CHARACTERIZATION OF CONSTRAINED CONTINUOUS MULTIOBJECTIVE OPTIMIZATION PROBLEMS

Aljoša Vodopija

Doctoral Dissertation
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia

Supervisor: Prof. Dr. Bogdan Filipič, Jožef Stefan Institute, Ljubljana, Slovenia

Evaluation Board:
Prof. Dr. Gregor Papa, Chair, Jožef Stefan Institute, Ljubljana, Slovenia
Prof. DDr. Janez Žerovnik, Member, Faculty of Mechanical Engineering, University of Ljubljana, Ljubljana, Slovenia
Prof. Dr. Michael Affenzeller, Member, Heuristic and Evolutionary Algorithms Laboratory, University of Applied Sciences Upper Austria, Hagenberg im Mühlkreis, Austria

MEDNARODNA PODIPLOMSKA ŠOLA JOŽEFA STEFANA
JOŽEF STEFAN INTERNATIONAL POSTGRADUATE SCHOOL

Aljoša Vodopija

CHARACTERIZATION OF CONSTRAINED CONTINUOUS MULTIOBJECTIVE OPTIMIZATION PROBLEMS

Doctoral Dissertation

KARAKTERIZACIJA ZVEZNIH VEČKRITERIJSKIH OPTIMIZACIJSKIH PROBLEMOV Z OMEJITVAMI

Doktorska disertacija

Supervisor: Prof. Dr. Bogdan Filipič

Ljubljana, Slovenia, January 2024

To Ajda, Ema, and Tim

Acknowledgments

First and foremost, I wish to thank my supervisor Prof. Dr. Bogdan Filipič, for the constant support and guidance throughout my doctoral study. I want to extend my sincere thanks to Dr. Tea Tušar for her assistance at every stage of the research and for polishing most of the figures included in the thesis. Many thanks to Jordan Cork who carefully read the thesis and provided valuable grammatical corrections. I would also like to thank the members of the dissertation evaluation board: Prof. Dr. Gregor Papa, Prof. DDr. Janez Žerovnik, and Prof. Dr. Michael Affenzeller, for reading and carefully evaluating my thesis.

My gratitude also extends to all colleagues at the Department of Intelligent Systems, from whom I learned most of what I know about computer science and who made my work at the institute exciting.

I acknowledge the financial support of the Slovenian Research and Innovation Agency, which funded my doctoral study through the Young Researcher program.

Finally, many thanks to my family and friends for supporting me during this journey. Special thanks go to my wife Ajda, daughter Ema, and son Tim. They were always there for me and gave me hope on bad days. Additionally, this work would not be possible without the hard work and sacrifice made by my mum Majda.

Abstract

Despite the large volume of recently published papers in the field of constrained multiobjective optimization, the understanding and characterization of constrained multiobjective optimization problems (CMOPs) for benchmarking multiobjective evolutionary algorithms (MOEAs) and the related constraint handling techniques (CHTs) remain unsatisfactory. Therefore, selecting appropriate CMOPs for benchmarking is challenging and lacks a formal basis. Under such conditions, preparing a robust and well-designed experimental setup for constrained multiobjective optimization becomes a difficult task. An inadequately designed benchmark could result in improper conclusions about CMOP landscapes and MOEA performance. This thesis takes a step towards resolving this issue by examining CMOPs from both the feature and the performance space perspective.

Regarding the feature space perspective, we expand the landscape analysis to constrained multiobjective optimization. Using four exploratory landscape analysis techniques, we propose various landscape features to characterize CMOPs. These features are then employed to compare eight commonly used artificial test suites against a recently proposed suite consisting of real-world problems based on physical models. The experimental results reveal that the artificial test problems do not sufficiently represent certain realistic characteristics, such as strong negative correlation between the objectives and constraints. Furthermore, our results show that all the studied artificial test suites have advantages and limitations, and no "perfect" suite exists. Additionally, the effectiveness of the proposed features at predicting algorithm performance is demonstrated for two MOEAs.

Concerning the performance space perspective, the thesis initially presents a novel performance assessment method designed specifically for constrained multiobjective optimization. This methodology provides a first attempt at measuring performance in approximating the Pareto front and constraint satisfaction simultaneously. Moreover, it proposes an approach to measure the capability of the given optimization problem to differentiate among algorithm performances. Additionally, this approach is used to compare eight commonly used artificial test suites of CMOPs. The experimental results reveal which suites are more effective at distinguishing between three well-known MOEAs.

Finally, we illustrate how the proposed methodology can be used in two real-world scenarios: design of cyclone dust separators and elevator group control. Both problems are characterized with seven landscape features proposed in this thesis, providing insights into their complexity. The analysis shows that the design of elevator group control is more challenging, which is additionally confirmed by a performance space study.

Povzetek

Kljub večjemu številu nedavno objavljenih člankov na področju večkriterijske optimizacije z omejitvami razumevanje in karakterizacija večkriterijskih optimizacijskih problemov z omejitvami ostajata nezadovoljiva. Zato je izbira ustreznih problemov za primerjalno analizo zahtevna in nima formalne osnove. V takšnih okoliščinah je priprava dobro zasnovanega eksperimenta težka naloga, ki ob neustrezni izbiri testnih problemov lahko privede do napačnih zaključkov o problemih in uspešnosti algoritmov. Ta doktorska disertacija predstavlja korak na poti reševanja te težave z analizo večkriterijskih optimizacijskih problemov z omejitvami z vidika prostora značilk in prostora učinkovitosti.

Z vidika prostora značilk razširimo analizo problemske pokrajine na večkriterijsko optimizacijo z omejitvami. Uporabimo štiri tehnike analize pokrajine in predlagamo številne značilke za karakterizacijo večkriterijskih optimizacijskih problemov z omejitvami. Te značilke nato uporabimo za primerjavo med umetnimi nabori testnih problemov in naborom realnih problemov, ki temeljijo na fizikalnih modelih. Eksperimentalni rezultati pokažejo, da umetni testni problemi ne predstavljajo zadostno nekaterih značilnosti realnih problemov, kot je negativna korelacija med kriteriji in omejitvami. Poleg tega naši rezultati razkrivajo, da imajo vsi pregledani umetni nabori testnih problemov svoje pomanjkljivosti in da popoln nabor ne obstaja. Prav tako prikažemo učinkovitost predlaganih značilk pri napovedovanju uspešnosti dveh večkriterijskih optimizacijskih algoritmov.

Z vidika prostora učinkovitosti predlagamo nov pristop za ocenjevanje uspešnosti, ki je bil posebej zasnovan za večkriterijsko optimizacijo z omejitvami. Ta metodologija ponuja prvi poskus sočasnega merjenja uspešnosti pri približevanju Pareto fronti in zadovoljevanju omejitev. Poleg tega predlagamo pristop za merjenje zmožnosti optimizacijskega problema, da razlikuje med uspešnostjo algoritmov. Na podlagi rezultatov analize osmih naborov testnih problemov ugotavljamo, kateri problemi so učinkovitejši pri razlikovanju treh znanih večkriterijskih optimizacijskih algoritmov.

Nazadnje prikažemo, kako se lahko metodologija uporabi na dveh realnih problemih: načrtovanju ciklonskih odstranjevalcev prahu in načrtovanju sistemov za krmiljenje skupin dvigal. Oba problema karakteriziramo s sedmimi značilkami, ki omogočajo vpogled v njuno zahtevnost. Analiza problemske pokrajine pokaže, da je načrtovanje sistemov za krmiljenje skupin dvigal zahtevnejši optimizacijski problem, kar dodatno potrdimo z analizo prostora učinkovitosti.

Contents

List of Figures

List of Tables

Abbreviations

ANOVA	... analysis of variance
CA	... convergence-oriented archive
CDP	... constrained domination principle
C-DTLZ	... constrained Deb, Thiele, Laumanns, and Zitzler problem
CE	... collection efficiency
CF	... constrained function problem
CHT	... constraint handling technique
CMOP	... constrained multiobjective optimization problem
CPU	... central processing unit
CSOP	... constrained single-objective optimization problem
C-TAEA	... constrained two-archive evolutionary algorithm
CTP	... constrained test problem
DA	... diversity-oriented archive
DAS-CMOP	... difficulty adjustable and scalable constrained multiobjective test problem
DBSCAN	... density-based spatial clustering of applications with noise
DC-DTLZ	... constrained Deb, Thiele, Laumanns, and Zitzler problem where constraints act in the decision space
DEMO	... differential evolution for multiobjective optimization
DTLZ	... Deb, Thiele, Laumanns, and Zitzler problem
ECDF	... empirical cumulative distribution function
EGC	... elevator group control
ELA	... exploratory landscape analysis
ERD	... empirical runtime distribution
IGD	... inverted generational distance
LIR-CMOP	... large infeasible region constrained multiobjective optimization problem
MAE	... mean absolute error
MOEA	... multiobjective optimization evolutionary algorithm
MOEA/D	... multiobjective evolutionary algorithm based on decomposition
MOEA/D-DE	... multiobjective evolutionary algorithm based on decomposition and differential evolution
MOEA/D-IEpsilon	... multiobjective evolutionary algorithm based on decomposition and improved epsilon constraint method
MOP	... multiobjective optimization problem
MSE	... mean square error
MW	... Ma and Wang problem
NCTP	... new constrained test problem
NSGA-II	... nondominated sorting genetic algorithm II
NSGA-III	... nondominated sorting genetic algorithm III
PD	... pressure drop

RCM	...	real-world constrained multiobjective problem
RS	...	random search
SPO	...	sequential parameter optimization
t-SNE	...	t-distributed stochastic neighbor embedding

Symbols

x	... decision vector
f_m	... objective function
f	... objective function vector
g_i	... inequality constraint function
h_i	... equality constraint function
v_i	... constraint violation
v	... overall constraint violation
d	... (Euclidean) distance metric
S	... search space
D	... dimensionality of the decision space
M	... dimensionality of the objective space
J	... number of inequality constraint functions
K	... number of equality constraint functions
$\preceq$	... Pareto dominance relation
x^*	... (local) Pareto-optimal solution
S_o	... set of all Pareto-optimal solutions
S_l	... set of all local Pareto-optimal solutions
P_o	... Pareto front
F	... feasible region
$\mathcal{F}$	... feasible component
F_m	... set of all minimum-violation solutions
F_l	... set of all local minimum-violation solutions
$\mathcal{M}$	... local minimum-violation component
z^ide	... ideal objective vector
z^nad	... nadir objective vector
g^te	... Tchebycheff function
A^T	... archive of solutions
τ	... target precision value
$T_a(\tau)$	... runtime needed for an algorithm a to reach τ
$\mathcal{D}(A^T)$	... set of all solutions from A^T dominating the reference point
$\mathcal{B}(\mathcal{M})$	... basin of attraction of $\mathcal{M}$
X_S	... sample for space-filling design features
X_R	... sample for random walk features
X_A	... sample for adaptive walk features
H	... Shannon entropy
$N_\mathcal{F}$	... number of feasible components
$\mathcal{F}_\mathrm{min}$	... smallest feasible component
$\mathcal{F}_\mathrm{med}$	... median feasible component
$\mathcal{F}_\mathrm{max}$	... largest feasible component
$\mathcal{O}(\mathcal{F}_\mathrm{max})$	... proportion of Pareto-optimal solutions in $\mathcal{F}_\mathrm{max}$
$\mathcal{F}_\mathrm{opt}$	... size of the “optimal” feasible component

ρ_{F} ... feasibility ratio
$\rho_{\min}$ ... minimum correlation
$\rho_{\max}$ ... maximum correlation
$\rho_{\partial S_o}$ ... proportion of boundary Pareto-optimal solutions
ε_s ... settling sensitivity
M_0 ... initial partial information
$(\rho_{\partial F})_{\min}$ ... minimum ratio of feasible boundary crossings
$(\rho_{\partial F})_{\mathrm{med}}$ ... median ratio of feasible boundary crossings
$(\rho_{\partial F})_{\max}$ ... maximum ratio of feasible boundary crossings
$N_{\mathcal{B}}$ ... number of basins
$\mathcal{B}_{\min}$ ... smallest basin
$\mathcal{B}_{\mathrm{med}}$ ... median basin
$\mathcal{B}_{\max}$ ... largest basin
$(\mathcal{B}_{\mathrm{F}})_{\min}$ ... smallest feasible basin
$(\mathcal{B}_{\mathrm{F}})_{\mathrm{med}}$ ... median feasible basin
$(\mathcal{B}_{\mathrm{F}})_{\max}$ ... largest feasible basin
$\cup\mathcal{B}_{\mathrm{F}}$ ... proportion of feasible basins
$v(\mathcal{B})_{\mathrm{med}}$ ... median constraint violation over all basins
$v(\mathcal{B})_{\max}$ ... maximum constraint violation of all basins
$v(\mathcal{B}_{\max})$ ... constraint violation of $\mathcal{B}_{\max}$
$\mathcal{O}(\mathcal{B}_{\max})$ ... proportion of Pareto-optimal solutions in $\mathcal{B}_{\max}$
$\mathcal{B}_{\mathrm{opt}}$ ... size of the "optimal" basin
$H_{\max}$ ... maximum information content
I ... quality indicator
I^{HV} ... hypervolume indicator
I^{SOP} ... quality indicator for SOPs
I^{CSOP} ... quality indicator for CSOPs
I^{MOP} ... quality indicator for MOPs
I^{CMOP} ... quality indicator for CMOPs
Δ ... distance metric between two ERDs
n_f ... number of solution evaluations
n_{p} ... population size
n_{g} ... number of generations
p_{c} ... crossover probability
p_{m} ... mutation probability
s_f ... scaling factor
n_τ ... number of target precision values
D_a ... cyclone diameter
D_t ... outlet pipe diameter
H ... total cyclone height
H_t ... outlet pipe immersion
H_e ... inlet section height
B_e ... inlet section width
H_z ... cylindrical body height
D_u ... dust discharge diameter
L_i ... lower bound for the i-th cyclone design parameter
U_i ... upper bound for the i-th cyclone design parameter
v_p ... gas flow rate
d_{50} ... cut-off diameter
ρ_{CE} ... correlation between CE and the overall constraint violation

ρ_{PD} ... correlation between PD and the overall constraint violation
n ... number of floors
m ... number of elevator cars
p ... probability of a newly arriving passenger
$s_i(t)$ ... state of the i-th node
$c_i(t)$ ... passenger state of the i-th node
$e_i(t)$ ... elevator state of the i-th node
π ... control policy
N_{n} ... number of nodes
N_{t} ... number of the performed state transition steps
M_{w} ... number of nodes with waiting passengers
M_{t} ... number of total elevator stops
N_{t} ... number of state transition steps
M_{s} ... maximum number of elevator skips
ρ_{f_1} ... correlation between f_1 and the overall constraint violation
ρ_{f_2} ... correlation between f_2 and the overall constraint violation

Chapter 1

Introduction

1.1 Background

Many real-world continuous optimization problems involve multiple objectives that need to be optimized simultaneously. If the objectives under consideration are in conflict, there exist multiple solutions to the given optimization problem representing various trade-offs between the objectives. Such problems are called *multiobjective optimization problems* (MOPs), and the set of all optimal solutions is known as the *Pareto-optimal set*. The image of the Pareto-optimal set in the objective space is called the *Pareto front*. For example, many engineering design problems are of this type. When designing a product, an industrial company often wants to achieve as good efficiency for the product as possible and at the same time manufacture the product using as few resources as possible. Besides multiple conflicting objectives, real-world problems often include various constraints that limit the search space (Ma & Wang, 2019). For example, in the engineering design problems, the products often have to respect geometric restrictions, a maximum allowed weight, etc. MOPs with constraints are called *constrained multiobjective optimization problems* (CMOPs).

Among the most widely used approaches for dealing with CMOPs are undoubtedly *multiobjective evolutionary algorithms* (MOEAs) equipped with *constraint handling techniques* (CHTs), since they are capable of finding a good set of *feasible solutions* in a single run. However, until recently the development of CHTs for multiobjective optimization problems has received little attention in the evolutionary computation community. According to (Mezura-Montes & Coello Coello, 2011), the main reason for this is that it was generally believed that CHTs designed for single-objective optimization can be easily adapted for multiobjective optimization. As the evolutionary computation community is now realizing that this is more difficult than expected, increasingly more research is focusing on constraint handling in multiobjective optimization. Indeed, many novel approaches for constraint handling and new benchmark suites of CMOPs have been proposed in the last years (Vodopija et al., 2019).

Despite the large number of recently published articles in the field of constrained multiobjective optimization, CMOPs for benchmarking MOEAs are still insufficiently understood and characterized (Picard & Schiffmann, 2021). According to (Bartz-Beielstein et al., 2020), there are two main options for characterizing or evaluating the quality of optimization problems, namely through the *feature space* and *performance space*.

The feature space can be seen as a space of problem characteristics, including basic characteristics such as problem dimensionality and type of objective and constraint functions, as well as more advanced problem characteristics derived using methods developed in the field of *exploratory landscape analysis* (ELA) (Mersmann et al., 2011). However,

there are only limited studies on ELA in the multiobjective context, and the situation is even worse in constrained multiobjective optimization. For example, recently proposed constraint handling techniques assume that CMOP violation landscapes consist of several "suboptimal" subregions—what we call *violation multimodality*—that severely aggravate algorithm performance (Zhu et al., 2020). While the multimodality for unconstrained multiobjective optimization has been extensively studied in (Grimme et al., 2021), the violation multimodality has not yet been studied from the ELA perspective (Filipič & Vodopija, 2021).

On the other hand, the performance space represents the problems based on the obtained algorithm performance (behavior) while solving these problems. Like in the feature space, basic statistics can be used here, such as mean or median algorithm performance, as well as more advanced methods, e.g., *data profiles* (Moré & Wild, 2009) or *empirical cumulative distribution functions* (ECDFs) (Hansen et al., 2021). In contrast to the aggregated values (means, medians, etc.), the latter two methods consider the progress of the whole algorithm run and, this way, provide more comprehensive information about the algorithm behavior. Unfortunately, the performance space perspective for characterizing CMOPs is even less explored than the feature space. In particular, there is no method to measure both the algorithm performance in approximating the Pareto front and constraint satisfaction simultaneously, which makes the experimental analysis incomplete.

In conclusion, due to the lack of appropriate techniques to characterize CMOPs, preparing a sound and well-designed experimental setup for constrained multiobjective optimization is a challenging task. A poorly designed benchmark might lead to inadequate conclusions about CMOP landscapes and MOEA performance (Vodopija, Tušar, et al., 2022).

1.2 Goals

The purpose of this dissertation is to address the above-stated shortcomings in the benchmarking of MOEAs. The goals of the dissertation are to:

- Generalize the concept of a problem landscape to constrained multiobjective optimization and extend the existing ELA techniques to CMOPs.
- Design and implement new ELA techniques for constrained multiobjective optimization, and assess their potential in characterizing CMOPs and predicting the MOEA performance.
- Design a novel algorithm performance assessment that will incorporate the constraint satisfaction into its framework.
- Assess the existing test suites of CMOPs based on the diversity and representativeness of various characteristics as seen from the feature and the performance space perspectives.
- Demonstrate the applicability of developed techniques in real-world applications.

1.3 Hypotheses

The dissertation verifies the following hypotheses:

1. Exploratory landscape analysis can be efficiently extended to constrained multiobjective optimization. The derived ELA features can provide meaningful information about the MOEA performance.

2. Monitoring the MOEA performance in approximating the Pareto front and constraint satisfaction simultaneously can improve the performance analysis and provide an important insight into the CMOPs and MOEAs.

3. The existing artificial test suites of CMOPs do not satisfactorily represent real-world problem characteristics (e.g., complex relationship between the objectives and constraints, unpredictable violation landscapes, etc.).

1.4 Original Contributions

The original contributions of the dissertation are:

- An exact mathematical formulation for the constrained multiobjective optimization problem landscape, including a formal definition for the violation multimodality. The rigorous mathematical formulation facilitates further consideration of the landscape analysis in constrained multiobjective optimization.

- The extension of ELA features to CMOPs, as well as the design of novel features specialized for CMOPs. Special attention is devoted to ELA features for describing violation multimodality as this has not been studied previously.

- The design of a novel method for algorithm performance assessment specialized for constrained multiobjective optimization. It simultaneously monitors both the performance in approximating the Pareto front and constraint satisfaction. The method improves the state of the art as it provides holistic information on MOEA performance.

- A critical investigation of the current test suites of CMOPs with respect to the feature and performance spaces. The obtained results can be used by benchmark designers to select or generate appropriate CMOP instances based on the characteristics they want to explore.

1.5 Organization of the Thesis

The thesis is further divided into two parts: Part I is dedicated to theoretical investigation and encompasses Chapters 2–5, while Part II presents two real-world case studies in Chapters 6 and 7.

Chapter 2 presents the background and the related work for the thesis. Here, we delve into the most critical concepts from multiobjective optimization, paying special attention to constraint handling. The theoretical advances are then introduced in Chapter 3, where we extend the concept of a fitness landscape to CMOPs. Additionally, we introduce two mathematical definitions to formally describe the violation multimodality in constrained multiobjective optimization.

Chapter 4 approaches the characterization of CMOPs from the feature space perspective. After introducing the ELA features designed for constrained multiobjective optimization, we contrast eight commonly used artificial test suites with a recently proposed suite of real-world problems.

Next, Chapter 5 presents the characterization of CMOPs from the performance space perspective. First, we introduce the algorithm performance assessment designed specifically for constrained multiobjective optimization. Then, we compare the eight aforementioned artificial test suites of CMOPs based on their ability to differentiate between MOEAs.

In Chapters 6 and 7, we demonstrate the applicability of our proposed methods for characterizing CMOPs by examining two real-world optimization problems: the design of cyclone dust separators and elevator group control. These optimization problems are analyzed both from the feature and the performance space perspectives.

Finally, Chapter 8 concludes the thesis by summarizing the work done, highlighting the essential findings, and outlining the key directions of future work in this domain.

Part I

Theory

Chapter 2

Theoretical Background and Related Work

This chapter provides the theoretical foundation for the thesis. We begin by introducing the definitions of a CMOP and constraint violation in Section 2.1. Following this, the concept of the fitness and violation landscape is presented in Section 2.2. We then discus the prevalent MOEAs and CHTs employed in constrained multiobjective optimization in Sections 2.3 and 2.4, respectively. Our attention shifts to the algorithm performance measurement in Sections 2.5 and 2.6, where we explore empirical runtime distributions and introduce specific quality indicators. Subsequently, we highlight the most frequently used test suites of CMOPs in Section 2.7. The chapter concludes with a brief review of the relevant literature on the characterization of CMOPs in Section 2.8.

2.1 Constrained Multiobjective Optimization Problems

A CMOP is, without loss of generality, formulated as:

$$\begin{aligned} \text{minimize} \quad & f_m(x), \quad m = 1, \ldots, M \\ \text{subject to} \quad & g_j(x) \leq 0, \quad j = 1, \ldots, J \\ & h_j(x) = 0, \quad j = J+1, \ldots, J+K \end{aligned} \tag{2.1}$$

where $x = (x_1, \ldots, x_D)$ is a *search vector*, $f_m : S \to \mathbb{R}$ are *objective functions*, $g_j : S \to \mathbb{R}$ *inequality constraint functions*, $h_j : S \to \mathbb{R}$ *equality constraint functions*, $S \subseteq \mathbb{R}^D$ is a *search space* of dimension D, and M, J and K are numbers of objectives, inequality and equality constraints, respectively.

