes in Computer Science. Rome, Italy: Springer, Heidelberg, Germany, pp. 31–45.
- Verel, S., Liefooghe, A., Jourdan, L., and Dhaenens, C. (2013). 'On the Structure of Multiobjective Combinatorial Search Space: MNK-landscapes with Correlated Objectives'. In: *European Journal of Operational Research* 227 (2), pp. 331–342.
- Visée, M., Teghem, J., Pirlot, M., and Ulungu, E. L. (1998). 'Two-phases Method and Branch and Bound Procedures to Solve the Bi-objective Knapsack Problem'. In: *Journal of Global Optimization* 12 (2), pp. 139–155.
- Vodopija, A., Tušar, T., and Filipič, B. (2021). *Characterization of Constrained Continuous Multiobjective Optimization Problems: A Feature Space Perspective*. Research Report CoRR abs/2109.04564. arXiv.
- Wagner, M., Friedrich, T., and Lindauer, M. T. (2017). 'Improving Local Search in a Minimum Vertex cover Solver for Classes of Networks'. In: *Proceedings of the 2017 Congress on Evolutionary Computation (CEC 2017)*. Piscataway, NJ: IEEE Press, pp. 1704–1711.
- Wang, P., Zhu, W., Liu, H., Liao, B., Cai, L., Wei, X., Ren, S., and Yang, J. (2019). 'A New Resource Allocation Strategy based on the Relationship between Subproblems for MOEA/D'. In: *Information Sciences* 501, pp. 337–362.

- Warnes, G. R., Bolker, B., Bonebakker, L., Gentleman, R., Liaw, W. H. A., Lumley, T., Maechler, M., Magnusson, A., Moeller, S., Schwartz, M., and Venables, B. (2016). *gpLots: Various R Programming Tools for Plotting Data*. R package version 3.0.1.
- Weinberger, E. D. (1990). 'Correlated and Uncorrelated Fitness Landscapes and How to Tell the Difference'. In: *Biological Cybernetics* 63 (5), pp. 325–336.
- Wessing, S. and Naujoks, B. (2010). 'Sequential Parameter Optimization for Multi-objective Problems'. In: *Proceedings of the 2010 Congress on Evolutionary Computation (CEC 2010)*. Ed. by H. Ishibuchi et al. Barcelona, Spain: IEEE Press, pp. 1–8.
- Whitley, L. D., Chicano, F., and Goldman, B. W. (2016). 'Gray Box Optimization for NK Landscapes (NK Landscapes and MAX-kSAT)'. In: *Evolutionary Computation* 24 (3), pp. 491–519.
- Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. New York, USA: Springer-Verlag.
- Xu, L., Hoos, H. H., and Leyton-Brown, K. (2010). 'Hydra: Automatically Configuring Algorithms for Portfolio-Based Selection'. In: *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010*. Ed. by M. Fox and D. Poole. AAAI Press.
- Yap, E., Muñoz, M. A., Smith-Miles, K., and Liefoghe, A. (2020). 'Instance Space Analysis of Combinatorial Multi-objective Optimization Problems'. In: *Proceedings of the 2020 Congress on Evolutionary Computation (CEC 2020)*. Glasgow, UK: IEEE Press, pp. 1–8.
- Zaefferer, M., Stork, J., Friese, M., Fischbach, A., Naujoks, B., and Bartz-Beielstein, T. (2014). 'Efficient Global Optimization for Combinatorial Problems'. In: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2014*. Ed. by C. Igel and D. V. Arnold. New York, NY: ACM Press, pp. 871–878.
- Zapotecas-Martínez, S. and Coello Coello, C. A. (2013). 'MOEA/D Assisted by RBF Networks for Expensive Multi-Objective Optimization Problems'. In: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2013*. Ed. by C. Blum and E. Alba. Amsterdam, The Netherlands: ACM Press, pp. 1405–1412.