In continuous optimization, the equality constraints are usually reformulated into inequality constraints as follows:

$$g_j(x) = |h_j(x)| - \eta \leq 0, \quad j = J+1, \ldots, J+K \tag{2.2}$$

where $\eta > 0$ is a user-defined tolerance value to relax equality constraints (in our work it is set to 10^{-4}). Using this reformulation, all constraints can be treated as inequality constraints. A solution x is said to be *feasible* if it satisfies all the constraints $g_j(x) \leq 0$ for $j = 1, \ldots, J+K$.

One of the most important concepts in constrained optimization is the notion of *constraint violation*. For a single constraint g_j it is defined as $v_j(x) = \max(0, g_j(x))$. For all constraints together it is combined as

$$v(x) = \sum_{j}^{J+K} v_j(x) \tag{2.3}$$

into the *overall constraint violation*. A solution x is feasible if and only if its overall constraint violation equals zero ($v(x) = 0$). Note that other definitions for overall constraint violation exist, and their use would impact the analysis performed in this thesis. However, the proposed definition for the overall constraint violation is by far the most commonly adopted in constrained optimization (Filipič & Vodopija, 2021), and as such, it represents the most appropriate choice.

A feasible solution $x \in S$ *dominates* another solution $y \in S$ if $f_m(x) \leq f_m(y)$ for all $1 \leq m \leq M$ and $f_m(x) < f_m(y)$ for at least one $1 \leq m \leq M$. Additionally, a solution $x^* \in S$ is *Pareto optimal* if there is no solution $x \in S$ such that it dominates x^*.

The set of all feasible solutions is called the *feasible region* and is denoted by $F = \{x \in S \mid v(x) = 0\}$. All nondominated feasible solutions represent a *Pareto-optimal set*, S_o. The image of the Pareto-optimal set in the objective space is the *Pareto front* and is denoted here by $P_\text{o} = \{f(x) \mid x \in S_\text{o}\}$.

The *ideal objective vector*, z^ide, is defined as the vector in the objective space that contains the optimal objective value for each objective separately and it is expressed as

$$z^\text{ide} = \left(\inf_{x \in F} f_1(x), \ldots, \inf_{x \in F} f_M(x) \right). \tag{2.4}$$

Additionally, the *nadir objective vector*, z^nad, consists in each objective of the worst value obtained by any Pareto-optimal solution. It can be expressed as

$$z^\text{nad} = \left(\sup_{x \in S_\text{o}} f_1(x), \ldots, \sup_{x \in S_\text{o}} f_M(x) \right). \tag{2.5}$$

An additional important concept is the region of interest in the objective space, Z, which represents the set of objective vectors bounded by the ideal and nadir objective vectors.

If good approximations for ideal and nadir objective vectors are known, the objective functions can be normalized to

$$\frac{f_m(x) - z_m^\text{ide}}{z_m^\text{nad} - z_m^\text{ide}}. \tag{2.6}$$

This way, the objective values are of approximately the same magnitude and the range of the objective values for Pareto-optimal solutions is $[0, 1]$. Note that after normalization z^ide consists of m zeros ($z^\text{ide} = (0, \ldots, 0)$), z^nad of m ones ($z^\text{nad} = (1, \ldots, 1)$), and the region of interest Z equals $[0, 1]^M$. In the rest of this thesis, we assume that all the objective functions are normalized.

2.2 Fitness and Violation Landscapes

The concept of a *fitness landscape* was formally defined in (Stadler, 2002) by identifying the following three key elements:

- $S \subseteq \mathbb{R}^D$ is the search space,
- $f : S \to \mathbb{R}$ is the objective function,
- $d : S \times S \to \mathbb{R}$ is the distance metric.

The distance metric d is used to quantify the similarity between solutions in the search space and to define *neighborhoods*. In continuous optimization, the Euclidean distance is often used. Close solutions are those whose Euclidean distance is within a predefined threshold, δ. Formally, the solution $y \in S$ is part of the neighborhood of the solution $x \in S$, denoted by $\mathcal{N}(x, \delta)$, if and only if $d(x, y) \leq \delta$.

In (Malan et al., 2015), an adaptation of the fitness landscape was proposed for *constrained single-objective optimization problems* (CSOPs) by replacing the objective function with the overall constraint violation function. The resulting idea is known as the *violation landscape*.

An additional very important notion for our work is *connectedness*. A set $X \in S$ of a space is said to be *disconnected* if it can be represented as the union of two or more disjoint non-empty open subsets from X. Otherwise, X is said to be *connected*. The maximal connected subsets (with respect to the inclusion order) of a non-empty space are called *connected components*. The components of any space X form a partition of X—they are disjoint, non-empty, and their union is the entire space.

2.3 Multiobjective Evolutionary Algorithms

To deal with CMOPs, population-based optimization methods such as MOEAs have been intensively studied over the last decades and are increasingly widely adopted in practice. They are particularly suitable for solving multiobjective optimization problems because they can find an approximation of the Pareto set in a single run. In the following, we introduce the five MOEAs used in this thesis.

2.3.1 Nondominated Sorting Genetic Algorithm II

Nondominated Sorting Genetic Algorithm II (NSGA-II) (Deb et al., 2001) is one of the most popular and widely used MOEAs in theory and practice. It uses nondominated sorting and crowding distance to select the best solutions for the new generation. Nondominated sorting divides the population into disjunctive subsets according to the dominance relation. These subsets are called nondomination levels. For example, all the nondominated solutions from the population correspond to the first nondomination level and represent a subset of the best solutions. Then, all the nondominated solutions after removing all the solutions in the first nondomination level from the population form the second nondomination level, and so on. Additionally, NSGA-II uses simulated binary crossover and polynomial mutation as variation operators. The main drawback of NSGA-II is that it is ineffective for manyobjective optimization problems ($M \geq 4$).

2.3.2 Differential Evolution for Multiobjective Optimization

Robič and Filipič (2005) proposed Differential Evolution for Multiobjective Optimization (DEMO). As its name indicates, this algorithm is an extension of the well-known single-objective optimizer differential evolution. The main idea of DEMO is to use differential evolution operators to generate new candidate solutions. There exist various variants of DEMO. However, we use the one that uses nondominated sorting in the survivor selection phase because it makes it possible to incorporate a CHT easily. Like NSGA-II, DEMO does not perform well for problems with many objectives.

2.3.3 Multiobjective Evolutionary Algorithm Based on Decomposition

Multiobjective Evolutionary Algorithm Based on Decomposition (MOEA/D) (Zhang & Li, 2007) was explicitly designed to deal with manyobjective optimization problems. The algorithm decomposes a MOP into several subproblems and optimizes them simultaneously. Each subproblem is defined by a decomposition function and a weight vector ν which determines the subproblem directions. One of the most frequently used decompositions is

the so-called Tchebycheff decomposition method defined as follows:

$$\min g^{\text{te}}(x \mid \nu) = \max_{1 \leq m \leq M} \{\nu_m \, |f_m(x) - z_m^*|\}. \tag{2.7}$$

Each subproblem defined with a decomposition method can be seen as an SOP. This is convenient for constraint handling as almost any single-objective CHT can be easily incorporated into the MOEA/D framework. As a result, MOEA/D is one of the most studied algorithms in constrained multiobjective optimization. The original version of MOEA/D uses simulated binary crossover and polynomial mutation as variation operators. An updated version, called MOEA/D-DE (H. Li & Zhang, 2009), employs differential evolution operators and, in general, produces better results for continuous MOPs.

2.3.4 Nondominated Sorting Genetic Algorithm III

Recognizing the limitations of NSGA-II in solving MOPs with many objectives, Deb and Jain (2014) proposed an updated version of the algorithm known as Nondominated Sorting Genetic Algorithm III (NSGA-III). This algorithm is similar to the original NSGA-II. However, instead of relying on crowding distance, it uses a decomposition-like approach to preserve the diversity in the population. The variation operators are still simulated binary crossover and polynomial mutation.

2.3.5 Constrained Two-Archive Evolutionary Algorithm

K. Li et al. (2019) introduced a two-archive evolutionary algorithm, denoted as C-TAEA. This algorithm maintains two collaborative and complementary archives: convergence-oriented archive (CA) and diversity-oriented archive (DA). The CA supports the convergence and feasibility of the evolutionary process. It provides the selection pressure toward the Pareto front. In contrast, the DA tends to maintain the diversity of the evolutionary process by exploring the areas that the CA has not exploited. Finally, a restricted mating selection mechanism was proposed for selecting from the CA and DA to enhance the complementary effect of the two archives.

2.4 Constraint Handling in Multiobjective Optimization

Many novel approaches to constraint handling and their application have been proposed in the last years, e.g., (Liu & Wang, 2019; Ma & Wang, 2021; Vodopija & Filipič, 2019; Vodopija et al., 2020; Zhu et al., 2020). This section presents some of the most frequently used CHTs that are also used throughout this thesis.

2.4.1 Constrained Domination Principle

The constrained domination principle (CDP) (Deb et al., 2002) is an extension of dominance relation and is the most widely-used technique to solve CMOPs. It strictly favors feasible solutions over infeasible ones. While feasible solutions are compared based on Pareto dominance, infeasible solutions are compared according to the overall constraint violation. The formal definition of CDP, as presented in (Zhu et al., 2020), is the following:

$$x \preceq_{\text{CDP}} y \Leftrightarrow \begin{cases} x \preceq y & \text{if } v(x) = v(y) = 0 \\ v(x) < v(y) & \text{otherwise} \end{cases}. \tag{2.8}$$

The CDP technique can be incorporated in NSGA-II, NSGA-III, and DEMO in the replacement phase, i.e., survivor selection, where dominance relation is replaced by $\preceq_{\text{CDP}}$. The other components in each of the algorithms remain unchanged.

2.4.2 Penalty Function Approach

Due to its simplicity, the penalty function approach is widely used, and it is especially suited to problems with disjoint feasible regions (Eiben & Smith, 2003). The idea is to enhance the objective function with a distance metric of the infeasible solution to the feasible region. In this thesis, the distance metric to the feasible region is the overall constraint violation of the infeasible solution.

The penalty function was considered in combination with MOEA/D (Campelo et al., 2020). In MOEA/D, the decomposition function (Tchebycheff function in this thesis) is enhanced with a penalty term as follows:

$$f^{\mathrm{p}}(x \mid \gamma, \nu, z^{\star}) = g^{\mathrm{te}}(x \mid \nu, z^{\star}) + \gamma v(x). \tag{2.9}$$

Here, g^{te} is the Tchebycheff decomposition function defined in (2.7), and $\gamma > 0$ the penalty weight. The enhanced decomposition function f^{p} is used to compare solutions in the update phase of MOEA/D.

2.4.3 Epsilon-Constraint Method

In addition to the penalty function approach, we study a CHT for MOEA/D based on the ε-constraint relation defined as:

$$x \preceq_{\varepsilon} y \Leftrightarrow \begin{cases} g^{\mathrm{te}}(x) < g^{\mathrm{te}}(y) & \text{if } (v(x) \leq \varepsilon \text{ and } v(y) \leq \varepsilon) \\ & \text{or } v(x) = v(y) \\ v(x) < v(y) & \text{otherwise} \end{cases}. \tag{2.10}$$

The MOEA/D algorithm using this approach is called MOEA/D-IEpsilon (Fan, Li, Cai, Huang, et al., 2019). In particular, the comparison level ε_t is updated in each generation following the expression:

$$\varepsilon_t = \begin{cases} v(x^{\theta}) & \text{if } t = 0 \\ (1-\tau)\varepsilon_{t-1} & \text{if } \rho_{\mathrm{F}}(P_t) < \alpha \text{ and } t < T_{\mathrm{c}} \\ (1+\tau)v_{\max} & \text{if } \rho_{\mathrm{F}}(P_t) \geq \alpha \text{ and } t < T_{\mathrm{c}} \\ 0 & \text{if } t \geq T_{\mathrm{c}} \end{cases} \tag{2.11}$$

where t is the generational counter, τ, α, T_{c} are user-defined parameters, $v(x^{\theta})$ is the overall constraint violation of the top θ-th individual (according to overall constraint violation value) in the initial population, and $\rho_{\mathrm{F}}(P_t)$ is the proportion of feasible solutions in the current population P_t.

2.4.4 Bi-objective Approach

This CHT is used by C-TAEA, and it is incorporated within the update of the convergence archive. Similarly to CDP, C-TAEA strictly favors feasible solutions which are compared based on Pareto dominance. On the other hand, the infeasible solutions are ranked using nondominated sorting for a custom bi-objective problem expressed as

$$\text{minimize} \quad (v(x), g^{\mathrm{te}}(x \mid \nu)). \tag{2.12}$$

The CA is updated with all feasible solutions and the best infeasible solutions according to the Pareto ranking applied in (2.12). In addition, the DA does not consider feasibility at all allowing infeasible solutions to persist in the population. More information on this method is available in (K. Li et al., 2019).

2.5 Empirical Runtime Distributions

The performance measurement approach used in the COCO framework (Hansen et al., 2022; Hansen et al., 2021), which is a widely used platform for comparing continuous optimizers, relies on the number of function evaluations[1]—called *runtime*—needed for an algorithm, a, to reach predefined quality indicator targets. More precisely, an algorithm run can be, after performing T function evaluations, presented as a sequence of candidate solutions, $A^T(a) = \{x^{(1)}, \ldots, x^{(T)}\}$. Within this framework, a quality indicator, I, is defined as a function mapping $A^T(a)$ to a real value. Here, an assumption is that low quality indicator values indicate better sequences of candidate solutions, and vice versa. Additionally, the runtime for a given quality indicator target equals to the lowest T for which $I(A^T(a))$ reaches the given target precision value, τ. Note that in the following, if there is no ambiguity, we remove the algorithm notation a from $A^T(a)$.

In practice, we define several target precision values to understand the algorithm behavior through the entire run. Runtimes can be formally defined as random variables and displayed as an empirical cumulative distribution function—called empirical runtime distribution (ERD) in the COCO framework. ERDs are used to display the proportion of target values reached within a specified budget and can be easily aggregated over multiple restarts, runs or even multiple problems. For more details on ERDs, see (Hansen et al., 2022; Hansen et al., 2021). The runtime data set for an algorithm a and all targets τ-s is denoted as $\{T_a(\tau)\}_\tau$. Finally, the runtimes in COCO are usually studied in a logarithmic scale and this perspective is used throughout this thesis as well.

2.6 Quality Indicators

Based on the nature of the optimization problem, there are various quality indicators used in the literature. Those relevant for this thesis and used in COCO are as follows.

2.6.1 Single-Objective Optimization

In this case, the quality indicator is the best so far observed objective function value:

$$I^{\mathrm{SOP}}(A^T) = \min_{x \in A^T} f(x). \tag{2.13}$$

2.6.2 Constrained Single-Objective Optimization

The quality indicator for unconstrained problems (2.13) is extended by the addition of the overall constraint violation as follows:

$$I^{\mathrm{CSOP}}(A^T) = \min_{x \in A^T} f(x) + v(x). \tag{2.14}$$

2.6.3 Multiobjective Optimization

The quality indicator for MOPs consists of two parts. When no solution from the sequence A^T dominates the nadir point (reference point), the distance to the region of interest Z is used to measure the quality of the solutions. In contrast, when at least one of the solutions

[1]When we refer to a function evaluation, we actually mean the evaluation of all the objective and constraint functions. For example, for a bi-objective problem with three constraints we need to perform five evaluations, however, we count this as a single function evaluation.

dominates the nadir point, the hypervolume indicator is used instead. This quality indicator can be mathematically expressed as:

$$I^{\mathrm{MOP}}(A^T) = \begin{cases} -I^{\mathrm{HV}}(A^T) & \text{if } A^T \preceq \{z^{\mathrm{nad}}\} \\ d(A^T, Z) & \text{otherwise.} \end{cases} \tag{2.15}$$

Here,

$$I^{\mathrm{HV}}(A^T) = V\left(\bigcup_{x \in \mathcal{D}(A^T)} [f_1(x), 1] \times \cdots \times [f_M(x), 1]\right) \tag{2.16}$$

represents the hypervolume of the archive A^T, $\mathcal{D}(A^T)$ is the set of all solutions from A^T dominating the reference point which equals $(1, \ldots, 1)$, and

$$d(A^T, Z) = \inf_{(x,z) \in A^T \times Z} \|f(x) - z\| \tag{2.17}$$

is the smallest Euclidean distance between the archive and the region of interest Z. Additional information on this quality indicator can be found in (Hansen et al., 2022).

2.6.4 Target Precision Values for MOPs

For each problem a set of quality indicator target values is chosen, which is used to measure algorithm runtimes and in turn to calculate ERDs. The target values are computed in the form of $\tau(\varepsilon) = \tau^{\mathrm{ref}} + \varepsilon$, where τ^{ref} is a reference I^{MOP} value. It is either based on the hypervolume of the true Pareto front or an estimation for it. In COCO, 51 positive target precision values $\varepsilon \in \{10^{-5}, 10^{-4.9}, \ldots, 10^{-0.1}, 10^{0}\}$ are chosen[2]. Note that it is not uncommon that the quality indicator value of the algorithm never reaches some of these target values, which leads to missing runtime measurements.

2.7 Test Suites of CMOPs

The most notable artificial test suites of CMOPs used to asses the performance of constrained multiobjective optimization algorithms are CTP (Deb et al., 2001), CF (Zhang et al., 2008), C-DTLZ (Jain & Deb, 2014), NCTP (J. P. Li et al., 2016), DC-DTLZ (K. Li et al., 2019), LIR-CMOP (Fan, Li, Cai, Huang, et al., 2019), DAS-CMOP (Fan, Li, Cai, Li, et al., 2019), MW (Ma & Wang, 2019), and RCM (Kumar et al., 2021). The basic characteristics of the test suites are summarized in Table 2.1.

2.7.1 CTP

The first suite of CMOPs with constraints that have adjustable difficulties was proposed in (Deb et al., 2001). The authors proposed eight test problems called CTPs (Constrained Test Problems). Most of them have disconnected or discrete Pareto fronts. The main shortcoming of this suite is that all the problems have large feasible regions, and thus the feasible solutions are easily discovered.

[2]In COCO, negative values are also introduced in case the algorithms find better Pareto front approximations than available ones.

Table 2.1: Characteristics of test suites: number of problems, dimension of the search space D, number of objectives M, and number of constraints J (all constraints are inequalities). The characteristics of the selected RCM problems (see Table 2.2) are shown in parentheses.

Test suite	#problems	D	M	J
CTP	8	*	2	2, 3
CF	10	*	2, 3	1, 2
C-DTLZ	6	*	*	1, *
NCTP	18	*	2	1, 2
DC-DTLZ	6	*	*	1, *
DAS-CMOP	9	*	2, 3	7, 11
LIR-CMOP	14	*	2, 3	2, 3
MW	14	*	2, *	1–4
RCM	50 (11)	2–34 (2–5)	2, 5	1–29 (1–8)

*Scalable parameter.

2.7.2 CF

In (Zhang et al., 2008), ten test problems were proposed and named CFs (Constrained Functions). The problems have disconnected Pareto fronts and contain complicated variable linkages. Consequently, the optimization algorithms often struggle with convergence. Like the CTP problems, CFs also have large feasible regions. Nevertheless, CFs are still one of the most frequently used test suites in the literature.

2.7.3 C-DTLZ

Jain and Deb (2014) designed six CMOPs built upon the well-known DTLZ suite by Deb, Thiele, Laumanns, and Zitzler (Deb et al., 2002) used in unconstrained multiobjective optimization. All the problems have a scalable number of objectives, while two problems are additionally scalable in the number of constraints. These problems are called C-DTLZ (Constrained DTLZ) and are often used for benchmarking MOEAs. It is worth noting that some problems have large feasible regions and are thus easily solved.

2.7.4 NCTP

Inspired by CTPs, 18 new test problems were introduced in (J. P. Li et al., 2016). They are known as NCTPs (New CTPs). Compared with the original CTPs, these problems include an extra constraint to reduce the size of feasible regions.

2.7.5 DC-DTLZ

Like in C-DTLZ problems, K. Li et al. (2019) proposed six CMOPs built upon the DTLZ suite. The problems are called DC-DTLZs and are scalable in the number of objectives. In addition, two problems are scalable in the number of constraints. In contrast to the C-DTLZ problems, DC-DTLZs have smaller feasible regions and more complex landscapes.

2.7.6 DAS-CMOP

In 2019, Fan, Li, Cai, Li, et al. (2019) introduced a DAS-CMOP suite (Difficulty-Adjustable and Scalable CMOPs). There are nine problems of increasing hardness which are scalable in the number of objectives. However, in the literature, we see DAS problems with two or

Table 2.2: Characteristics of the selected RCM problems: dimension of the search space D, number of objectives M, and number of constraints J (all constraints are inequalities).

Test problem	Name	D	M	J
RCM2	Vibrating platform design	5	2	5
RCM3	Two bar truss design	3	2	3
RCM4	Welded beam design	4	2	4
RCM5	Disc brake design	4	2	4
RCM10	Two bar plane truss design	2	2	2
RCM11	Water resources management	3	5	7
RCM12	Simply supported beam design	4	2	1
RCM14	Multiple disk clutch brake design	5	2	8
RCM16	Cantilever beam design	2	2	2
RCM18	Front rail design	3	2	3
RCM20	Hydro-static thrust bearing design	4	2	7

three objectives only. The problems are known for introducing a mechanism to adjust the difficulty of the CMOPs: difficulty in obtaining convergence and feasibility, and maintaining diversity.

2.7.7 LIR-CMOP

The authors of DAS-CMOPs proposed a second suite named LIR-CMOP (Large Infeasible Region CMOPs) (Fan, Li, Cai, Huang, et al., 2019). The characteristics of these problems are similar to those of DAS-CMOPs. The suite includes 14 constrained multiobjective optimization problems with two or three objectives and constraints.

2.7.8 MW

The MW suite was proposed by Ma and Wang (2019). It is one of the most comprehensive suites as the proposed CMOPs reflect various problem characteristics. In particular, there are four types of CMOPs based on how the presence of constraints affects the objective space. This suite consists of eleven bi-objective problems and three problems that are scalable in the number of objectives. The problems include one to four constraints.

2.7.9 RCM

All the test suites described in the previous subsections consist of artificially designed CMOPs and are not derived from real-world applications. To overcome this weakness, a novel suite named RCM (Real-world Constrained Multiobjective) optimization problems was proposed in (Kumar et al., 2021). The RCM suite collects 50 real-world optimization problems based on physical models, including problems from mechanical design, chemical engineering, power electronics, etc. The problem instances come in various dimensions and numbers of objectives. After excluding all the problems containing discrete variables and those with more than five variables, eleven RCM problems remained for our study, and their characteristics are summarized in Table 2.2. As we can see, ten of the selected RCM problems have two objectives, while the remaining one has five objectives. The number of constraints varies from one to eight, and all the constraints are inequalities.

2.8 Characterization of CMOPs

There are two main options for characterizing and evaluating the quality of optimization problems: through the feature space and performance space (Bartz-Beielstein et al., 2020).

2.8.1 Feature Space

There are only limited studies on ELA in the multiobjective combinatorial context (Daolio et al., 2017; Liefooghe et al., 2020; Verel et al., 2013), an initial study in continuous multiobjective optimization (Liefooghe et al., 2021), and some first attempts to visualize bi-objective continuous problem landscapes with up to three variables (Fonseca, 1995; Kerschke & Grimme, 2017; Kerschke et al., 2016; Schäpermeier et al., 2021). As for constrained optimization, two works characterize linear and quadratic constraints (Poursoltan & Neumann, 2014, 2015). In addition to features for determining the relationship between the constraints, the total number of constraints and the feasibility ratio in the vicinity of the optimal solution were considered to measure the difficulty of CSOPs. Malan et al. (2015) introduced five features to characterize CSOPs with arbitrary constraints. They aimed at describing the feasibility ratio, disjointedness of feasible regions, and relationships between the objectives and constraints. This work was later extended to multiobjective optimization (Picard & Schiffmann, 2021).

Although Picard and Schiffmann (2021) considered ELA techniques for multiple objectives and constraints simultaneously, many other crucial aspects still need to be investigated. For example, recently proposed constraint handling techniques assume that CMOP violation landscapes consist of several "suboptimal" subregions—what we will call *violation multimodality*—or have a complex local structure that both severely aggravate algorithm performance (Zhu et al., 2020). To the best of our knowledge, none of these aspects have been studied from the ELA perspective yet (Filipič & Vodopija, 2021).

2.8.2 Performance Space

In the literature, the performance indicators used in constrained multiobjective optimization are the same as those used in unconstrained multiobjective optimization, they are simply applied only to feasible solutions. The most frequently employed indicators are the *hypervolume indicator* (Zitzler & Thiele, 1999) and *inverted generational distance* (Bosman & Thierens, 2003) since they can provide information about the convergence and the diversity of the obtained Pareto front approximation. For monitoring the performance during the run, one can use convergence graphs, data profiles or ECDFs. However, none of these techniques provide relevant information until feasible solutions are discovered. As a result, essential insights about the algorithm behavior and CMOP characteristics are missed. To overcome this situation, some papers report also the constraint satisfaction progress (Fan, Li, Cai, Huang, et al., 2019). Nevertheless, to the best of our knowledge, no method from the literature simultaneously measures both the convergence towards the Pareto front and constraint satisfaction, making the experimental analysis incomplete.

Moreover, we are aware of only a single work analyzing CMOPs from the performance space perspective which was conducted in 2017 (Tanabe & Oyama, 2017). The authors used five CHTs to characterize five artificial and seven real-world test problems. The results revealed that only a single artificial test problem was suitable for benchmarking algorithms since the other four problems could be solved even without employing a CHT. Additionally, the studied real-world problems were inadequate since they could not differentiate among MOEAs—a desired property of a test problem as it provides relevant information for algorithm designers (Bartz-Beielstein et al., 2020). Since 2017, several novel test suites of

CMOPs have been proposed, and their ability to differentiate among MOEAs has not been investigated yet (Filipič & Vodopija, 2021).

Chapter 3

Theoretical Advances

In this chapter, we provide an overview of the theoretical advances in landscape analysis for CMOPs (Vodopija, Tušar, et al., 2022). After defining constrained multiobjective problem landscapes in Section 3.1, we introduce feasible components in Section 3.2 and basins of attraction in the violation landscape in Section 3.3. Finally, Section 3.4 concludes with a visualization of three CMOP landscapes depicting these novel concepts.

3.1 Constrained Multiobjective Problem Landscapes

Following the definitions of fitness and violation landscapes from Section 2.2, we generalize these concepts to CMOPs by including multiple objective functions and the overall constraint violation function simultaneously. A *constrained multiobjective problem landscape* is defined with the following four elements:

- $S \subseteq \mathbb{R}^D$ is the search space,
- $f : S \to \mathbb{R}^M$ is the objective vector function,
- $v : S \to \mathbb{R}$ is the overall constraint violation function,
- $d : S \times S \to \mathbb{R}$ is the distance metric.

The notion of a neighborhood can be applied to CMOP landscapes without any modification. For example, a solution $x^* \in S$ is a *local Pareto-optimal solution* if it is feasible and there exists a $\delta > 0$ such that no feasible solution from the neighborhood $\mathcal{N}(x^*, \delta)$ dominates x^*. The set of all local Pareto-optimal solutions will be denoted by S_{l}. It is obvious that $S_{\text{o}} \subseteq S_{\text{l}}$.

In (Zhu et al., 2020), two topological structures concerning violation multimodality were identified that might cause premature algorithm stagnation: feasible subregions and areas that locally violate the constraints the least. The feasible region can consist of multiple disconnected subregions. The more such subregions there are, the harder it is for the algorithm to find a good approximation for the whole Pareto-optimal set. Moreover, the search space of a CMOP can have many areas that locally violate the constraints the least and are infeasible. Again, many such areas might aggravate the performance of an algorithm since it can get stuck in the infeasible region. Although these concepts are essential, they are not formally defined, which can lead to ambiguous interpretations. To overcome this situation, we provide a rigorous mathematical formulation and, in this way, facilitate their further consideration.

3.2 Feasible Components

A connected component $\mathcal{F} \subseteq F$ of the feasible region is called a *feasible component*. Note that all feasible components form a partition of the feasible region. With feasible components, we provide a formal definition of feasible subregions. A feasible component can be treated as "suboptimal" when it contains no Pareto-optimal solutions. Obviously, if a problem landscape contains several "suboptimal" feasible components, it might be harder for an optimizer to find good approximations for all Pareto-optimal solutions.

3.3 Basins of Attraction in the Violation Landscape

Basins of attraction in the fitness landscape are well-known objects widely used to characterize unconstrained single-objective optimization problems. In this thesis, we reuse this concept in the violation landscape. However, we need to introduce some crucial notions first.

Similarly to the local Pareto-optimal solution, we can define a local "optimal" solution in the violation landscape—a solution that locally violates the constraints the least. Formally, x^* is a *local minimum-violation solution* if there exist a $\delta > 0$ such that $v(x^*) \leq v(x)$ for all $x \in \mathcal{N}(x^*, \delta)$. Additionally, if there exists no solution $x \in S$ such that $v(x^*) > v(x)$, then x^* is a *(global) minimum-violation solution*. The set of all local minimum-violation solutions will be denoted by F_{l} and of all minimum-violation solutions by F_{m}. It is obvious that $F_{\mathrm{m}} \subseteq F_{\mathrm{l}}$ and if there exist feasible solutions in S, then $F_{\mathrm{m}} = F$.

A connected component $\mathcal{M} \subseteq F_{\mathrm{l}}$ is called a *local minimum-violation component* and it provides a formal definition for areas that locally violate the constraints the least. All the local minimum-violation components form a partition of F_{l}.

Now, we can finally define a basin of attraction. Let us consider an abstract local search procedure as a mapping from the search space to the set of local minimum-violation solutions, $\mu : S \to F_{\mathrm{l}}$, such that $\mu(x) = x$ for all $x \in F_{\mathrm{l}}$. Then, a *basin of attraction* of a local minimum-violation component $\mathcal{M}$ and local search μ is a subset of S in which μ converges towards a solution from $\mathcal{M}$, i.e., $\mathcal{B}(\mathcal{M}) = \{x \in S \mid \mu(x) \in \mathcal{M}\}$. In this case, the local minimum-violation component $\mathcal{M}$ is said to be an *attractor* of the basin $\mathcal{B}(\mathcal{M})$. If there is only one basin in S, then the corresponding violation landscape is said to be *unimodal*. Otherwise, it is *multimodal*.

Each feasible component is a subset of one of the local minimum-violation components. The reverse is, in general, not true. However, if a local minimum-violation component contains feasible solutions, then it contains at least one feasible component as well. Consequently, one can discover feasible components by first identifying local minimum-violation components with feasible solutions.

3.4 Visualization of CMOP Landscapes

Figure 3.1 illustrates problem landscapes and violation landscapes for problems C2-DTLZ2 (Jain & Deb, 2014), MW6 (Ma & Wang, 2019), DAS-CMOP1 (Fan, Li, Cai, Li, et al., 2019) and MW7 (Ma & Wang, 2019) with two variables and two objectives. All figures were obtained by evaluating a grid of 501×501 solutions from the search space. The first column shows the dominance ratio for each grid solution expressed as a proportion of grid solutions that dominate it (Fonseca, 1995). The solutions that are not dominated by any grid solution represent approximations of the Pareto-optimal solutions and are shown in black in the plots. The second column depicts the violation landscape in terms of constraint violation values with feasible components presented in white. The third column shows the

problem landscape by visualizing the dominance ratio of the feasible regions in blue hues and the infeasible regions in pink. Again, black denotes approximations of Pareto-optimal solutions.

The Pareto-optimal set of C2-DTLZ2 without considering constraints is the horizontal line $L : x_2 = 0.5$ (Figure 3.1a). Similarly holds for MW6 (Figure 3.1d), while the Pareto-optimal sets of DAS-CMOP1 and MW7 without constraints are curves in the search space (Figures 3.1g and 3.1j).

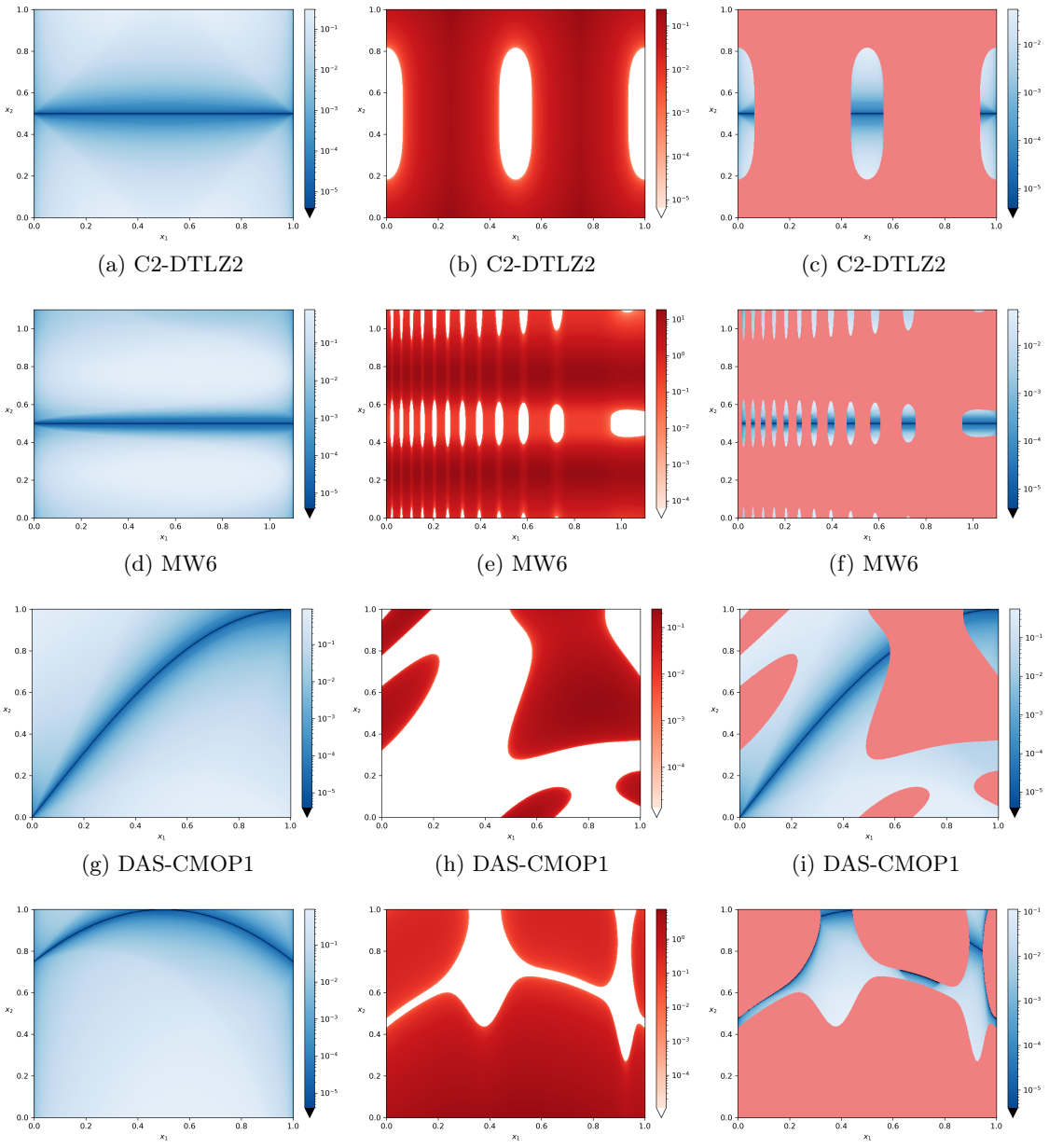
As we can see from Figure 3.1b, C2-DTLZ2 has three feasible components that are also the only three local minimum-violation components. The corresponding basins are separated with two vertical lines $L_1 : x_1 = 0.25$ and $L_2 : x_1 = 0.75$. All the feasible components and basins intersect with the Pareto-optimal set (Figure 3.1c). In other words, each basin attracts the search towards nondominated feasible solutions, making this problem easy for constraint handling.

On the other hand, MW6 consists of 35 feasible components and 72 basins (Figure 3.1e). Between two adjacent horizontal components, there is an additional basin that contains no feasible solutions. It is also interesting that not all the feasible components contain Pareto-optimal solutions (Figure 3.1f). This makes the MW6 problem harder to handle, especially for separation-based constraint handling techniques that separately handle feasible and infeasible solutions and strictly favor feasible solutions over the infeasible ones. The Pareto-optimal set is a subset of the Pareto-optimal set corresponding to MW6 without constraints.

The DAS-CMOP1 problem has three feasible components: a small one in the upper left corner, a medium-sized one in the upper right corner, and a large one in the middle. In contrast to C2-DTLZ2 and MW6, the feasible components have irregular shapes. The feasible components are also the only local minimum-violation components. The Pareto-optimal set is a subset of the Pareto-optimal set corresponding to DAS-CMOP1 without considering constraints.

In contrast, the Pareto-optimal set of the MW7 problem consists of two disconnected parts of the Pareto-optimal set belonging to the MW7 problem without constraints and parts of the boundary between the feasible and infeasible regions (Figure 3.1l). This problem has one feasible component and also only one basin of attraction (Figure 3.1k).

To identify and count the basins of attraction, we use the truncated Newton method (Nash, 1984) as a local search procedure μ. This method approximates the Newton's direction by approximately solving the Newton equation using an iterative technique. In this thesis, the conjugate gradient technique is used as the iterative solver. Note that any other deterministic local search could be used instead and possibly result in a different number of basins. Nevertheless, the truncated Newton method proved to be efficient and reliable for identifying basins of attraction.



(a) C2-DTLZ2 (b) C2-DTLZ2 (c) C2-DTLZ2

(d) MW6 (e) MW6 (f) MW6

(g) DAS-CMOP1 (h) DAS-CMOP1 (i) DAS-CMOP1

(j) MW7
Dominance ratio

(k) MW7
Violation landscape

(l) MW7
Problem landscape

Figure 3.1: Plots of the dominance ratio, the violation landscape, and the problem landscape of three benchmark problems. The approximation of the Pareto-optimal set is depicted in black.

Chapter 4

Characterization of CMOPs: Feature Space Perspective

In this chapter, we first introduce the ELA features designed for CMOPs (Vodopija, Tušar, et al., 2022) in Section 4.1. Using these features, we then analyze the CMOPs from the feature space perspective in Section 4.2. Specifically, we provide an analysis of the commonly used test suites by comparing their feature values.

4.1 ELA Features

Most of the novel constraint handling techniques proposed in multiobjective optimization are designed to tackle severe violation multimodality, complex local structure of the violation landscapes (disjoint feasible components and rugged constraint functions), and small feasibility ratios (Filipič & Vodopija, 2021; Ma et al., 2021; Picard & Schiffmann, 2021; Zhu et al., 2020). This preposition was the primary motivation for selecting the ELA approaches used in this thesis. They are all well-known and frequently used methods to derive various features for characterizing problem landscapes. In particular, we include four ELA approaches that we find reliable for deriving landscape features: space-filling design, information content, random walks, and adaptive walks. These are used to describe the multimodality and local structure of the violation landscapes and to provide global information, such as the feasibility ratio.

4.1.1 Space-Filling Design Features

The space-filling design approach is used to derive various features for characterizing feasible components, e.g., the number of components, their sizes and optimality. This information can be in turn used to quantify the violation multimodality. Additionally, features expressing the feasibility ratio, and correlations between the objectives and constraints are also derived using this approach. It comprises the following three steps:

1. Generate an initial sample $X_{\mathrm{S}} \subseteq S$ of solutions following a selected space-filling design and derive feasible solutions.
2. Cluster feasible solutions to obtain approximations for feasible components.
3. Find nondominated solutions among the obtained feasible solutions.

The first feature of this group is the number of feasible components, $N_{\mathcal{F}}$, and it is estimated as the total number of the obtained clusters. Next, the proportion of solutions

Table 4.1: The proposed ELA features to characterize CMOPs categorized into four groups: space-filling design, information content, random walk, and adaptive walk. "Our" indicates that the corresponding feature was proposed in (Vodopija, Tušar, et al., 2022).

Space-filling design features		
$N_{\mathcal{F}}$	Number of feasible components	Our
$\mathcal{F}_{\min}$	Smallest feasible component	Our
$\mathcal{F}_{\text{med}}$	Median feasible component	Our
$\mathcal{F}_{\max}$	Largest feasible component	Our
$\mathcal{O}(\mathcal{F}_{\max})$	Proportion of Pareto-optimal solutions in $\mathcal{F}_{\max}$	Our
$\mathcal{F}_{\text{opt}}$	Size of the "optimal" feasible component	Our
ρ_{F}	Feasibility ratio	(Picard & Schiffmann, 2021)
$\rho_{\min}$	Minimum correlation	(Malan et al., 2015)*
$\rho_{\max}$	Maximum correlation	(Malan et al., 2015)*
$\rho_{\partial S_o}$	Proportion of boundary Pareto-optimal solutions	Our
Information content features		
$H_{\max}$	Maximum information content	(Muñoz, Kirley, et al., 2015)*
ε_s	Settling sensitivity	(Muñoz, Kirley, et al., 2015)*
M_0	Initial partial information	(Muñoz, Kirley, et al., 2015)*
Random walk features		
$(\rho_{\partial F})_{\min}$	Minimum ratio of feasible boundary crossings	(Picard & Schiffmann, 2021)
$(\rho_{\partial F})_{\text{med}}$	Median ratio of feasible boundary crossings	(Picard & Schiffmann, 2021)
$(\rho_{\partial F})_{\max}$	Maximum ratio of feasible boundary crossings	(Picard & Schiffmann, 2021)
Adaptive walk features		
$N_{\mathcal{B}}$	Number of basins	(Kerschke & Trautmann, 2019)*
$\mathcal{B}_{\min}$	Smallest basin	Our
$\mathcal{B}_{\text{med}}$	Median basin	Our
$\mathcal{B}_{\max}$	Largest basin	Our
$(\mathcal{B}_{\text{F}})_{\min}$	Smallest feasible basin	Our
$(\mathcal{B}_{\text{F}})_{\text{med}}$	Median feasible basin	Our
$(\mathcal{B}_{\text{F}})_{\max}$	Largest feasible basin	Our
$\cup\mathcal{B}_{\text{F}}$	Proportion of feasible basins	Our
$v(\mathcal{B})_{\text{med}}$	Median constraint violation over all basins	Our
$v(\mathcal{B})_{\max}$	Maximum constraint violation of all basins	Our
$v(\mathcal{B}_{\max})$	Constraint violation of $\mathcal{B}_{\max}$	Our
$\mathcal{O}(\mathcal{B}_{\max})$	Proportion of Pareto-optimal solutions in $\mathcal{B}_{\max}$	Our
$\mathcal{B}_{\text{opt}}$	Size of the "optimal" basin	Our

*For the first time applied on CMOPs in (Vodopija, Tušar, et al., 2022).

from X_S that form a particular cluster is used to measure the size of the corresponding feasible component. Three features were derived with this respect: the smallest feasible component, $\mathcal{F}_{\min}$, the median feasible component, $\mathcal{F}_{\text{med}}$, and the largest feasible component, $\mathcal{F}_{\max}$.

The following two features considered within this group express the "optimality" of feasible components. The first feature reflects the "optimality" of the largest feasible component, $\mathcal{O}(\mathcal{F}_{\max})$, and it is expressed as the proportion of nondominated feasible solutions in the largest cluster. The second feature is the size of the "optimal" feasible component, $\mathcal{F}_{\text{opt}}$. It is approximated as the proportion of all solutions from the initial sample that form a cluster with the maximum number of nondominated feasible solutions.

The proportion of feasible solutions in the initial sample is used to estimate the feasibility ratio, ρ_F, of the corresponding CMOP.

In addition, two features express the correlations between the objectives and constraints. They are the minimum and maximum correlations among objectives and the overall constraint violation and are denoted by $\rho_{\min}$ and $\rho_{\max}$, respectively.

Finally, the proportion of Pareto-optimal solutions on the feasible region boundary is approximated as the proportion of nondominated solutions on the boundary of one of the obtained clusters and is denoted by $\rho_{\partial S_o}$.

4.1.2 Information Content Features

The information content approach was originally used to analyze a sequence of fitness values obtained by sorting a sample of solutions and characterizes the local structure of a fitness landscape. We extend this approach to the violation landscape. Instead of analyzing fitness sequences, we analyze sequences of overall constraint violation values. This way we can derive information about the smoothness, ruggedness and neutrality of the violation landscape.

The approach follows the description provided in (Muñoz, Kirley, et al., 2015) and it is here presented only briefly. The interested reader can find a detailed explanation in the original paper. First, a sample of solutions $X_I = \{x^{(1)}, \ldots, x^{(n)}\}$ is generated following a space-filling design. The generated solutions from X_I are sorted in a sequence as follows: The first solution is selected at random, while each subsequent solution is selected as the nearest solution to the current one. Then, a new sequence, $\phi(\lambda) = \{\phi^{(1)}(\lambda), \ldots, \phi^{(n-1)}(\lambda)\}$, is generated following the rule:

$$\phi^{(i)}(\lambda) = \begin{cases} \searrow & \text{if } \frac{\Delta v^{(i)}}{\|\Delta x^{(i)}\|} < -\lambda \\ \rightarrow & \text{if } \frac{|\Delta v^{(i)}|}{\|\Delta x^{(i)}\|} \leq \lambda \\ \nearrow & \text{if } \frac{\Delta v^{(i)}}{\|\Delta x^{(i)}\|} > \lambda \end{cases} \tag{4.1}$$

where $\lambda > 0$, $\Delta x^{(i)} = x^{(i+1)} - x^{(i)}$, $\Delta v^{(i)} = v^{(i+1)} - v^{(i)}$ and $v^{(i)} = v(x^{(i)})$. In more detail, the symbol $\searrow$ indicates a decrease of the overall constraint violation, $\rightarrow$ a neutral area within the threshold λ, and $\nearrow$ an increase of the overall constraint violation. In particular, two consecutive symbols compose a block, ab, where $a, b \in \{\searrow, \rightarrow, \nearrow\}$, which represents an object in the violation landscape, i.e., a slope, peak, etc. For example, $\nearrow\nearrow$ indicates a constant increase of the overall constraint violation, while $\searrow\nearrow$ indicates a local minimum. In particular, two identical consecutive symbols represent a smooth area in the violation landscape, while two different consecutive symbols indicate a rugged area.