- Zapotecas-Martínez, S., Derbel, B., Liefoghe, A., Aguirre, H. E., and Tanaka, K. (2015a). 'Geometric Differential Evolution in MOEA/D: A Preliminary Study'. In: *Advances in Artificial Intelligence and Soft Computing (MICAI 2015)*. Ed. by G. Sidorov and S. N. Galicia-Haro. Vol. 9413. Lecture Notes in Computer Science. Cuernavaca, Mexico: Springer, pp. 364–376.
- Zapotecas-Martínez, S., Derbel, B., Liefoghe, A., Brockhoff, D., Aguirre, H. E., and Tanaka, K. (2015b). 'Injecting CMA-ES into MOEA/D'. In: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2015*. Ed. by S. Silva and A. I. Esparcia-Alcázar. Madrid, Spain: ACM Press, pp. 783–790.
- Zhang, Q., Liu, W., Tsang, E., and Virginas, B. (2010). 'Expensive Multiobjective Optimization by MOEA/D With Gaussian Process Model'. In: *IEEE Transactions on Evolutionary Computation* 14 (3), pp. 456–474.
- Zhang, Q. and Li, H. (2007). 'MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition'. In: *IEEE Transactions on Evolutionary Computation* 11 (6), pp. 712–731.
- Zhang, Q., Zhou, A., Zhao, S., Suganthan, P. N., Liu, W., and Tiwari, S. (2009). *Multiobjective Optimization Test Instances for the CEC 2009 Special Session and Competition*. Tech. rep. CES-887. School of Computer Science and Electrical Engineering, University of Essex.
- Zhou, A. and Zhang, Q. (2016). 'Are All the Subproblems Equally Important? Resource Allocation in Decomposition-Based Multiobjective Evolutionary Algorithms'. In: *IEEE Transactions on Evolutionary Computation* 20 (1), pp. 52–64.
- Zhou, Z.-H. (2021). 'Ensemble Learning'. In: *Machine Learning*. Singapore: Springer, pp. 181–210.
- Zilberstein, S. (1996). 'Using Anytime Algorithms in Intelligent Systems'. In: *AI Magazine* 17 (3), pp. 73–83.

- Zitzler, E., Brockhoff, D., and Thiele, L. (2007). 'The Hypervolume Indicator Revisited: On the Design of Pareto-compliant Indicators Via Weighted Integration'. In: *Evolutionary Multi-criterion Optimization, EMO 2007*. Ed. by S. Obayashi et al. Vol. 4403. Lecture Notes in Computer Science. Springer, Heidelberg, Germany, pp. 862–876.
- Zitzler, E., Knowles, J. D., and Thiele, L. (2008). 'Quality Assessment of Pareto Set Approximations'. In: *Multi-objective Optimization: Interactive and Evolutionary Approaches*. Ed. by J. Branke, K. Deb, K. Miettinen, and R. Słowiński. Vol. 5252. Lecture Notes in Computer Science. Springer, Heidelberg, Germany, pp. 373–404.
- Zitzler, E. and Künzli, S. (2004). 'Indicator-based Selection in Multiobjective Search'. In: *Parallel Problem Solving from Nature, PPSN VIII*. Ed. by X. Yao et al. Vol. 3242. Lecture Notes in Computer Science. Springer, Heidelberg, Germany, pp. 832–842.
- Zitzler, E. and Thiele, L. (1998). 'Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study'. In: *Parallel Problem Solving from Nature, PPSN V*. Ed. by A. E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel. Vol. 1498. Lecture Notes in Computer Science. Springer, Heidelberg, Germany, pp. 292–301.
- Zitzler, E., Thiele, L., and Bader, J. (2010). 'On Set-Based Multiobjective Optimization'. In: *IEEE Transactions on Evolutionary Computation* 14 (1), pp. 58–79.
- Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., and Grunert da Fonseca, V. (2003). 'Performance Assessment of Multiobjective Optimizers: an Analysis and Review'. In: *IEEE Transactions on Evolutionary Computation* 7 (2), pp. 117–132.