Finally, the information content of X_I is defined as the Shannon entropy of the distribu-

tion for the blocks ab:

$$H(\lambda) = -\sum_{a \neq b} p_{ab} \log_6 p_{ab} \tag{4.2}$$

where p_{ab} is the probability of finding one of possible blocks ab, where $a \neq b$, within the sequence $\phi(\lambda)$. The logarithm base is six as this is the number of different blocks where $a \neq b$. The main intuition behind information content is that a large value of $H(\lambda)$ is an indication of perpetual changes in the violation landscape and therefore its ruggedness. Another important measure is the partial information content defined as $M(\lambda) = |\phi'(\lambda)| / (n-1)$, where $\phi'(\lambda)$ is a sequence derived from $\phi(\lambda)$ by removing all $\rightarrow$ and repeated symbols.

From $H(\lambda)$ and $M(\lambda)$, various features can be derived representing smoothness of the violation landscape. In this thesis, we study three features often used in the literature (Muñoz, Sun, et al., 2015): the maximum information content $H_{\max} = \max_\lambda\{H(\lambda)\}$, the settling sensitivity $\varepsilon_S = \log_{10}(\min_\lambda\{\lambda \mid H(\lambda) < 0.05\})$, and the initial partial information $M_0 = M(0)$. Features $H_{\max}$ and M_0 express the violation landscape smoothness (or ruggedness). Smooth landscapes have small values of $H_{\max}$ and M_0, and vice versa. The feature ε_S, on the other hand, represents the maximum change of the overall constraint violation within the original sequence.

4.1.3 Random Walk Features

A random walk through the problem search space is a well-known and often used ELA technique in single-objective optimization. A walk is an ordered sequence of solutions $X_{\mathrm{R}} = \{x^{(1)}, \ldots, x^{(n)}\}$ such that $x_i \in \mathcal{N}(x^{(i-1)}, \delta)$ for all $i \in \{2, \ldots, n\}$. During a random walk, there is no particular criterion to select the neighboring solution at each step, i.e., a random neighbor is selected (Liefooghe et al., 2020). This approach is used to describe the disjointedness of feasible regions, thus providing an insight into the local structure of feasible components.

For constrained single-objective optimization, a ratio of feasible boundary crossings, $\rho_{\partial F}$, was introduced in (Malan et al., 2015). This feature quantifies the number of boundary crossings from feasible to infeasible space and vice versa, as encountered by a random walk through the search space. Given a sequence X_{R} generated from a random walk, a binary sequence is derived $B = \{b^{(1)}, \ldots, b^{(n)}\}$ such that $b_i = 0$ iff x_i is a feasible solution. The feature $\rho_{\partial F}$ is defined as the proportion of steps in the random walk that cross the boundary of the feasible region:

$$\rho_{\partial F} = \frac{1}{n-1} \sum_{i=2}^{n} \left| b^{(i)} - b^{(i-1)} \right|. \tag{4.3}$$

Since the ratio of feasible boundary crossings does not consider the objective values, it is a feature characterizing violation landscapes. Therefore, it can be used in multiobjective optimization without any modifications (Picard & Schiffmann, 2021).

To provide a more robust measure, (Malan et al., 2015) defines $\rho_{\partial F}$ as the average value over N independent random walks. However, this aggregate value provides no local information at all. To avoid this shortcoming, we also consider the maximum and minimum values of $\rho_{\partial F}$ observed in N random walks. The resulting features are: the minimum ratio of feasible boundary crossings $(\rho_{\partial F})_{\min}$, the median ratio of feasible boundary crossings $(\rho_{\partial F})_{\mathrm{med}}$, and the maximum ratio of feasible boundary crossings $(\rho_{\partial F})_{\max}$. A value of $(\rho_{\partial F})_{\min}$ close to zero indicates that there exists at least one larger feasible (or infeasible) region. Similarly, a large value of $(\rho_{\partial F})_{\max}$ may suggests that there is an area in the violation landscape with a heavily dissected feasible component.

4.1.4 Adaptive Walk Features

In contrast to random walks, an improving neighbor is selected at each step of an adaptive walk, as a local search would do. In unconstrained single-objective optimization, adaptive walks are often used to derive various features to characterize basins of attraction of the fitness landscape. In this thesis, we extend these approaches to the violation landscape to quantitatively describe its basins of attraction. Similarly to the space-filling design, this approach is mainly used to quantify the violation multimodality, however, this time by also studying local minimum-violation components without feasible solutions.

The approach considered in this study is similar to the technique described in (Kerschke & Trautmann, 2019). However, various novel features specialized for CMOPs are introduced here for the first time. The approach consists of four steps:

1. Generate an initial sample $X_{\mathrm{A}} \subseteq S$ of solutions following a selected space-filling design.

2. Perform a local search for each solution $x \in X_{\mathrm{A}}$ to obtain local minimum-violation solutions.

3. Cluster local minimum-violation solutions to obtain approximations for local minimum-violation components and basins of attraction.

4. Find nondominated feasible solutions among the obtained local minimum-violation solutions.

Using the above approach, we can derive various features to characterize basins of attraction of the violation landscape. The resulting total number of the found clusters provides an approximation for the number of basins, $N_{\mathcal{B}}$, the first feature of this group. The proportion of all solutions from the initial sample that converge to a specific cluster is used to measure the size of the corresponding basin. The resulting features are: the smallest basin, $\mathcal{B}_{\mathrm{min}}$, the median basin, $\mathcal{B}_{\mathrm{med}}$, and the largest basin, $\mathcal{B}_{\mathrm{max}}$. Similarly, we derive three equivalent features for feasible basins (basins containing feasible solutions): the smallest feasible basin, $(\mathcal{B}_{\mathrm{F}})_{\mathrm{min}}$, the median feasible basin, $(\mathcal{B}_{\mathrm{F}})_{\mathrm{med}}$, and the largest feasible basin, $(\mathcal{B}_{\mathrm{F}})_{\mathrm{max}}$. In addition, the proportion of all solutions that converge to one of the feasible basins reflects the size of the union of all feasible basins, $\cup\mathcal{B}_{\mathrm{F}}$.

Next, the minimum constraint violation value of all local minimum-violation solutions in a cluster approximates the constraint violation of the corresponding basin defined as $v(B) = \inf\{v(x) \mid x \in \mathcal{B}\}$. Three features are considered: the median constraint violation over all basins, $v(\mathcal{B})_{\mathrm{med}}$, the maximum constraint violation of all basins, $v(\mathcal{B})_{\mathrm{max}}$, and the constraint violation of the largest basin, $v(\mathcal{B}_{\mathrm{max}})$.

Finally, the last two features considered in this group indicate the "optimality" of basins of attraction. The first feature is the "optimality" of the largest basin, $\mathcal{O}(\mathcal{B}_{\mathrm{max}})$, expressed as the number of nondominated feasible solutions in the largest cluster. The second feature is the size of the "optimal" basin, $\mathcal{B}_{\mathrm{opt}}$, expressed as the proportion of solutions that converge to the cluster with the maximum number of nondominated feasible solutions.

4.2 Test Suites Evaluation

In this section, we first discuss the experimental setup, including the specific techniques and parameter settings used to derive the ELA features. These features are then used to evaluate the existing test suites. Additionally, the effectiveness of the features in predicting MOEA performance is demonstrated on two MOEAs. Finally, we present the sensitivity

Table 4.2: Parameters used in the experimental analysis: dimension of the search space D, initial sample size for space-filling design $|X_S|$ and adaptive walk $|X_A|$, and maximum distance between two solutions ε.

D	2	3	4, 5
$\lvert X_S\rvert$	25 000	100 000	250 000
$\lvert X_A\rvert$	10 000	25 000	50 000
ε	0.02	0.04	0.12

analysis of the parameters used for feature extraction and comment on the scalability of the applied ELA approaches.

4.2.1 Experimental Setup

The proposed features are demonstrated on the test suites listed in Section 2.7. In particular, bi-objective C-DTLZ and DC-DTLZ problems were considered with the default number of constraints. Additionally, a difficulty triplet of (0.5, 0.5, 0.5) was used for the DAS-CMOP suite as this is by far the most frequently used difficulty triplet in the literature. Finally, the 11 problems from Table 2.2 were used to represent the RCM suite.

Three dimensions of the search space (2, 3, 5) were considered to measure the scalability of various techniques used to derive the ELA features. For each dimension, initial sample sizes were decided based on some initial experiments and were selected as the minimum number of solutions needed for the feature values to converge. The details are presented in Table 4.2. Note that because the real-world problems from the RCM test suite are not scalable, the parameter values used for artificial problems with five variables were used also for the four-dimensional RCM problems.

To produce the initial samples used by all the proposed techniques to derive ELA features, we used the space-filling design based on Latin hypercube sampling.

We used the Density-based Spatial Clustering of Applications with Noise (DBSCAN) (Ester et al., 1996) as the clustering algorithm for the identification of feasible components and basins, as well as to discover nondominated solutions on the boundary of the feasible region. The following DBSCAN configuration was used in the experiments: distance metric was set to the Euclidean distance, the number of samples in a neighborhood for a point to be considered a core point was set to five. At the same time, the maximum distance between two solutions for one to be considered as in the neighborhood of the other, ε, was defined based on initial experiments for each dimension separately (see Table 4.2).

To express the relationship between the objectives and constraints ($\rho_{\min}$ and $\rho_{\max}$), the Spearman's rank correlation coefficient was used.

Following the recommendation from (Muñoz, Kirley, et al., 2015), the initial sample size X_I for the information content features was set to $1\,000D$.

Simple random walks (Malan & Engelbrecht, 2014) were employed for calculating features representing ratios of feasible boundary crossings. The number of steps X_R was set to 1 000, each with maximum step size, δ, of 1% of the range of the domain for each test problem (Malan et al., 2015). Thirty independent runs were conducted to obtain the values for $(\rho_{\partial F})_{\min}$, $(\rho_{\partial F})_{\text{med}}$ and $(\rho_{\partial F})_{\max}$.

Finally, we used the truncated Newton method (Nash, 1984) as a local search procedure in adaptive walks.

4.2.2 Implementation Details

All CMOPs and techniques to derive the ELA features were implemented in the Python programming language (Van Rossum & Drake, 2009). We used the `pymoo` (Blank & Deb, 2020) implementation for CTP, DAS-CMOP and MW, while the rest of the suites were reimplemented from scratch. Next, `pymoo` was used for NSGA-III and C-TAEA, `pyDOE` (Lee, 2015) for Latin hypercube sampling, `scikit-learn` (Pedregosa & et al., 2011) for DBSCAN, `pflacco` (Prager, 2019) for the calculation of information content features, and `SciPy` (Virtanen & et al., 2020) for the truncated Newton method and to calculate Spearman's rank correlation coefficients. The rest of the functionalities, including the simple random walk, were implemented from scratch.

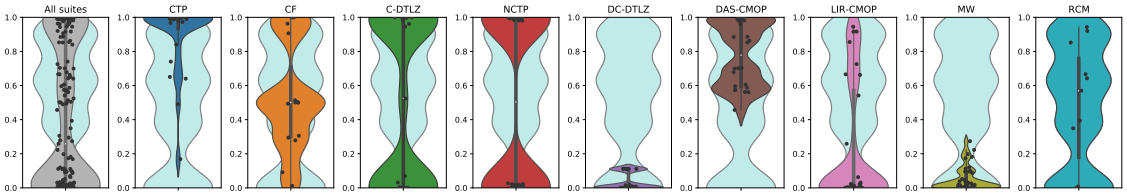
4.2.3 Results

Figures 4.1 and 4.2 show violin plots of distributions for seven selected features $\mathcal{F}_{\text{opt}}$, $\rho_{\min}$, $H_{\max}$, $N_{\mathcal{B}}$, $\mathcal{O}(\mathcal{F}_{\max})$, $(\rho_{\partial F})_{\text{med}}$ and $\mathcal{O}(\mathcal{B}_{\max})$. The first column represents the distribution for the set of all the considered CMOPs, while the rest of the columns correspond to each suite separately. Each black dot represents one problem instance. Its y-axis depicts the feature value, while the x-axis has no specific meaning and is used for better visualization only. The violin plot (colored area) approximates the probability density function for the feature distribution. For example, Figure 4.1b shows there are more problem instances in the CF suite (third column, violin plot in orange) with $\rho_{\min} \approx 0$ than those with $\rho_{\min} \approx -0.75$. In addition, there are no problem instances with $\rho_{\min} \approx -1$. The violin plot in light blue behind each suite corresponds to the distribution of RCM in Figure 4.1, and in light gray to the distribution of the set of all CMOPs in Figure 4.2. The idea is to show the coverage of RCM characteristics by the artificial test suites (Figure 4.1) and to expose the differences between a single test suite and the rest of the suites (Figure 4.2).

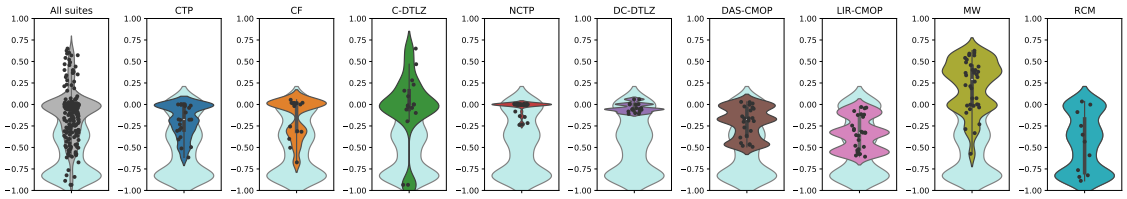
In addition, Figure 4.3 shows two heat maps depicting the coverage of RCM characteristics by the artificial test suites, and the coverage of characteristics present in the set of all the considered CMOPs. The coverage is expressed as one minus the mean value of all distances between the feature values from the RCM suite (or the set of all CMOPs) and the nearest feature value from a selected single suite. Note that feature values need to be normalized to $[0, 1]$ before calculating the coverage values. A coverage value near one indicates excellent coverage. This means that for each feature value from RCM (or the set of all CMOPs), there exists at least one problem from the selected single suite whose feature value is close to that one from RCM (or the set of all CMOPs). In Figure 4.3, a lighter color indicates that a feature value is well covered by a test suite, and vice versa. For example, in Figure 4.3a, $N_{\mathcal{F}}$ (first row), as represented by RCM, is well covered by all the artificial suites, while $\mathcal{F}_{\max}$ (fourth row) is poorly covered, especially by DC-DTLZ and MW.

According to (Bartz-Beielstein et al., 2020), a good benchmark suite should "include the difficulties that are typical of real world instances of the problem class under investigation". To evaluate whether the considered artificial test suites contain the characteristics observed in real-world CMOPs, we use 11 continuous and low-dimensional problems from the RCM suite and the coverage metric proposed in this subsection. Note that the low number of considered real-world problems limits the generality of the conclusions that can be drawn from these results. For example, we cannot claim that some problem characteristics that are missing from these 11 problems are in fact unrealistic. Nevertheless, we can identify which characteristics of these real-world problems are not present in the artificial problems.

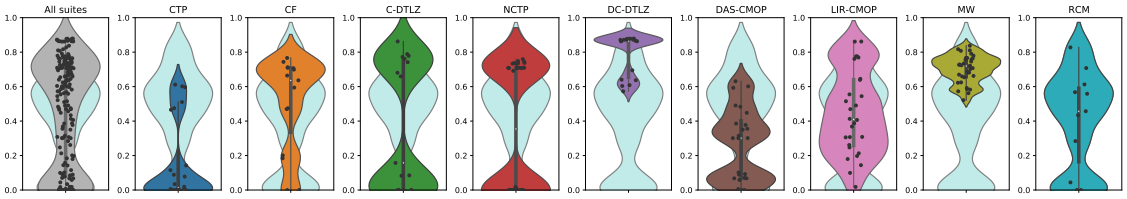
The results show that several features characterizing feasible components are not adequately covered by the studied artificial test suites (Figure 4.3a, features from $N_{\mathcal{F}}$ to



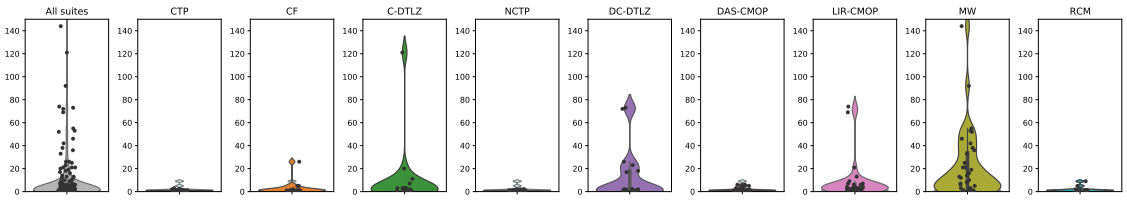
(a) Size of the optimal feasible component, $\mathcal{F}_{\text{opt}}$



(b) Minimum correlation, ρ_{min}



(c) Maximum information content, H_{max}



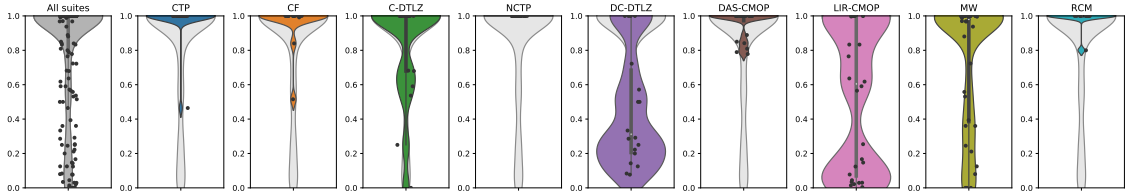
(d) Number of basins, $N_{\mathcal{B}}$

Figure 4.1: Violin plots depicting distributions of selected features for the set of all the considered CMOPs and for each suite separately. The violin plot in light blue behind each suite corresponds to the RCM suite.

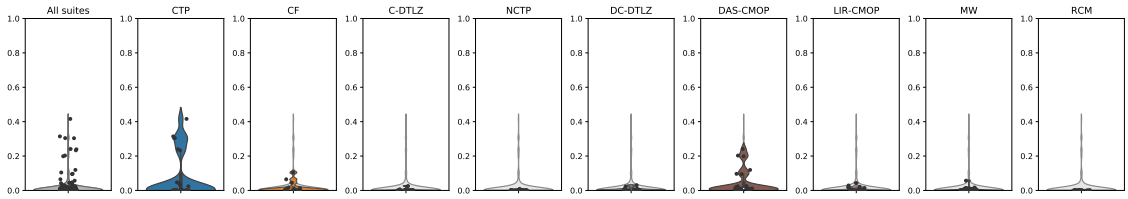
$\mathcal{F}_{\text{opt}}$). Especially the recently proposed artificial test suites NCTP, DC-DTLZ, LIR-CMOP, and MW insufficiently represent problem characteristics of the studied RCM problems. For example, many of the considered RCM problems have only one feasible component, while recent artificial test problems have many feasible components. As a consequence, the RCM problems result in feasible components of any size, while, for example, the MW suite contains problems with rather small feasible components only. This shortcoming of the artificial test suites can be observed nicely for $\mathcal{F}_{\text{opt}}$ (Figure 4.1a).

Similarly, the existing artificial test suites fail to satisfactorily represent the conflicts between the objectives and the overall constraint violation. As we can see in Figure 4.1b, strong negative correlations, $\rho_{\text{min}} \in [-1.00, -0.75]$, observed in the studied RCM problems, are found in the C-DTLZ suite only. We can argue that a general multiobjective optimization algorithm would most likely struggle to maintain good convergence and diversity when the objectives and the overall constraint violation are strongly negatively correlated. However, our findings show that the studied artificial test suites do not cover this particular characteristic well.

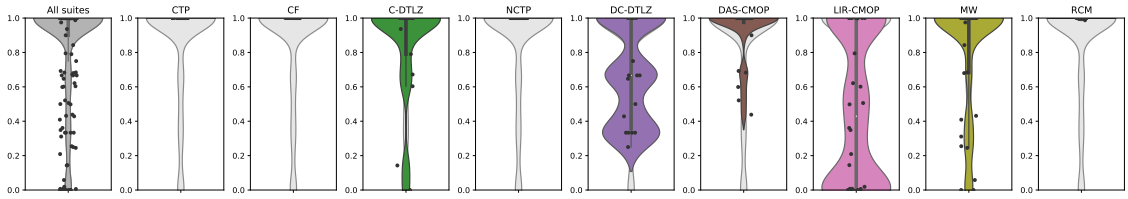
The results also indicate that the feasibility ratio and information content features are



(a) Proportion of Pareto-optimal solutions in largest feasible component, $\mathcal{O}(\mathcal{F}_{\max})$



(b) Median ratio of feasible boundary crossings, $(\rho_{\partial F})_{\text{med}}$



(c) Proportion of Pareto-optimal solutions in largest basin, $\mathcal{O}(\mathcal{B}_{\max})$

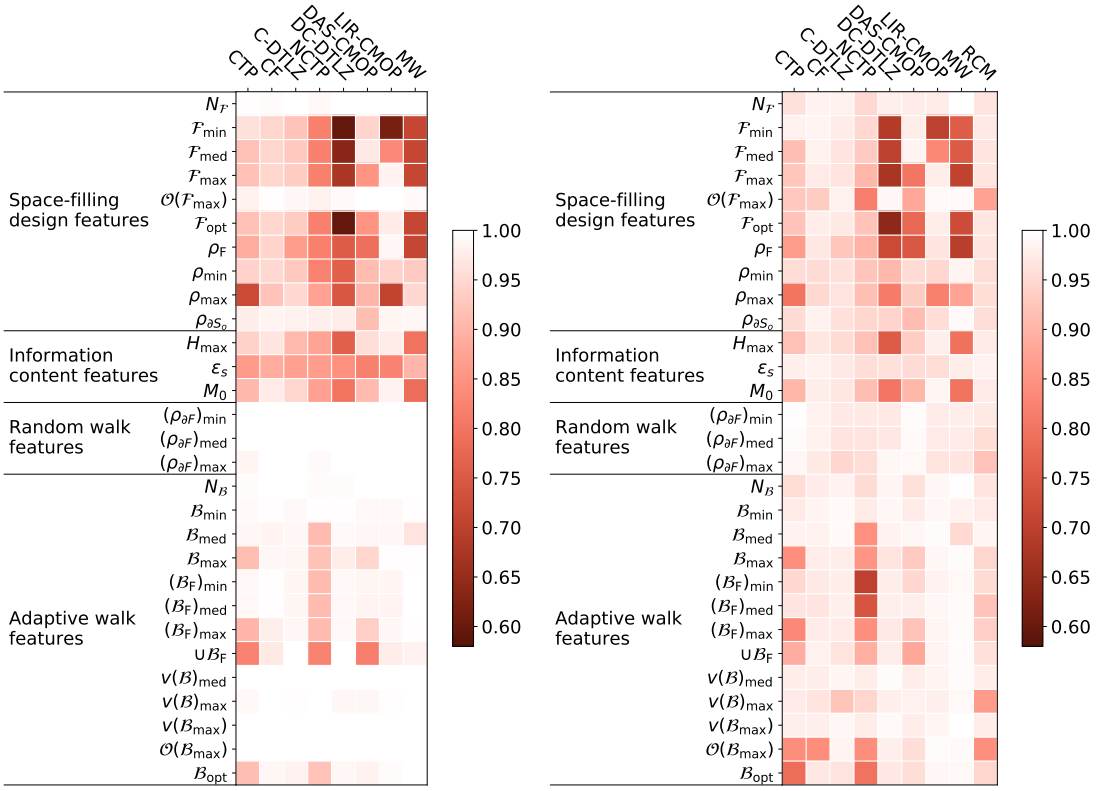
Figure 4.2: Violin plots depicting distributions of selected features for the set of all the considered CMOPs and for each suite separately. The violin plot in light gray behind each suite corresponds to the set of all CMOPs.

poorly covered by the artificial test suites except for CFs and LIR-CMOPs (Figure 4.3a, features ρ_{F}, $H_{\max}$, ε_s and M_0). While the RCM problems take a full range of feature values, the artificial test problems have often extremely small or large feasible ratios and extremely smooth or very rugged violation landscapes (Figure 4.1c).

On the other hand, the features related to basins of attraction are mostly covered well by the studied artificial test suites (Figure 4.3a, features from $N_{\mathcal{B}}$ to $\mathcal{B}_{\text{opt}}$). The suites that are not diverse enough considering the basin-related features are CF, NCTP, and DAS-CMOP. Additionally, the ratios of feasible boundary crossings are well covered by the considered artificial test suites, while $\rho_{\partial S_o}$ is very well covered by all the artificial test suites except for DAS-CMOP.

Finally, the studied artificial test suites have many basin-related characteristics that cannot be observed in the selected RCM problems. While the considered RCM problems often contain only one basin where all Pareto-optimal solutions are located, many recently proposed artificial test problems have several basins, and the Pareto-optimal solutions are spread among multiple basins (Figures 4.1d, 4.2a and 4.2c). This is in line with the recent development in constrained multiobjective optimization where the primary source of complexity is severe violation multimodality (e.g., Figure 3.1f) (Filipič & Vodopija, 2021). However, our study suggests that other relevant aspects, such as the sizes of feasible components and basins of attraction, negative correlation among the objectives and overall constraint violation, diverse ruggedness and feasibility ratios, are frequently insufficiently addressed by the literature.

An additional analysis is devoted to measuring whether the considered test suites contain a wide variety of problems with different characteristics. This is done in the same way as



(a) Coverage of RCM

(b) Overall coverage

Figure 4.3: Heat maps showing the coverage of landscape characteristics.

the comparison with RCM, except that in this case the test suites are contrasted against the set of all CMOPs. The results are shown in Figure 4.3b.

Interestingly, the results are not drastically different from the previous ones, indicating that RCM contains problems with various characteristics. For example, the coverage of most of the space-filling design and information content features is very similar (Figure 4.3b, features from $N_{\mathcal{F}}$ to M_0). This is also true for seven features concerning basins of attraction $\mathcal{B}_{\text{med}}$, $\mathcal{B}_{\max}$, $(\mathcal{B}_{\text{F}})_{\min}$, $(\mathcal{B}_{\text{F}})_{\text{med}}$, $(\mathcal{B}_{\text{F}})_{\max}$, $\cup\mathcal{B}_{\text{F}}$ and $\mathcal{B}_{\text{opt}}$, except for the NCTP suite which has significantly worse coverage when contrasted against the set of all CMOPs.

Within the space-filling design, the only true exception is $\mathcal{O}(\mathcal{F}_{\max})$. The coverage for this feature is significantly worse when comparing the suites against the set of all CMOPs. This happens because RCM does not cover $\mathcal{O}(\mathcal{F}_{\max})$ particularly well (Figure 4.2a), which has already been explained before. As we can see, this aspect is even more drastically manifested for $\mathcal{O}(\mathcal{B}_{\max})$ (Figure 4.2c).

Finally, the coverage of random walk features is slightly worse when the suites are compared against the set of all CMOPs (Figure 4.3b, features $(\rho_{\partial F})_{\min}$, $(\rho_{\partial F})_{\text{med}}$ and $(\rho_{\partial F})_{\max}$). This is true because RCM contains only problems with small values of these particular features. However, in general, this set of features is well covered. Almost all the problems have only small ratios of feasible boundary crossings except for a few problems from CTP, CF and DAS-CMOP (Figure 4.2b). The same observation holds for the space-filling design feature $N_{\mathcal{F}}$, and for the following adaptive walk features $N_{\mathcal{B}}$, $\mathcal{B}_{\min}$, $v(\mathcal{B})_{\text{med}}$, $v(\mathcal{B})_{\max}$ and $v(\mathcal{B}_{\max})$.

In summary, we can see that all the studied test suites have some advantages and some limitations. For this reason, we recommend to include CMOPs from various suites to construct a comprehensive and sound benchmark.

4.2.4 Analyzing the Diversity of Test Suites of CMOPs

In this subsection, we employ the ELA features to express and discuss the diversity of the test suites of CMOPs. This is achieved by firstly computing the landscape features and then employing the t-distributed Stochastic Neighbor Embedding (t-SNE) (van der Maaten & Hinton, 2008) to embed the 29-D CMOP feature space into the 2-D space.

The t-SNE algorithm is a popular nonlinear dimensionality reduction technique designed to represent high-dimensional data in a low-dimensional space, typically the 2-D plane. First, it converts similarities between data points to distributions. Then, it tries to find a low-dimensional embedding of the points that minimizes the divergence between the two distributions that measure neighbor similarity—one in the original space and the other in the projected space. This means that t-SNE tries to preserve the local relationships between neighboring points, while the global structure is generally lost.

Finding the best embedding is an optimization problem with a non-convex fitness function. To solve it, t-SNE uses a gradient descent method with a random starting point, which means that different runs can yield different results. The output of t-SNE depends also on other parameters, such as the *perplexity* (similar to the number of nearest neighbors in other graph-based dimensionality reduction techniques), *early exaggeration* (separation of clusters in the embedded space) and *learning rate* (also called ε). The gradients can be computed exactly or estimated using the Barnes-Hut approximation, which substantially accelerates the method without degrading its performance (van der Maaten, 2014).

For dimensionality reduction, we used the t-SNE implementation from the `scikit-learn` Python package (Pedregosa & et al., 2011) with default parameter values. That is, we used the Euclidean distance metric, random initialization of the embedding, perplexity of 30, early exaggeration of 12, learning rate of 200, the maximum number of iterations of 1000, and the maximum number of iterations without progress before aborting of 300. The

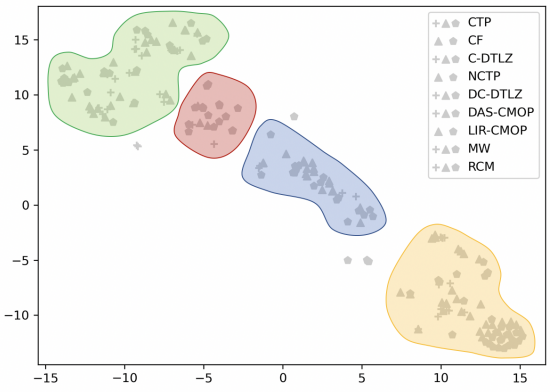


Figure 4.4: Embedding of the feature space as obtained by t-SNE. The four regions are depicted in green, red, blue, and orange. The points that are not contained in any region are considered to be outliers.

gradient was computed by the Barnes-Hut approximation with the angular size of 0.5. For more details, see (Vodopija, Tušar, et al., 2021).

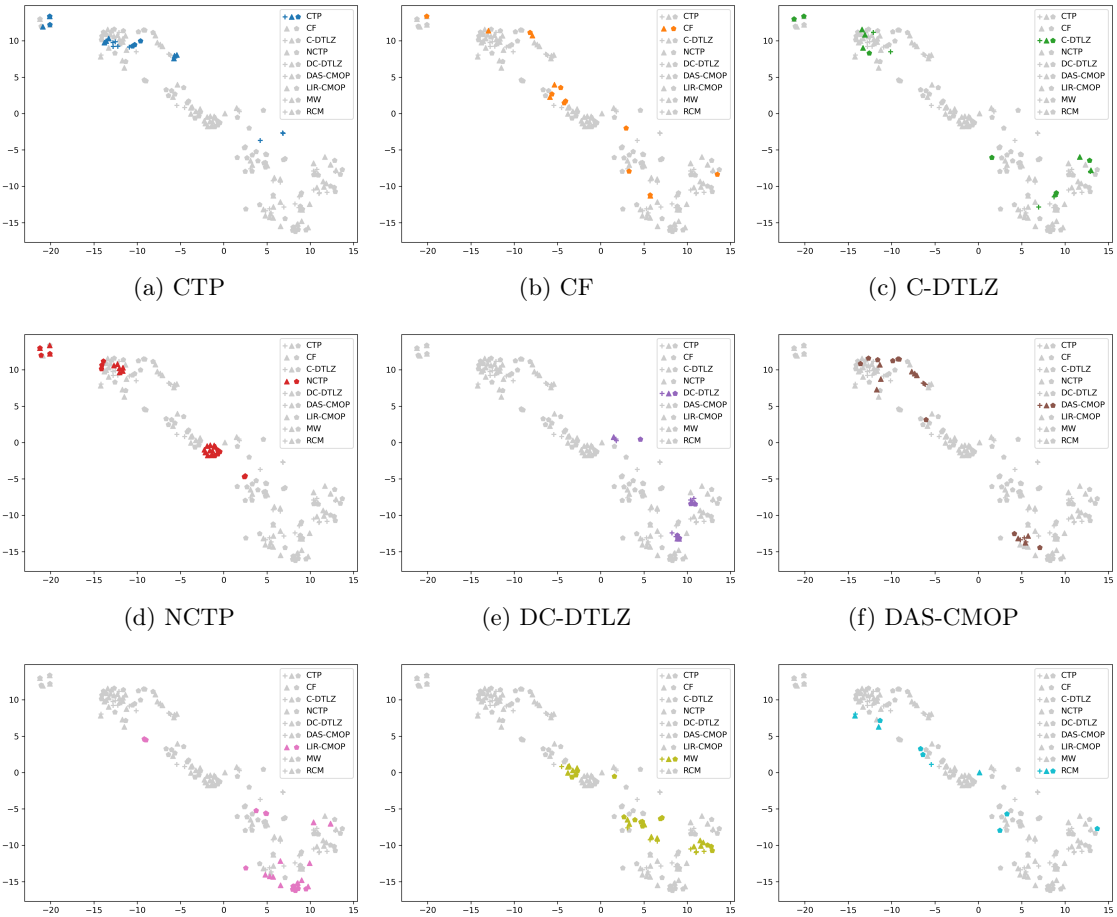
The results obtained by t-SNE are shown in Figures 4.4 and 4.5. Specifically, the figures show the 2-D embedding of the 29-D feature space consisting of the landscape features presented in Table 4.1. Each subfigure in Figure 4.5 corresponds to one of the test suites. For example, Figure 4.5a exposes the embedding of the CTP suite in blue, while the gray points correspond to the rest of the test suites. Points in the shape of a plus (+) correspond to CMOPs with two variables, points with a shape of a triangle (▲) to CMOPs with three variables, and points with a shape of a pentagon (⬟) to CMOPs with five variables.

An additional analysis shows that the embedding of the feature space can be, based on the corresponding characteristics, split into four regions: green, red, blue and yellow (Figure 4.4). The green region corresponds to CMOPs with severe violation multimodality, small basins of attraction, and rugged violation landscapes. The red region corresponds to CMOPs with moderate violation multimodality, rugged violation landscapes, and small feasibility ratios. The blue region corresponds to relatively low violation multimodality, rugged violation landscapes, small feasibility ratios, and positive correlations between objectives and constraints. Finally, the yellow region corresponds to unimodal CMOPs with large feasible components, smooth violation landscapes, and large feasible regions.

As we can see from Figure 4.5a, almost all CTP problems are located in the orange region. Therefore, many relevant characteristics are poorly represented by CTP, e.g., violation multimodality, small feasibility ratios, etc. Similarly, NCTP fails to sufficiently represent severe multimodality since it contains no problems from the green region (Figure 4.5d). On the other hand, DC-DTLZ, LIR-CMOP, and MW are biased towards highly multimodal violation landscapes or those with small basins of attraction (Figure 4.5e, Figure 4.5g, and Figure 4.5h). Nevertheless, MW is one of the most diverse suites considering other characteristics (Figure 4.5h).

The C-DTLZ and DAS-CMOP suites are mainly located in the green and orange regions and fail to sufficiently represent the characteristics of the red and blue regions.

Finally, the results show that CF and RCM are well spread through the whole embedded feature space (Figure 4.5b and Figure 4.5i). As we can see, they have at least one representative CMOP instance in each region. Therefore, CF and RCM are the most diverse test suites according to the employed landscape features.



(a) CTP (b) CF (c) C-DTLZ

(d) NCTP (e) DC-DTLZ (f) DAS-CMOP

(g) LIR-CMOP (h) MW (i) RCM

Figure 4.5: Embedding of the feature space as obtained by t-SNE. Each subfigure exposes the embedding of a selected suite.

4.2.5 Feature-Based MOEA Performance Prediction

To analyze the relationship between the proposed ELA features and empirical problem hardness, we built a regression model to predict the performance of two well-known MOEAs: the NSGA-III algorithm and a more recently proposed C-TAEA, both equipped with their default constraint handling techniques, as presented in Sections 2.3 and 2.4. The MOEA performance was measured as the difference between the hypervolume of the Pareto front and the hypervolume of the Pareto front approximation generated by an algorithm. In more detail, both MOEAs were run with a population of 200 solutions for 500 generations. For every 50 generations (10% of the total budget), the difference in hypervolume between the obtained front and the Pareto front was recorded. These differences in hypervolume represented the multi-target variables for the regression analysis. We selected the multi-target approach as we wanted to study the relationship between the proposed features and MOEA performance for different budgets. The features from Table 4.1 and three additional dimensionality features (D, M, and J) represented the predictor variables.

To build regression models, we used the well-known Random forest algorithm (Breiman, 2001). This approach was selected as it is known to generate reliable models without hyperparameter tuning or feature selection (Mersmann et al., 2011), and can be used to measure the feature importance. We used the Random forest implementation from the `scikit-learn` Python package (Pedregosa & et al., 2011) with default parameter values. The evaluation of the models was performed using a leave-one-suite-out cross-validation, i.e., the models were repeatedly trained on all the suites except one and then tested on problems from the unseen suite. The results in terms of mean absolute error (MAE), mean square error (MSE), and coefficient of determination (R^2) are reported in Table 4.3. The scores are averaged over multiple outputs of the regression model and all the suites. Values for MAE and MSE close to 0 indicate good performance of the model. In contrast, R^2 reaches 0 for the baseline model, which always predicts the mean of the differences in hypervolume, and 1 when the model exactly matches the observed values. If the model performs worse than the baseline model, R^2 is less than 0.

The results show the impact of the considered groups of features. As we can see, the most impactful features for both MOEAs are those derived from the space-filling design and adaptive walks. Less effective are features related to the local structure of the violation landscape, namely information content, and random walk features. The least valuable features are those associated with the dimensionality of the problems. Indeed, the models trained solely with dimensionality features perform worse than the baseline model. Finally, the best overall results are obtained using all the features.

Additionally, Figure 4.6 shows the relative importance based on the mean decrease in impurity of the ELA features at predicting the MOEA performance. As we can see, the most important features are similar for both MOEAs. In both cases, the most informative feature is $\mathcal{B}_{\text{med}}$ showing that basin sizes contain relevant information. Next in the ranking are the features expressing the correlations between the objectives and constraints (ρ_{min} and ρ_{max}), and feasibility ratio (ρ_{F}). This additionally confirms the findings already discussed in (Malan et al., 2015; Picard & Schiffmann, 2021). Then come the features related to the local structure of the violation landscape, M_0 and $(\rho_{\partial F})_{\text{max}}$ for NSGA-III, and M_0 and H_{max} for C-TAEA.

Interestingly, the features characterizing feasible components ($N_{\mathcal{F}}$, $\mathcal{F}_{\text{min}}$, $\mathcal{F}_{\text{med}}$, $\mathcal{F}_{\text{max}}$, $\mathcal{O}(\mathcal{F}_{\text{max}})$, and $\mathcal{F}_{\text{opt}}$) are important at predicting C-TAEA performance, while they appear irrelevant for predicting NSGA-III performance. A possible explanation is that NSGA-III has difficulties in escaping from local minimum-violation components without feasible solutions, thus the distribution of the basins has a higher impact on the performance of NSGA-III. In contrast, C-TAEA avoids local minimum-violation components without

Table 4.3: Leave-one-suite-out cross-validation results: Mean absolute error (MAE), mean square error (MSE), and coefficient of determination (R^2). The mean (μ) and standard deviation (σ) calculated over all the suites are reported.

NSGA-III			
Feature group	MAE $[\mu \pm \sigma]$	MSE $[\mu \pm \sigma]$	R^2 $[\mu \pm \sigma]$
Space filling	0.081 ± 0.032	0.019 ± 0.017	0.619 ± 0.075
Info. content	0.103 ± 0.039	0.030 ± 0.025	0.353 ± 0.238
Random walk	0.101 ± 0.038	0.028 ± 0.024	0.457 ± 0.111
Adaptive walk	0.085 ± 0.022	0.018 ± 0.010	0.617 ± 0.026
Dimensionality	0.131 ± 0.028	0.049 ± 0.025	−0.108 ± 0.237
All	0.066 ± 0.020	0.010 ± 0.006	0.787 ± 0.013
C-TAEA			
Feature group	MAE $[\mu \pm \sigma]$	MSE $[\mu \pm \sigma]$	R^2 $[\mu \pm \sigma]$
Space filling	0.090 ± 0.039	0.024 ± 0.022	0.623 ± 0.059
Info. content	0.122 ± 0.061	0.048 ± 0.053	0.294 ± 0.283
Random walk	0.121 ± 0.072	0.048 ± 0.050	0.320 ± 0.199
Adaptive walk	0.105 ± 0.051	0.037 ± 0.035	0.457 ± 0.114
Dimensionality	0.134 ± 0.070	0.060 ± 0.072	−0.091 ± 0.495
All	0.070 ± 0.012	0.011 ± 0.004	0.851 ± 0.002

feasible solutions efficiently, therefore, the distribution of feasible components is more informative for its performance.

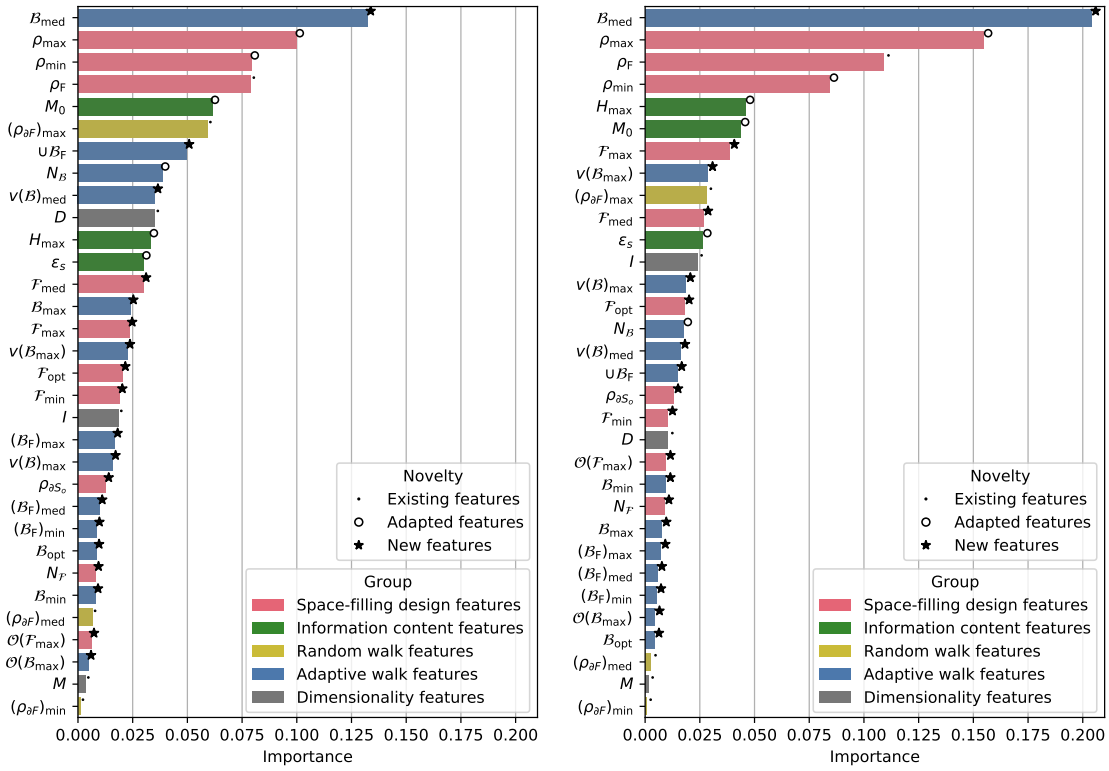
Finally, the results show that none of the dimensionality features is among the most important ones. The most relevant at predicting NSGA-III performance is D, while for C-TAEA it is J.

4.2.6 Sensitivity Analysis

The crucial step in deriving space-filling design and adaptive walk features is the accurate identification of feasible components and basins of attraction, respectively. Once the feasible components and basins are identified, all other features can be unequivocally derived. On the other hand, if the clustering step incorrectly determines the feasible components and basins of attraction, the rest of the features are also affected. The identification of both feasible components and basins of attraction strongly depends on two parameters: initial sample size ($|X_\mathrm{S}|$ and $|X_\mathrm{A}|$) and the maximum distance between two solutions ε used by the DBSCAN algorithm.

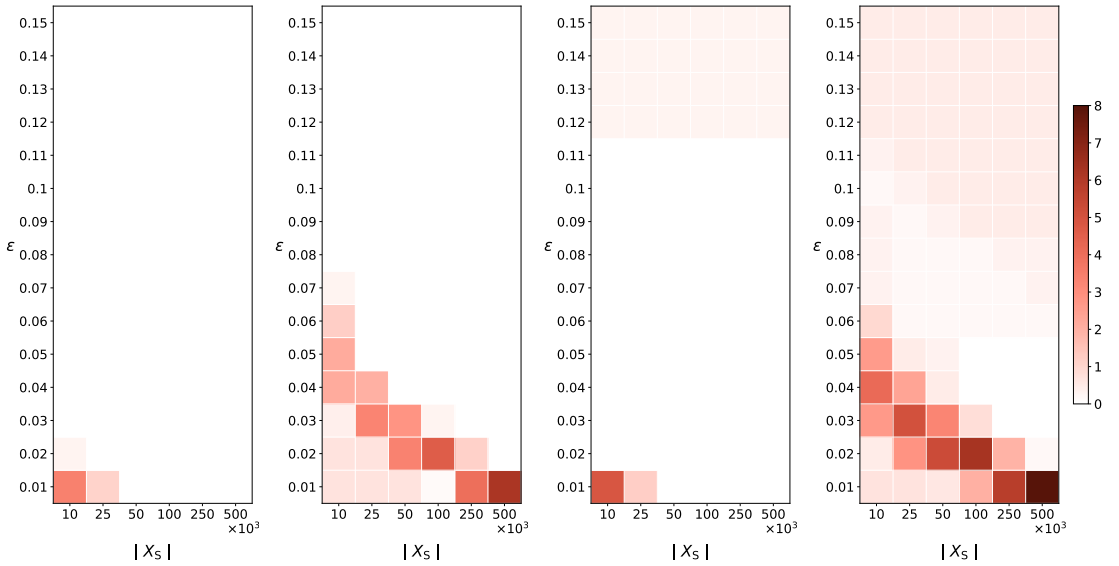
The goal is to find the smallest sample size and the smallest ε that would still yield reliable results. Small sample sizes are preferred since they require fewer function evaluations. Additionally, a small ε is necessary to not erroneously merge two or more disjoint feasible components (or basins of attraction). On the other hand, a too small ε can result in the separation of feasible components (or basins) into two or more components (or basins). For this reason, a careful selection of these parameters is necessary.

Space-filling design features are more sensitive to parameter setting than adaptive walk features. All infeasible solutions in the space-filling design approach are discarded and used only for enumeration. On the other hand, almost all initial solutions for adaptive walk features produce local-minimum violation solutions, thus facilitating the clustering procedure. For this reason, we investigate the sensitivity of space-filling design only. The presented analysis is very similar for adaptive walk features, except that smaller initial sample sizes are required in this case.



(a) NSGA-III (b) C-TAEA

Figure 4.6: Feature importance based on mean decrease in impurity for MOEA performance prediction: (a) NSGA-III and (b) C-TAEA. Bars correspond to the mean importance value averaged over all budgets, the additional marks denote feature novelty (see Table 4.1).



(a) C2-DTLZ2 2-D (b) C2-DTLZ2 3-D (c) DAS-CMOP1 2-D (d) DAS-CMOP1 3-D

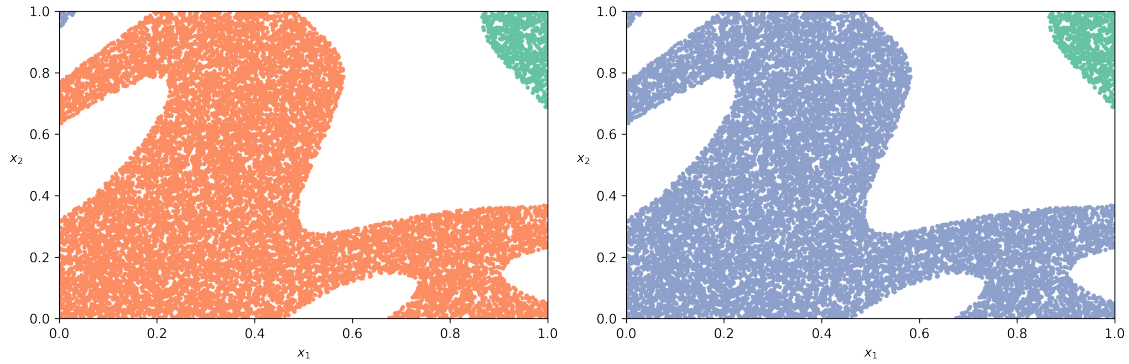
Figure 4.7: Median log differences between the exact number of feasible components and the derived number of clusters for each combination of $|X_S|$ and ϵ for the 2-D and 3-D problems C2-DTLZ2 and DAS-CMOP1.

The goal of the presented analysis is to measure the parameter sensitivity for the identification of feasible components. More specifically, we study the extraction of the number of feasible components $N_{\mathcal{F}}$ with respect to 1) different parameter values and 2) repetition of the experiments, since a stochastic approach is used to obtain the initial sample. To address 1), we test the space-filling design approach using all combinations of the following parameter values: $|X_S| \in \{10\,000, 25\,000, 50\,000, 100\,000, 250\,000, 500\,000\}$ and $\varepsilon \in \{0.01, 0.02, \ldots, 0.14, 0.15\}$. For 2), we repeat all the experiments 30 times, each time with a different initial sample. The analysis is performed for 2-D and 3-D, as the correctness of the derived components can be visually checked, while this is not possible for 5-D.

The sensitivity analysis is performed on two CMOPs: C2-DTLZ2 and DAS-CMOP1. On the one hand, C2-DTLZ2 represents a simple problem concerning feature extraction. It has three well-shaped feasible components that are spaced apart (Figure 3.1b). On the other hand, the feasible components of DAS-CMOP1 have irregular shapes, and the small one in the upper left corner is located close to the large one in the middle (Figure 3.1h). This is precisely what makes the identification of components hard for DBSCAN. The 2-D DAS-CMOP1 problem with two variables has three components, while the DAS-CMOP1 with three variables has five components.

The results are shown in the form of heat maps in Figure 4.7, where the median logarithmic difference between the exact number of feasible components and the derived number of clusters is depicted. The values are normalized with the number of feasible components and logarithmic to obtain better visualization. Smaller values in lighter colors indicate better performance, and vice versa. Specifically, white color indicates a perfect match.

As we can see from Figure 4.7a, for C2-DTLZ2 2-D the exact number of feasible components can be obtained for all $|X_S| \geq 25\,000$ and all $\varepsilon \geq 0.02$, while for DAS-CMOP1 2-D already the combination $|X_S| = 10\,000$ and $\varepsilon = 0.02$ is sufficient to identify the feasible



(a) $\varepsilon = 0.02$ (b) $\varepsilon = 0.12$

Figure 4.8: Feasible components for DAS-CMOP1 as identified by space-filling design approach for $\varepsilon = 0.02$, $\varepsilon = 0.12$ and $|X_S| = 25\,000$. Different colors correspond to disjoint feasible components. For $\varepsilon = 0.12$, we can see that the small component in the upper left corner is merged with the large one in the middle.

components (Figure 4.7c). Interestingly, for DAS-CMOP1 2-D we can see that for $\varepsilon \geq 0.12$ one component is recognized incorrectly. What happens is that the small component in the top left corner is merged with the large one due to the oversized ε (Figure 4.8). Nevertheless, a combination of $|X_S| = 25\,000$ and $\varepsilon = 0.02$ proved to be the most reliable for 2-D, resulting in a perfect match for all 30 repetitions on C2-DTLZ2 and DAS-CMOP1. Moreover, Figure 4.8 shows that DBSCAN is really effective also when dealing with CMOPs with irregularly-shaped feasible components.

For 3-D, Figures 4.7b and 4.7d clearly show we need larger $|X_S|$ and ε to obtain the exact number of components. For both CMOPs a combination of $|X_S| = 100\,000$ and $\varepsilon = 0.04$ proved to be the most effective, correctly matching all the components in all 30 repetitions. Additionally, we can see that for DAS-CMOP1 two or more components are incorrectly merged for $\varepsilon \geq 0.06$. An interesting observation is that the technique seems to work better for smaller initial sample sizes when using $\varepsilon \leq 0.02$. The reason behind this behavior is that DBSCAN finds no cluster at all in such cases and the number of identified feasible components is thus zero.

In conclusion, for feasible components that are spaced apart, the space-filling design and similarly adaptive walk approaches are highly reliable and insensitive to small changes in parameter values. Furthermore, the performance is not affected by the shape of the feasible components. On the other hand, for problems with feasible components that are placed close together, the techniques are more sensitive to the selection of parameter values.

A discussion on the parameter sensitivity for information content and random walks can be found in (Malan et al., 2015; Muñoz, Kirley, et al., 2015). For this reason, we just briefly summarize the key findings. In (Muñoz, Kirley, et al., 2015), a comprehensive sensitivity analysis was performed to measure the robustness of the information content approach. Among others, the authors addressed the effect of varying the size of X_I. The experimental results showed that already a sample size of $100D$ proved to be robust for deriving information content features. Similarly, Malan et al. (2015) demonstrated that the random walk approach was reliable for deriving the ratio of feasible boundary crossings. This was shown on a wide variety of violation landscapes by obtaining small standard deviations while performing the experiments 30 times.

Finally, the space-filling design and information content methods do not require additional function evaluations except those from the initial sample. For this reason, the number

of required function evaluations equals $|X_S|$ and $|X_I|$, respectively. Similarly, the random walk approach requires the same number of function evaluations as there are steps in the walk, $|X_R|$. The exact number of function evaluations for space-filling design, information content, and random walks can be found in Subsection 4.2.1. In contrast, in the proposed adaptive walk approach, the Newton method requires additional function evaluations for each solution from the initial sample. The total number of function evaluations required by the adaptive walk approach is $n_f\,|X_A|$ in the median case, where n_f is 7.8, 12.4, and 16.1 for D values of 2, 3, and 5, respectively.

4.2.7 A Note on Scalability

For problems with up to five variables the derived ELA feature values closely coincided with actual values. In contrast, it was virtually impossible to investigate high-dimensional CMOPs since the size of the initial sample used by the space-filling design techniques grows exponentially. Indeed, a more detailed experimental analysis showed that to obtain meaningful results, we would need at least 2 000 000 solutions for problems with seven variables and 15 000 000 solutions for problems with ten variables. This is not unexpected since it is a well-known consequence of the curse of dimensionality.

In addition, the feasible components of the artificial test problems often have the shape of an ellipsoid (Figures 3.1b and 3.1e). As the dimension of the search space increases, a feasible component (ellipsoid) becomes an insignificant volume relative to that of the decision space (rectangular area). For this reason, it is very hard to obtain relevant results using space-filling design techniques for such large-scale problems.

It should also be noted that although the proposed ELA features are appropriate to characterize problems with up to five variables, the required number of function evaluations for space-filling design and adaptive walk approaches is significant even for low-dimensional CMOPs. This implies that these features cannot be derived for expensive real-world applications such as simulation-based optimization.

Chapter 5

Characterization of CMOPs: Performance Space Perspective

This chapter provides an extension of ERDs to constrained multiobjective optimization, and discusses an approach to measure the problem's effectiveness to distinguish algorithms based on the distance between ERDs (Vodopija et al., 2024) in Section 5.1. Then, in Section 5.2, it presents the experimental setup, and discusses the existing test suites of CMOPs and their potential in differentiating algorithms.

5.1 Algorithm Performance Measurement

5.1.1 Quality Indicator for CMOPs

There are two main paradigms to approach constrained optimization problems. The first one is applicable when the constraints must be satisfied at any cost, while the second one allows for partial violation of constraints if the objective values of a solution are of good quality. In practice, the first paradigm applies when it is known in advance that particular requirements, such as the geometrical limitations of the design, must be met at any cost. On the other hand, the second paradigm is useful when certain requirements can be partially violated. This flexibility is acceptable when it significantly improves the objectives and we are willing to relax some constraints. In the literature, the second paradigm has been the subject of more extensive research, which is why we consider it in our work[1].

Furthermore, in many real-world scenarios, the objective values cannot be calculated if the solution is infeasible (Eiben & Smith, 2003). This often happens in simulation-based optimization where the simulator cannot return meaningful results if some of the constraints are not satisfied. Consequently, our main assumption in constructing a quality indicator for constrained multiobjective optimization is that an infeasible solution is strictly worse than any feasible solutions regardless of the objective value quality—this is exactly the pillar of the first paradigm. For example, in Figure 5.1c, solution z^4 has better objective values than z^5. Actually, if no constraints were considered, z^4 would dominate z^5. Nevertheless, z^5 is considered to be strictly better than z^4. This desired property of a quality indicator can be mathematically expressed as follows:

$$I(x) < I(y) \quad \text{for all } (x, y) \in F \times S \backslash F. \tag{5.1}$$

The quality indicator for CSOPs (2.14) does not satisfy this property. The biggest disadvantage of an indicator not satisfying (5.1) is that no matter how small the quality

[1]The COCO framework uses the second paradigm for CSOPs.

indicator value is, we cannot know whether there exists a feasible solution in A^T. For example, for a certain CSOP there might exist a solution in A^T with $f(x) = 0$ and an arbitrarily small overall constraint violation value. In other words, unless I^{CSOP} equals zero ($x^* \in A^T$), we cannot know for certain whether we found a feasible solution relying solely on the quality indicator values. From a practical point of view, we wish for a quality indicator to involve a threshold that unequivocally indicates when the algorithm reached the feasible space.

Considering this, we propose an extension of the quality indicator for MOPs (2.15) as follows:

$$I^{\text{CMOP}}(A^T) = \begin{cases} \min_{x \in A^T} v(x) + \tau^* & A^T \cap F = \emptyset \\ \min(I^{\text{MOP}}(A^T \cap F), \tau^*) & \text{otherwise} \end{cases} \tag{5.2}$$

where τ^* is a threshold to indicate that the feasible space was reached. For example, in the COCO framework, it can be set to the largest considered target for MOPs, which equals 1. It is also obvious that I^{CMOP} satisfies the property (5.1).

The behavior of the proposed quality indicator is illustrated in Fig. 5.1, depicting the three stages of the algorithm search. Initially, in the first stage (Fig. 5.1a), all solutions generated by the algorithm are infeasible, shown as contained in the gray areas. During this phase, I^{CMOP} measures the extent to which these solutions violate the constraints, as defined in the first term of (5.2). The algorithm then progresses to a stage where it finds at least one feasible solution (Fig. 5.1b), corresponding to the second term in (5.2), a situation already discussed in Section 2.6.

Additionally, note that in (5.2) only feasible solutions are considered in the calculation of I^{MOP}. This can be seen in Figs. 5.1b and 5.1c where the infeasible solutions are not considered for the calculation of the distance and hypervolume, respectively, once feasible solutions have been found.

5.1.2 Performance Space Comparison

According to (Bartz-Beielstein et al., 2020), a good test suite should include problems that "enable the user to tell the algorithms apart in the performance space". To measure the ability of a given problem to differentiate among MOEAs, we rely on the area between the corresponding ERDs (see Figure 5.2, area in gray). The intuition is that a large area between ERDs indicates large differences between the runtimes and, in turn, between the algorithms.

Based on the area between the two ERDs, we propose a metric, Δ, that provides information on the similarity between algorithms and their performance using a single number. Let us assume we are dealing with two algorithms a and b, and we have their corresponding runtime data sets for a certain optimization problem $\{T_a(\tau)\}_\tau$ and $\{T_b(\tau)\}_\tau$. Then, the area of a single segment between the runtimes (in the logarithmic scale) for the same target can be obviously calculated as follows (see Figure 5.2, the bold line between runtimes):

$$\frac{|\log(T_a(\tau)) - \log(T_b(\tau))|}{n_\tau}, \tag{5.3}$$

where n_τ is the number of targets. In particular, when a certain runtime is missing, we set its value to the maximal budget (number of function evaluations). This is done for calculation purpose and has no particular meaning. Using the formula (5.3), the area bounded by the two ERDs and thus Δ can be expressed as the sum of these segment areas over all the targets:

$$\Delta(a, b) = \frac{\sum_\tau \left| \log\left(\frac{T_a(\tau)}{T_b(\tau)}\right) \right|}{n_\tau \log n_f}, \tag{5.4}$$

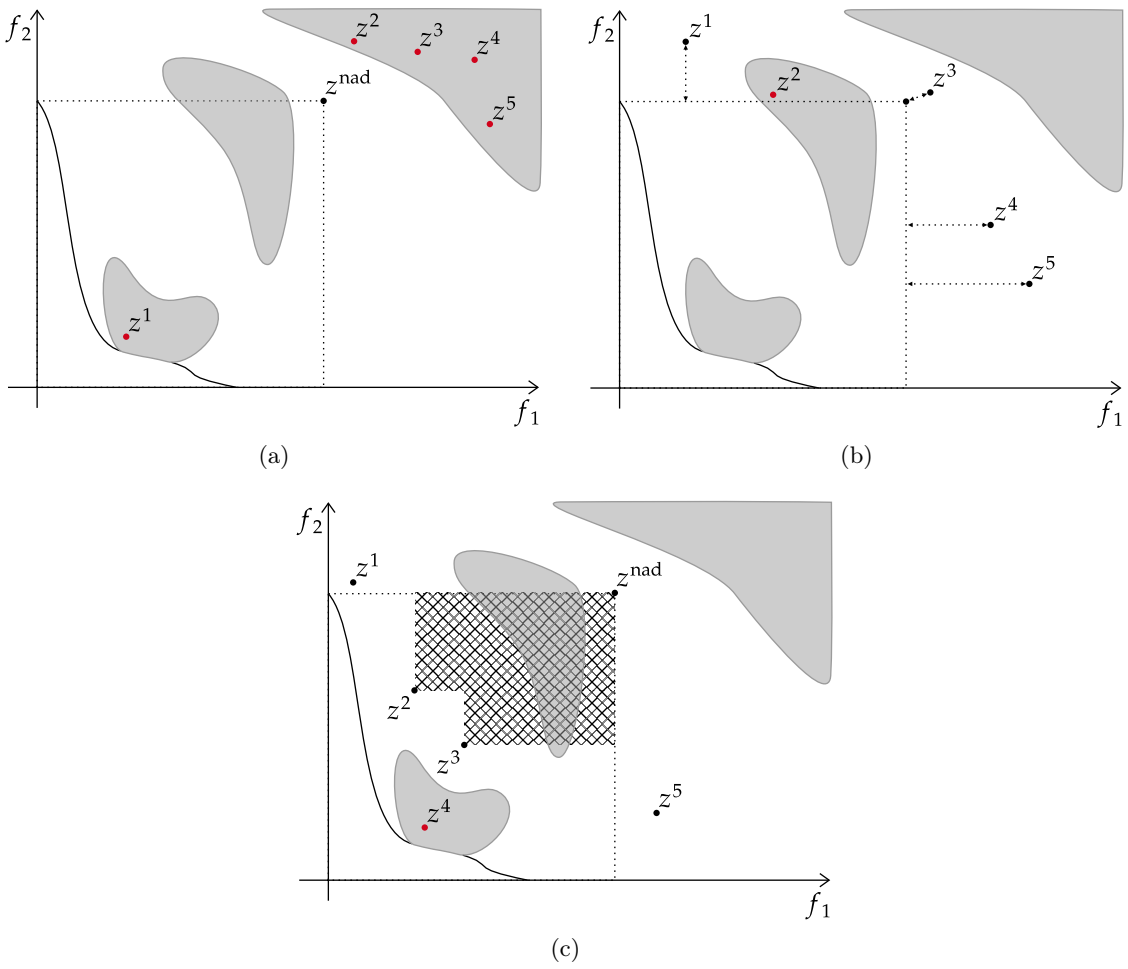


Figure 5.1: The quality indicator I^{CMOP} at three stages of the algorithm search: (a) All the solutions belong to the infeasible space (areas in gray) and the quality indicator relies on the overall constraint violation. (b) There exists at least one feasible solution but no solutions dominate the reference point z^{nad}. The quality indicator relies on the distance to the region of interest Z (area bounded by the dotted lines and the coordinate axes). (c) There exists at least one feasible solution dominating the reference point. The quality indicator is based on the hypervolume (area depicted with a mesh).

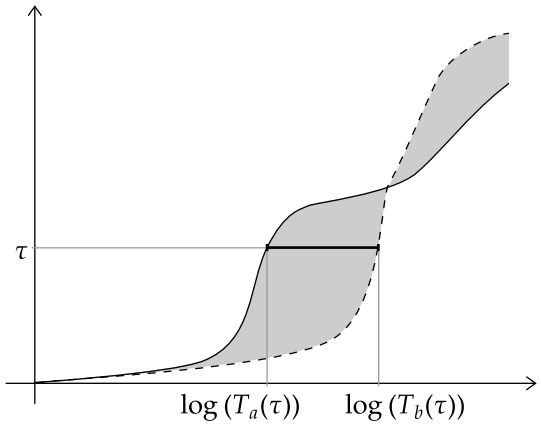


Figure 5.2: ERDs corresponding to algorithms a (solid line) and b (dashed line). The area between the lines (in gray) represents the difference in algorithm performance, $\Delta(a, b)$.

where n_f is the number of function evaluations. The formula is additionally divided by $\log n_f$ for normalization purposes. It can be easily seen that $\Delta(a, b) \in [0, 1]$ for all algorithms and problems. In particular, small values indicate similar behavior of the chosen algorithms, and vice versa. For example, in the extreme case, algorithm a might solve all the targets within a single evaluation, while algorithm b is unable to reach any target. In this case $\Delta(a, b) = 1$. In the second extreme case all the runtimes coincide and $\Delta(a, b) = 0$.

The $\Delta(a, b)$ metric can be additionally expressed as:

$$\Delta(a, b) = \frac{n_{\tau^-}}{n_\tau}\Delta^-(a, b) + \frac{n_{\tau^+}}{n_\tau}\Delta^+(a, b) \tag{5.5}$$

where Δ^- and Δ^+ represent the sum of segments (5.3) over targets measuring constraint satisfaction, τ^-, and targets expressing the algorithm effectiveness in approximating the Pareto front (called front approximation for short), τ^+, respectively. Additionally, n_{τ^+} is the number of τ^+ targets, and n_{τ^-} the number of τ^- targets. In particular, Δ^- can be seen as a measure of algorithm differences in constraint satisfaction, while Δ^+ is a metric measuring differences in front approximation.

5.2 Test Suites Evaluation

5.2.1 Experimental Setup

The proposed performance assessment is demonstrated on the artificial test suites listed in Table 2.1. In particular, three-objective C-DTLZ and DC-DTLZ problems were considered with the default number of constraints. Additionally, a difficulty triplet of (0.5, 0.5, 0.5) was used for the DAS-CMOP suite, as this is by far the most frequently used difficulty triplet in the literature. Three dimensions of the search space $D \in \{5, 10, 30\}$ were used to evaluate the proposed performance assessment methodology and compare the test suites. All the Pareto fronts can be analytically expressed and the corresponding hypervolume values exactly calculated.

Given that we wish to measure the capability of CMOPs to differentiate among optimization algorithms, we experiment with three well-known MOEAs with diverse operators, selection mechanisms and, most importantly, CHTs. The rationale for this approach is that if these distinct algorithms perform similarly on a CMOP, the CMOP is not a good choice for

Table 5.1: The population size n_{p} and number of generations n_{g} used in the experimental analysis, based on the number of objectives M and the dimension of the search space D.

	$M = 2$			$M = 3$		
D	5	10	30	5	10	30
n_{p}	200	200	200	300	300	300
n_{g}	300	600	1800	200	400	1200

benchmarking algorithms. The chosen algorithms are NSGA-III, C-TAEA and MOEA/D-IEpsilon, all equipped with their default CHTs and discussed in Sections 2.3 and 2.4.

All MOEAs were run with an equal population size, n_{p}, and the same number of generations, n_{g}. In particular, the population size was set to $n_{\mathrm{p}} = 100M$. The number of generations was set to $n_{\mathrm{g}} = 120D/M$ and was selected as approximately the minimum value to obtain convergence for all the MOEAs (in total, $12000D$ function evaluations). Note, the division by M in the expression for n_{g} is necessary to enable aggregation over CMOPs with different numbers of objectives. The resulting values for n_{p} and n_{g} are shown in Table 5.1.

Other parameters of the algorithms and their operators were set to their default values (Deb & Jain, 2014; Fan, Li, Cai, Huang, et al., 2019; K. Li et al., 2019): Polynomial mutation was used in all the MOEAs. The mutation probability was set to $1/D$ and the distribution index to 20. Specifically, simulated binary crossover was used in NSGA-III and C-TAEA with a crossover probability of 1 and distribution index of 30. In contrast, a differential-evolution-based crossover was used in MOEA/D with a crossover probability of 1 and scaling factor of 0.5. Additionally, in MOEA/D, the neighborhood size was set to 30, the probability of neighborhood mating to 0.9, the maximum number of solutions replaced by a child to 2, τ to 0.1, α to 0.95, T_{c} to 0.8, and θ to $0.05n_{\mathrm{p}}$.

Finally, ERDs were computed without employing restarts or bootstrapping.

5.2.2 Target Precision Values for CMOPs

The values of the distance metric, d, defined in (2.17), as well as those of the overall constraint violation, v, can result in different magnitudes for different CMOPs. Consequently, it is impossible to define a set of target precision values that would provide meaningful results for all the studied CMOPs. As we wanted to compare different CMOPs and suites, we first sampled 100 solutions $\{x^i\}_i$ for each CMOP and normalized d and v as follows:

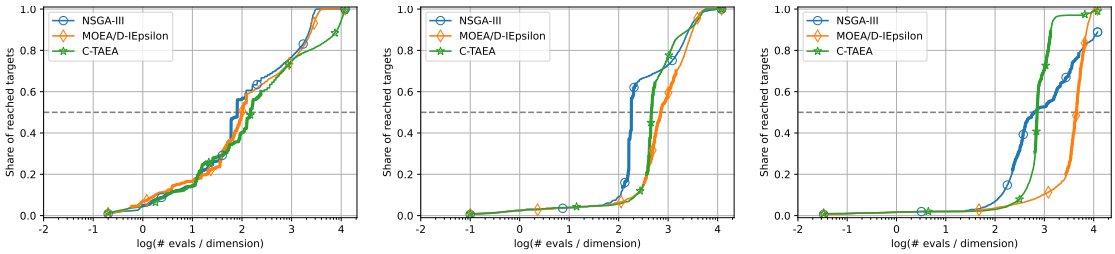
$$\widetilde{d} = d/10^{\lceil \log(\mathrm{med}(\{d^i\}_i)) \rceil} \tag{5.6}$$

and

$$\widetilde{v} = d/10^{\lceil \log(\mathrm{med}(\{v^i\}_i)) \rceil} \tag{5.7}$$

where $\mathrm{med}(\{d^i\}_i)$ and $\mathrm{med}(\{v^i\}_i)$ are median values of the sets $\{d^i\}_i = \{d(x^i, Z)\}_i$ and $\{v^i\}_i = \{v(x^i)\}_i$, respectively.

After applying this procedure and performing several experiments, we set τ^* to 1. Additionally, a good set of target precision values for I^{CMOP} corresponds to $\tau(\varepsilon) = \tau^{\mathrm{ref}} + \varepsilon$, where $\varepsilon \in \{10^k \mid k \in \{-5, -4.9, \ldots, 0\}\} \cup \{1 + 10^k \mid k \in \{-5, -4.9, \ldots, 0\}\}$. The first half of the target precision values τ^+ apply to the feasible solutions and represent how well the algorithm approximates the Pareto front, while the second half of the targets τ^- is used to understand the algorithm constraint satisfaction performance.



(a) MW12 (5-D) (b) MW12 (10-D) (c) MW12 (30-D)

Figure 5.3: Empirical runtime distribution aggregated over multiple runs for the MW12 problem for the three MOEAs and dimension 5, 10 and 30.

5.2.3 Implementation Details

All of the CMOPs and MOEAs, and the performance measurement procedure were implemented in the Python programming language (Van Rossum & Drake, 2009). We used the `pymoo` (Blank & Deb, 2020) implementations for CTP, DAS-CMOP, MW, NSGA-III, C-TAEA and the hypervolume calculation. The rest of the suites, MOEA/D-IEpsilon and other functionalities were implemented from scratch.

5.2.4 Results

Let us first look at the results for a single problem. As an example we select the MW12 problem. Figure 5.3 shows the ERDs for this problem aggregated over 30 runs for each of the three algorithms. To aid visualization and comparison, the function evaluations (x-axis) are divided by problem dimension and shown in logarithmic scale.

The horizontal dashed line divides the targets into τ^- and τ^+. Note that this intuition is true only for a single run without aggregation. Nevertheless, if an ERD (single or aggregated) starts above this line, then all the algorithm runs started in the feasible region—the first initialized solution is feasible. On the other hand, if an ERD starts below the dashed line and never crosses it, then this indicates the corresponding algorithm runs never reached the feasible region. Moreover, the line is thicker when some, but not all, runs have found feasible solutions.

As we can see, each MOEA reaches most of the targets in the two smaller dimensions, while NSGA-III fails to reach some targets on 30-D problems. All the algorithms are able to find feasible solutions in all of the runs, with NSGA-III being generally quickest in this regard.

The aggregated results over all problems of a suite are shown in Figs. 5.4 and 5.5 on the left hand side of each subfigure. For example, Figure 5.4a shows the ERDs for the selected MOEAs aggregated over all problems from the CTP suite in 5-D. On the right hand side of each subfigure, we see violin plots approximating the distributions for Δ^+ (top left), Δ^- (bottom left), and Δ (right) values. Each of these values was computed for 30 runs of each pair of algorithms on each problem and is represented in the violin plot as a black dot. The y-axis depicts the Δ^+, Δ^-, or Δ value, while the x-axis has no specific meaning and is used solely for better visualization. For example, Figure 5.5l shows there are more problem instances in the MW suite with $\Delta \approx 0.05$ than those with $\Delta \approx 0.15$. In addition, there are no problem instances with $\Delta > 0.25$.

To help better understand the violin plots, it may be helpful to consider that the distribution of Δ^+, Δ^-, and Δ in an ideal suite should be skewed towards higher values. This would suggest that the majority of the problems in the suite are effective in differentiating

among algorithms. For instance, when examining MW in 30-D, we can see that this suite consists of a wide range of CMOPs, including a sufficient number of CMOPs resulting in high values of Δ^+, Δ^-, and Δ (Figure 5.5l). In contrast, the distribution for Δ^- and suite CTP in 30-D is concentrated around 0, indicating that these CMOPs are not effective in differentiating among algorithms with respect to constraint satisfaction (Figure 5.4c).

As already discussed in Subsection 5.1.2, a well-designed test suite should include a wide variety of problems that can differentiate among MOEAs (Δ). Since we are dealing with constrained problems, we are particularly interested in evaluating the ability of the problems to differentiate among the algorithms with respect to constraint handling (Δ^-).

- CTP: As we can see, for all problems the algorithms start in the feasible space (Figs. 5.4a, 5.4b, and 5.4c). The main difficulty they face is front approximation. This is additionally confirmed by the violin plots showing that Δ^- equals 0 for all dimensions.

- CF: Unlike for CTPs, the MOEAs find no feasible solutions in the very beginning of the evolutionary process (Figs. 5.4d, 5.4e, and 5.4f). As it can be seen from the violin plots, the difference in algorithm performance originates from both constraint satisfaction and front approximation. However, the performance is quite similar except for a few CMOPs in 10-D and 30-D.

- C-DTLZ: From the performance space perspective, this suite is well-designed. The algorithms struggle to find feasible solutions in the initial phase of the evolutionary process (Figs. 5.4g, 5.4h, and 5.4i). In addition, the suite can differentiate between algorithms in both constraint satisfaction and front approximation, as depicted in the corresponding violin plots.

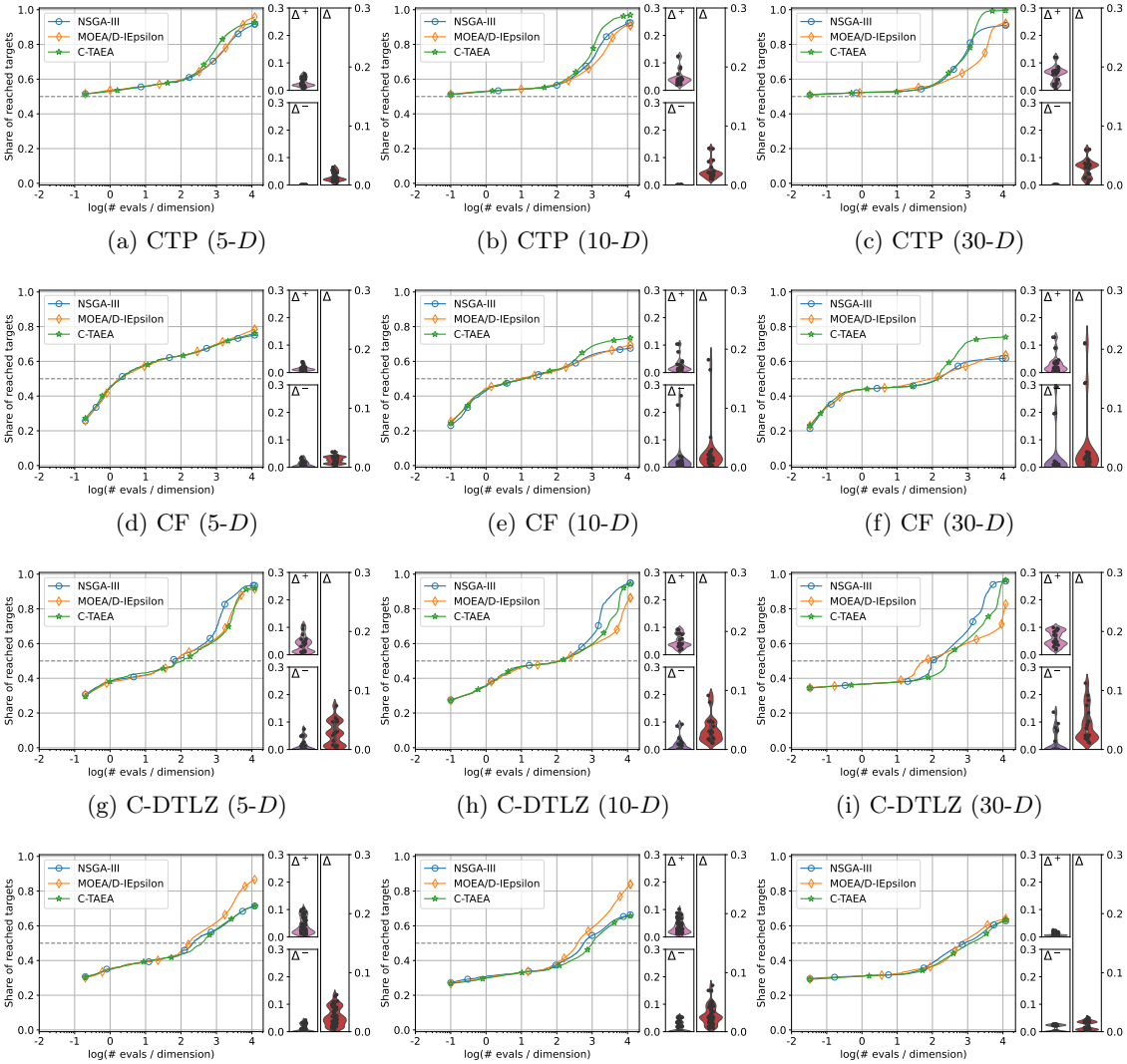
- NCTP: Although the three MOEAs need a large number of function evaluations to reach a feasible region, their main challenge is front approximation (Figs. 5.4j, 5.4k, and 5.4l). Moreover, as shown in the violin plots, the vast majority of the difference in algorithm performance also comes from front approximation.

- DC-DTLZ: We can observe that all the three MOEAs struggle to obtain feasible solutions, which are discovered only late in the evolutionary process (Figs. 5.5a, 5.5b, and 5.5c). Like CF, the DC-DTLZ suite can differentiate the constraint satisfaction of algorithm performance, as well as the front approximation. However, the differences in algorithm performance are small (see the violin plots).

- DAS-CMOP: As we can see, feasible solutions are quickly discovered by all the three algorithms (Figs. 5.5d, 5.5e, and 5.5f). Moreover, algorithm performance differences are almost exclusively obtained in front approximation, since $\Delta^- \approx 0$ for all problems and MOEAs.

- LIR-CMOP: The performance space characteristics of this suite are very similar to those of NCTPs. Although the studied algorithms need some time to find feasible solutions, the main difference in the algorithm performance is contained in the front approximation phase, as shown in the violin plots (Figs. 5.5g, 5.5h, and 5.5i).

- MW: From the performance space perspective, MW is one of the most versatile and well-designed artificial test suites found in the literature. It is the best among the studied suites in differentiating the three MOEAs (Figs. 5.5j, 5.5k, and 5.5l). Moreover, as shown in the violin plots, the algorithm performance is diverse in both constraint satisfaction and front approximation.



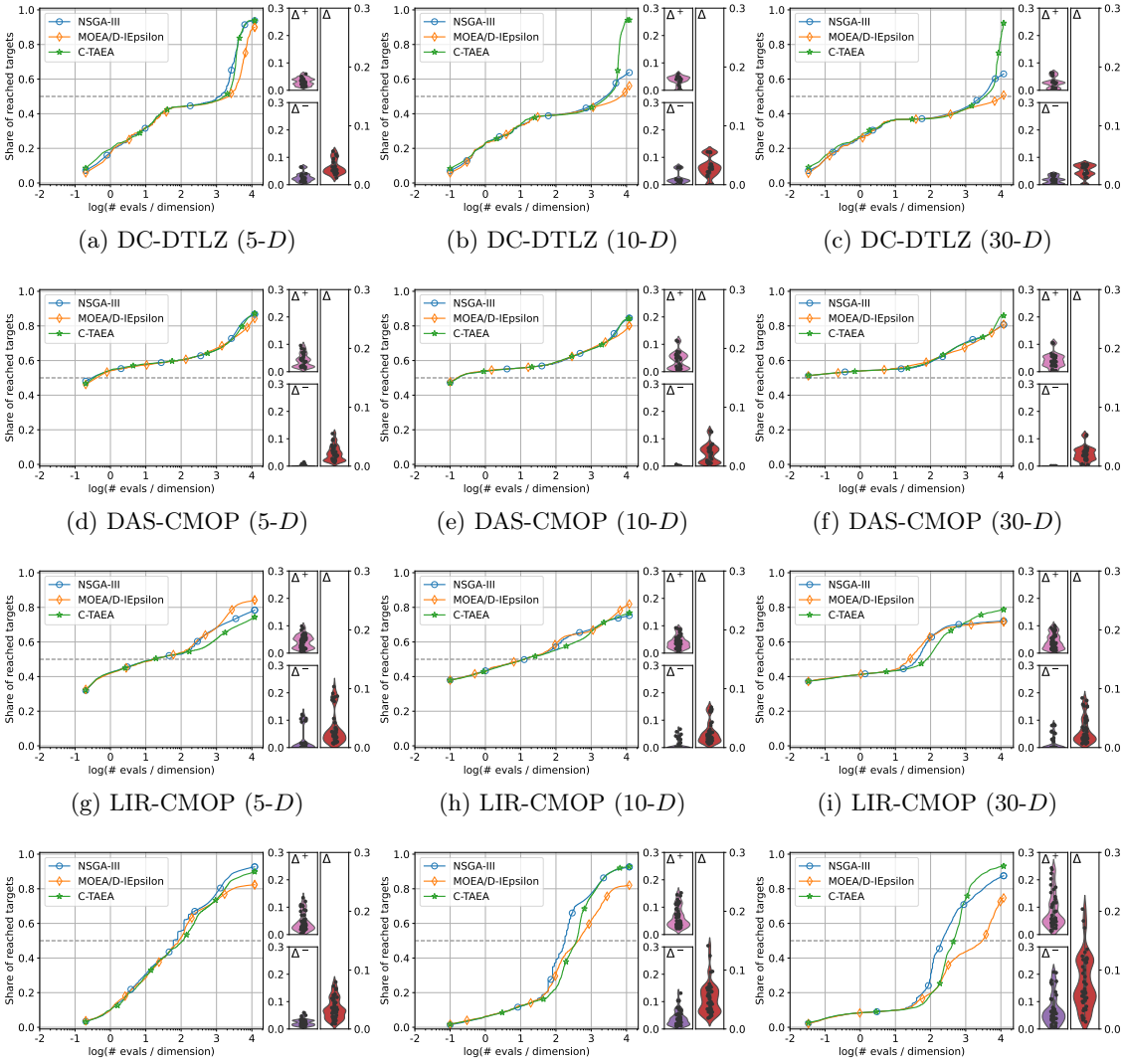
(a) CTP (5-D) (b) CTP (10-D) (c) CTP (30-D)

(d) CF (5-D) (e) CF (10-D) (f) CF (30-D)

(g) C-DTLZ (5-D) (h) C-DTLZ (10-D) (i) C-DTLZ (30-D)

(j) NCTP (5-D) (k) NCTP (10-D) (l) NCTP (30-D)

Figure 5.4: Results of the three MOEAs on CMOPs from CTP, CF, C-DTLZ, and NCTP suites. The left plot of each subfigure shows empirical runtime distribution aggregated over all CMOPs in the suite and all targets in dimension 5, 10 and 30. On the right of each subfigure, violin plots depict distributions of Δ^+ (top left), Δ^- (bottom left), and Δ (right) values. The larger the diversity, the better.



(a) DC-DTLZ (5-D) (b) DC-DTLZ (10-D) (c) DC-DTLZ (30-D)

(d) DAS-CMOP (5-D) (e) DAS-CMOP (10-D) (f) DAS-CMOP (30-D)

(g) LIR-CMOP (5-D) (h) LIR-CMOP (10-D) (i) LIR-CMOP (30-D)

(j) MW (5-D) (k) MW (10-D) (l) MW (30-D)

Figure 5.5: Results of the three MOEAs on CMOPs from DC-DTLZ, DAS-CMOP, LIR-CMOP, and MW suites. The left plot of each subfigure shows empirical runtime distribution aggregated over all CMOPs in the suite and all targets in dimension 5, 10 and 30. On the right of each subfigure, violin plots depict distributions of Δ^+ (top left), Δ^- (bottom left), and Δ (right) values. The larger the diversity, the better.

5.2.5 Limitations

We see two potential limitations of the proposed methodology to evaluate the performance space. Firstly, the results can be affected by the selection of the algorithms and their budgets, and secondly, the choice of target precision values also has a great impact on the outcome.

To alleviate the first issue, we selected three different MOEAs equipped with distinct CHTs. Additionally, the number of function evaluations was set large enough to ensure convergence, thus revealing the deviations between the algorithms. Finally, the logarithmic scale was applied to the budget to not bias the results towards the tail of the convergence graphs where the algorithms have already converged.

On the other hand, we were not able to satisfactorily address the issue of some targets having greater impact than others. For example, there is no assurance that progressing from target $1 + 10^{-4.2}$ to target $1 + 10^{-4.3}$ is equally important or difficult as progressing from target $1 + 10^{-4.3}$ to target $1 + 10^{-4.4}$ for all the problems and algorithms at hand. Nevertheless, using the target approach and logarithmic scale to define these targets is argued to be much more efficient in comparing algorithm performance than just relying on regular convergence graphs (Hansen et al., 2022).

Part II

Real-World Case Studies

Chapter 6

Design of Cyclone Dust Separators

This chapter shows how the proposed methodology for characterizing CMOPs can be applied to the cyclone design problem. In Section 6.1, we present the cyclone design problem formulation and particularly focus on its constraints (Vodopija, Breiderhoff, et al., 2021). Then, the feature space analysis is presented in Section 6.2 and the performance space analysis in Section 6.3. Finally, Section 6.4 puts the two perspectives together.

6.1 Problem Formulation

Cyclone dust separators (see an example in Figure 6.1) are frequently used devices to remove dispersed particles from a flue gas. The principle of cyclone dust separation is simple: the gas-solid composite enters at the top section tangentially. The cylindrical body induces a spin and a highly swirling flow pattern to the gas-dust composite. The centrifugal force separates the dust particles from the gas stream. The dust is moved to the wall of the cylinder and goes down the conical section and further on to the dust outlet, while the clean gas exits trough the outlet pipe.

The main advantage of these cyclones is their simple structure that leads, of course, to low cost and ease of operation. Moreover, cyclones are the only devices that can operate under high pressure, high temperature, and in harsh operating environments. The absence of small and moving parts make cyclone separators robust and one of the most efficient dust separators. Nevertheless, their disadvantage is a low efficiency for small particles. For this reason, emission regulations encourage designers to optimize the efficiency of cyclones. Every cyclone manufacturer has their own design, but most designers follow widely accepted rules for cyclone design (Hoffmann & Stein, 2007; Weiß, 1986).

6.1.1 Cyclone Design Optimization

From an engineering perspective, cyclone performance is measured by collection efficiency (CE) and pressure drop (PD), which are typically derived from simulation and/or analytical models (Cortes & Gil, 2007). CE needs to be maximized and PD minimized. CE describes the amount or fraction of particles filtered from the gas. PD is the difference of pressures between inlet and outlet and occurs with frictional forces. It can be thought of as the amount of energy required to move the gas through the separator and is therefore directly related to cyclone operation costs. Several additional parameters enable a designer to describe the entire cyclone optimization problem. These are geometric shape, properties of the carrying gas like flow rate and viscosity, cut-off diameter (also called “cut size” or “cut diameter”) and properties of the particles like density and distribution.

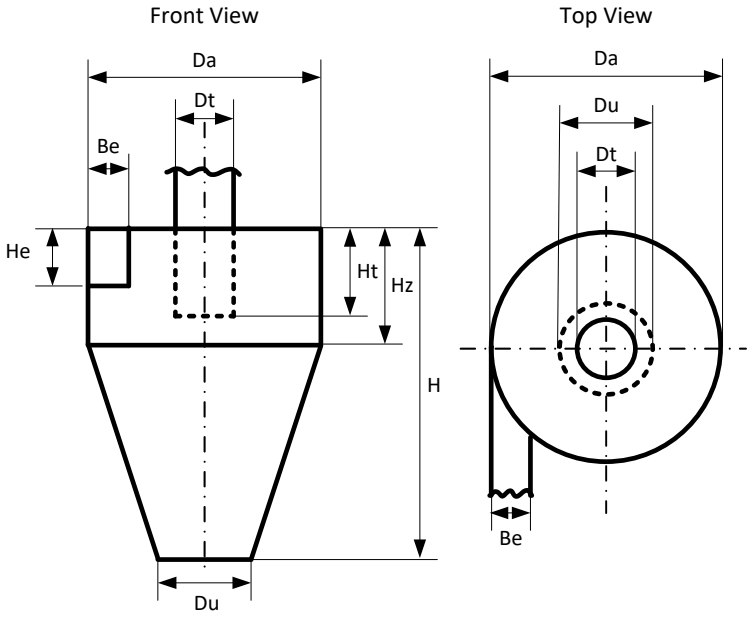


Figure 6.1: Reverse-flow cyclone.

Table 6.1: Geometric parameters of reverse-flow cyclone.

Name	Symbol
Cyclone diameter	D_a
Outlet pipe diameter	D_t
Total cyclone height	H
Outlet pipe immersion	H_t
Inlet section height	H_e
Inlet section width	B_e
Cylindrical body height	H_z
Dust discharge diameter	D_u

Cyclones exist in different shapes but the reverse-flow model presented in Figure 6.1 is the one most commonly used in industry (Weiß, 1986). Reverse-flow cyclone separators are cone shaped. Gas enters at the top of the separator body, flows downwards, then flows back upwards and is discharged. The shape is essentially determined by eight parameters. It depends on the simulation model which of them are needed to calculate CE and PD. In our approach, only six parameters are required. These are the cyclone diameter, D_a, and total height, H, the outlet pipe diameter, D_t, and immersion, H_t, as well as width, B_e, and height, H_e, of the inlet area (see Table 6.1 and Figure 6.1). The tendency of following the design standards is confirmed by a large number of cyclone layouts in the literature (Hoffmann & Stein, 2007). For example, (Weiß, 1986) gives a list of high-efficiency cyclone sizes depending on the flow rate of the incoming gas stream. The geometric parameters of the reverse-flow cyclones used in our experiments were calculated from these instances.

The cyclone geometry, together with the volumetric gas flow rate, defines the cut-off diameter of the cyclone. This is the size of the particles that are separated from the gas stream with the efficiency of 50%. Particles larger than the cut-off diameter are removed with a higher efficiency, and smaller particles with a lower efficiency. Particle composition

is described by density, concentration, and particle size distribution. To get an idea of the right particle properties for our test cyclone instances, we use the cut-off diameter as the main indicator to obtain an estimate of the particle size distribution. The chosen properties and distribution are taken from a common test dust distribution ("Dust distributions," n.d.).

6.1.2 Analytical Model

The simulation of reverse-flow cyclones has been the subject of many different models and approaches (Cortes & Gil, 2007). To obtain CE, we focus on the model proposed in (Barth, 1956). The authors provide a simple but still useful model by which friction was introduced in cyclone modeling for the first time. Some assumptions can be considered reasonable enough to obtain a good compromise between accurate prediction and simplification of the equations. The model is based on the balance of two forces acting on a particle rotating in the cyclone body: the centrifugal force acting towards the wall and the drag acting towards the outlet pipe. The cut-off diameter, i.e., the size of the particles for which the centrifugal force and the drag are equal, is calculated by equating these two forces. The analytical approach developed in (Barth, 1956), which was further extended in (Muschelknautz & Brunner, 1967), is now widely used in the literature (Löffler, 1988).

PD essentially depends on the cyclone geometry, surface roughness and operating conditions including gas temperature, inlet velocity, and solid loading. We use the mathematical model described in (Löffler, 1988) to calculate PD.

Computational fluid dynamics models were applied successfully to the prediction of CE and PD as well (Gimbun et al., 2005). However, for its much better computational efficiency, we use the analytical model described in detail in (Löffler, 1988) in our investigation.

6.1.3 Constraints

As mentioned before, cyclone design optimization consists of maximizing CE and minimizing PD. Based on the available literature (Coker, 1993; Z. Li et al., 1988; Salcedo & Cândido, 2007), we consider five geometric constraints for cyclone design. In addition, we include two more constraints related to CE and PD. Altogether, the constraints are defined as follows:

$$B_e \leq \frac{D_a - D_t}{2} \tag{6.1}$$

$$\frac{4H_eB_e}{\pi D_t^2} \geq 0.44 \tag{6.2}$$

$$\frac{4H_eB_e}{\pi D_t^2} \leq 0.735 \tag{6.3}$$

$$H_t \geq 1.25\, H_e \tag{6.4}$$

$$2.23\, D_t \left(\frac{D_a^2}{H_eB_e}\right)^{\frac{1}{3}} \leq (H - H_t) \tag{6.5}$$

$$\mathrm{CE} \geq \mathrm{CE}_{\min} \tag{6.6}$$

$$\mathrm{PD} \leq \mathrm{PD}_{\max} \tag{6.7}$$

Constraint (6.1) ensures a simple tangential inlet. This is the most common design of inlet for high-efficiency cyclones, which allows easy transition of gas at the entry section and covers a large range of throughput. Furthermore, this kind of inlet is cheaper to produce. Constraints (6.2) and (6.3) provide the lower and the upper bound for the ratio between the entry section H_eB_e and the effective cyclone gas discharge area, $\pi(\frac{D_t}{2})^2$. Constraint (6.4)

Table 6.2: The particle size distribution of esqua DOR 15 ("Dust distributions," n.d.) and the percentage of particles below each size.

Size [µm]	Percentage	Size [µm]	Percentage
0.9	5.37	11.5	41.27
1.1	7.46	15.0	51.04
1.3	9.28	21.0	61.53
1.8	12.90	30.0	73.45
2.6	17.11	43.0	85.79
3.7	21.47	61.0	95.29
5.0	21.47	87.0	99.45
7.5	33.52	123.0	100.00

prevents short time circuiting of the gas. Constraint (6.5) reflects the natural length of the cyclone as given in (Salcedo & Cândido, 2007). The authors suggested that the distance the outlet pipe extends into the cyclone body depends on the inlet and outlet dimensions and proposed an empirical relationship to estimate the natural length. This definition has been used in most studies since then. Constraint (6.6) states a lower bound for CE. Constraint (6.7) limits the cyclone PD which is directly related to operational costs.

6.1.4 Test Problems

For the purpose of optimizing CE and PD, the design of a cyclone needs to be chosen for a given task. For this specific task a target performance is imposed. The designer should know the volumetric gas flow rate, v_p, and the gas properties to determine the type of the cyclone needed and the required size of this device. The gas flow rate and the fitting sizes for our experiments are given in (Weiß, 1986). However, CE varies largely not only with the cyclone design and gas properties but also with particle properties. In this context, not only the cut-off diameter but also the particle distribution largely affects the CE. The overall CE of a cyclone is calculated based on the fractional efficiency of the cyclone, i.e., the weighted average of the collection efficiencies over various particle sizes.

For this purpose, we needed a realistic particle distribution for the chosen default cyclone sizes. In the first step, we calculated the theoretical cut-off diameter and, based on this, we decided for a distribution taken from common test dust data ("Dust distributions," n.d.) given in Table 6.2 (the distribution sum per volume is shown in Figure 6.2). For the sake of a fair comparison, we used the same distribution for all experiments.

The multiobjective optimization of cyclone designs was experimentally evaluated on three real-world cyclone configurations of various sizes. The default geometric parameter values of test cyclone configurations are shown in Table 6.3. Given these values, we constructed test problem instances by varying box constraints of the corresponding decision spaces. In more detail, lower (L_i) and upper (U_i) bounds were calculated as

$$L_i = x_i - d \cdot x_i \qquad U_i = x_i + d \cdot x_i \tag{6.8}$$

with x_i for $i \in \{1, \ldots, 6\}$ corresponding to the six considered geometric parameters. To provide a better insight into the cyclone optimization problem, we considered four test problem instances with $d \in \{0.05, 0.1, 0.2, 0.3\}$ for each of the three cyclone configurations. For all 12 problem instances, $\mathrm{CE}_{\min}$ was set to 0.9 and $\mathrm{PD}_{\max}$ to 1500, since these are commonly used values for high-efficiency cyclones.

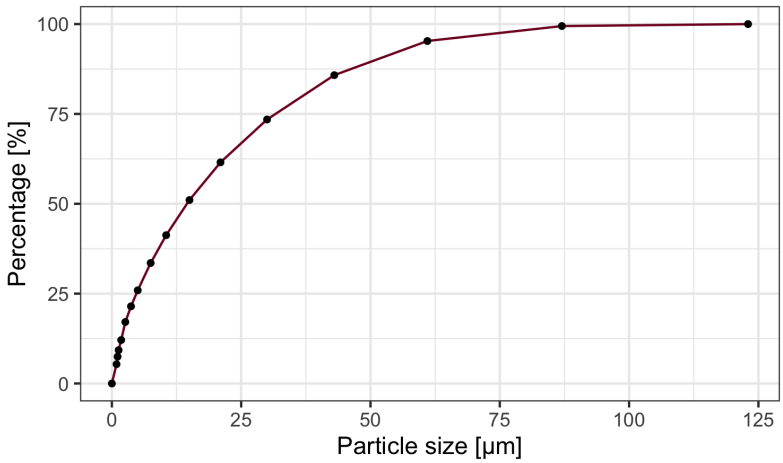


Figure 6.2: The particle size distribution and the percentage of particles below each size as listed in Table 6.2.

Table 6.3: Gas flow rate, v_p, cut-off diameter, d_{50}, and the six geometric parameters with their default values of the addressed test cyclone configurations.

ConFigure	v_p [m^3/s]	d_{50} [µm]	D_a [m]	D_t [m]
S25	0.25	7.4	0.45	0.236
S50	0.50	8.8	0.63	0.335
S100	1.00	10.5	0.90	0.475
ConFigure	H [m]	H_t [m]	H_e [m]	B_e [m]
S25	1.57	0.40	0.224	0.090
S50	2.23	0.58	0.315	0.125
S100	3.14	0.85	0.450	0.180

Table 6.4: ELA features used to characterize test cyclone configurations.

ρ_{F}	Feasibility ratio
ρ_{PD}	Correlation between constraints and PD
ρ_{CE}	Correlation between constraints and CE
$H_{\max}$	Maximum information content
ε_s	Settling sensitivity
M_0	Initial partial information
$(\rho_{\partial F})_{\mathrm{med}}$	Median ratio of feasible boundary crossings

Table 6.5: Resulting feature values for test cyclone configurations.

ConFigure	d	ρ_{F}	ρ_{CE}	ρ_{PD}	$H_{\max}$	ε_s	M_0	$(\rho_{\partial F})_{\mathrm{med}}$
S25	0.05	0.57	0.55	−0.31	0.81	0.95	0.51	0.012
	0.10	0.40	0.49	−0.43	0.77	0.99	0.50	0.011
	0.20	0.24	0.60	−0.54	0.77	0.98	0.51	0.008
	0.30	0.11	0.41	−0.34	0.81	0.99	0.53	0.007
S50	0.05	0.46	0.78	−0.74	0.80	0.93	0.50	0.015
	0.10	0.35	0.72	−0.66	0.79	0.97	0.51	0.010
	0.20	0.23	0.68	−0.62	0.78	0.97	0.52	0.009
	0.30	0.11	0.59	−0.53	0.81	0.99	0.54	0.009
S100	0.05	0.44	0.74	−0.68	0.80	0.91	0.50	0.011
	0.10	0.34	0.74	−0.68	0.79	0.91	0.50	0.010
	0.20	0.22	0.70	−0.64	0.80	0.93	0.50	0.007
	0.30	0.11	0.64	−0.56	0.81	0.95	0.53	0.005

6.2 Feature Space Analysis

6.2.1 Experimental Setup

To analyze the feature space of the cyclone design problem, we refer to seven features listed in Table 6.4. All these features have been previously utilized and described in Section 4.1. In this subsection, we focus on the parameter values used for their computation.

The initial sample size for space-filling design, $|X_{\mathrm{S}}|$, was set to 10 000. Then, the Spearman's rank correlation coefficient was used to express the relationship between the constraints and the objectives, ρ_{CE} and ρ_{PD}. The initial sample size, $|X_{\mathrm{I}}|$, for the information content features was set to 6 000. Finally, the number of steps for the random walks, $|X_{\mathrm{R}}|$, was set to 1 000, each with a maximum step size, δ, of 1% of the range of the domain for each test problem. Thirty independent runs were conducted to obtain the values for $(\rho_{\partial F})_{\mathrm{med}}$.

6.2.2 Results

Table 6.5 shows feature values for the selected cyclone configurations. As we can observe, the feasibility ratios are smaller for larger decision spaces and reduce the feasible solution space to less than a forth or even less of its original size for $d \geq 0.2$. These reduced feasible solution spaces cast doubt on previous results of cyclone design not considering geometrical constraints and stress the importance of constraint handling.

We observe a relatively high correlation between CE and the overall constraint violation, as indicated by the ρ_{CE} values. Consequently, we anticipate that a search for feasible

solutions will direct an algorithm towards solutions with high-quality CE. Moreover, the small value of $(\rho_{\partial F})_{\text{med}}$ suggests that the feasible region is compact and probably connected.

In contrast, the constraints are strongly negatively correlated with PD. We can expect that the algorithm will find it challenging to obtain solutions with high-quality PD. Furthermore, these problems demonstrate a complex local structure, as evident from the high values of information content features.

6.3 Performance Space Analysis

6.3.1 Experimental Setup

Based on the multiobjective formulation of the cyclone design problem, the experimental evaluation aimed at finding sets of trade-off solutions in the form of Pareto front approximations. For this purpose, we used three well-known multiobjective optimization algorithms NSGA-II, DEMO and MOEA/D all equipped with a constraint handling technique. The CDP technique was used in NSGA-II and DEMO, while the penalty function approach in MOEA/D. The algorithms and CHTs are described in Sections 2.3 and 2.4. Additional information can also be found in (Vodopija, Breiderhoff, et al., 2021).

The experimental setup was structured as follows. Initial runs of algorithms were performed, setting the population size at 100 solutions and the number of generations at 100. This configuration was sufficient for algorithms to reach convergence. Additionally, other algorithm parameters were maintained at their default values. Simulated binary crossover and polynomial mutation were used in NSGA-II and MOEA/D. The distribution index was set to 20 for both operators, while the crossover probability was set to 0.9 and the mutation probability to 0.1. Specifically, in MOEA/D, the neighborhood size was set to 20 and the penalty weight to 1.0. DEMO was run using the rand/1/bin strategy, a crossover probability of 0.9 and a scaling factor of 0.7. On each test problem instance, all algorithms were run 31 times, each time with a new randomly initialized population. Finally, PD was normalized to the interval $[0, 1]$ for fairer comparison between the objectives. Note, that CE is within the interval $[0, 1]$ by definition.

6.3.2 Results

The proposed methodology was primarily assessed from the point of view of effectiveness. The quality of an algorithm run was measured with the cumulative hypervolume of the Pareto front approximation found in the run. Here, all nondominated feasible solutions found in the entire run were used for hypervolume calculation. Given CE $\in [0.9, 1]$ and PD $\in [0, 1]$, the reference point for hypervolume calculations was set to $(0.8, 1.1)^{\text{T}}$.

The results in terms of cumulative hypervolume values obtained by each algorithm are shown in Figure 6.3. We can observe statistically significant differences in algorithm performance (determined by the one-way ANOVA, $p < 0.01$) on all test problem instances. The post hoc analysis using the Holm correction to adjust p-values indicates that NSGA-II significantly outperforms both DEMO and MOEA/D on all test problem instances except for S25 ($d = 0.3$), where NSGA-II only outperforms DEMO. Additionally, MOEA/D performs significantly better than DEMO on all instances except for S100 ($d = 0.05, 0.2, 0.3$). Finally, DEMO significantly outperforms MOEA/D on S100 ($d = 0.2, 0.3$), while their performance is comparable on S100 ($d = 0.05$).

Statistical results clearly show the superiority of NSGA-II. In eleven out of twelve test problem instances, NSGA-II outperforms both DEMO and MOEA/D. Moreover, differences among the runs are generally smaller for NSGA-II than those for DEMO and MOEA/D

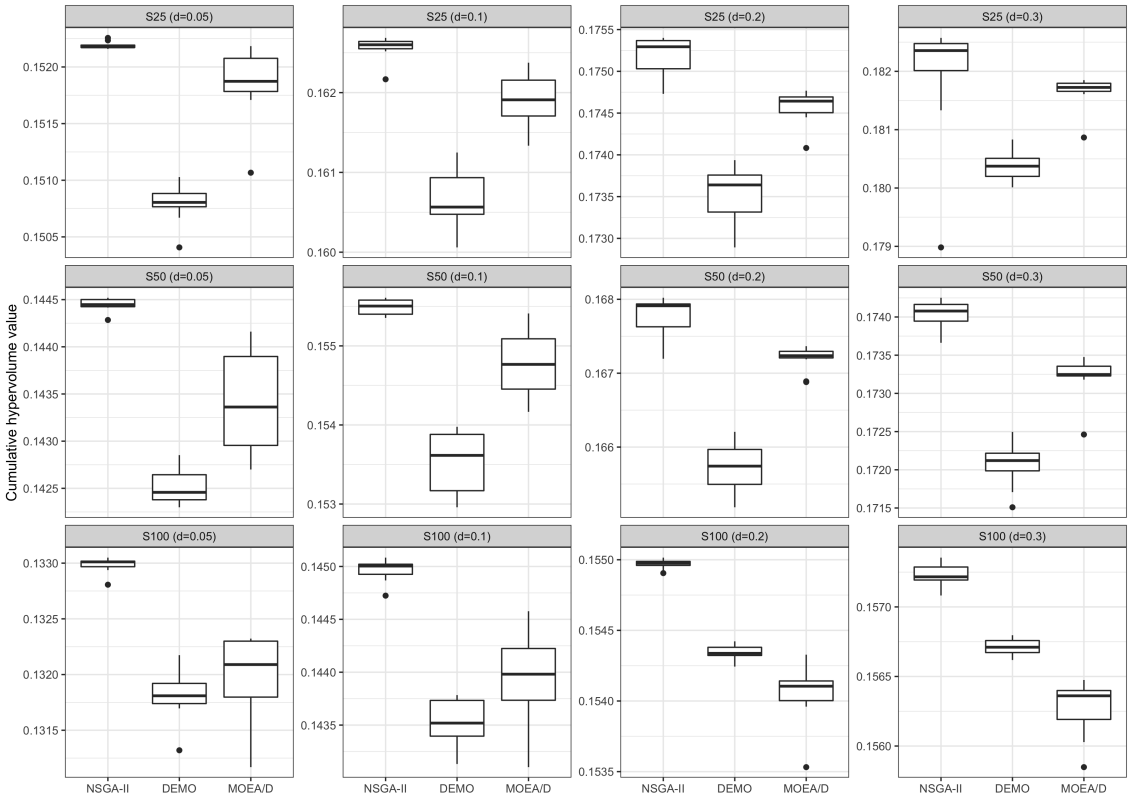


Figure 6.3: Box plots of cumulative hypervolume values obtained for test problem instances by MOEAs.

(Figure 6.3). Besides being the best-performing algorithm, NSGA-II is also the most robust in the sense of producing the most repeatable results.

6.4 Putting the Two Perspectives Together

As detailed in Section 6.2, the test cyclone configurations exhibit positive values for ρ_{CE} and negative values for ρ_{PD}. The negative values for ρ_{PD} suggest that there is a negative correlation between the overall constraint violation and PD. This effect is particularly evident in the Pareto fronts derived from NSGA-II and DEMO, as shown in Figure 6.4. While both algorithms converge with high quality, both NSGA-II and especially DEMO face challenges in achieving good coverage for solutions associated with smaller pressure drops.

Conversely, MOEA/D struggles more in securing coverage for solutions with high CE values. This outcome aligns with expectations, given that MOEA/D is known to have difficulties with concave[1] Pareto fronts (Zhang et al., 2009). However, it is interesting to observe that the negative correlation, ρ_{PD}, does not significantly affect MOEA/D. This implies that MOEA/D might be an appropriate choice for CMOPs that exhibit strong negative correlations between their objectives and constraints.

[1]During the optimization, we minimize both objectives. Specifically, the first objective function is $1 - \text{CE}$. This is why the resulting Pareto front is concave and not convex, as shown in Figure 6.4.

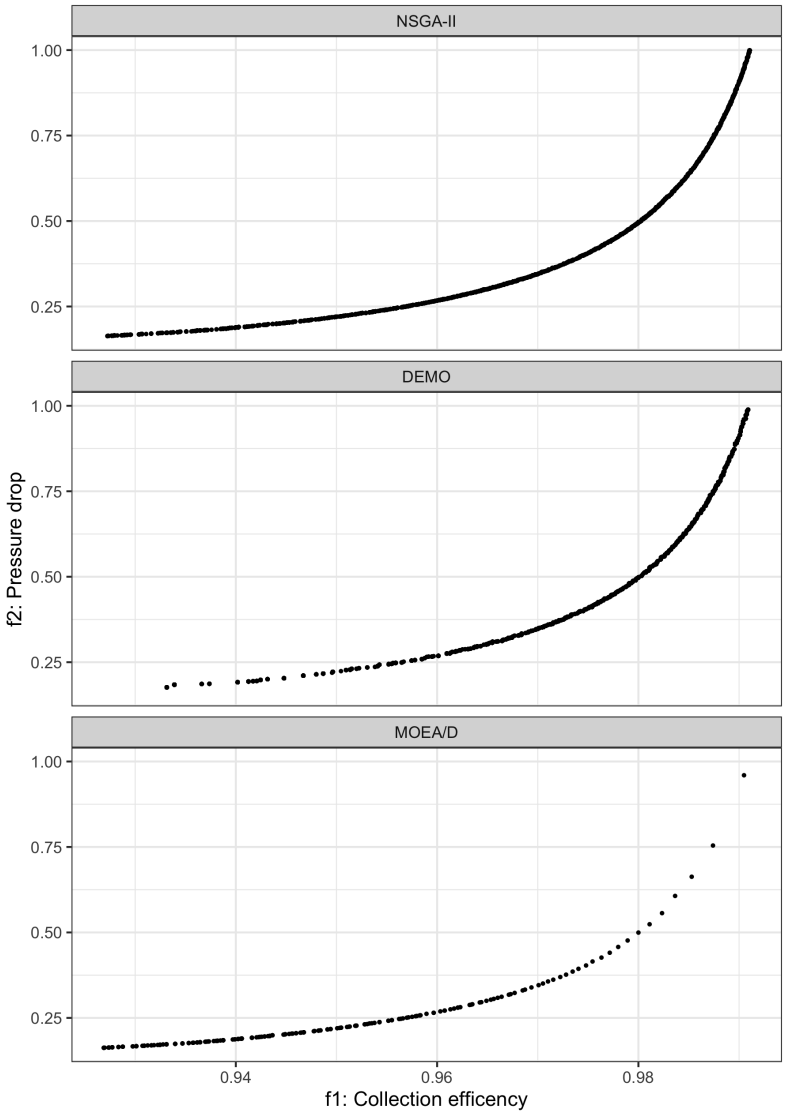


Figure 6.4: Pareto front approximations for cyclone configuration S50 ($d = 0.2$) resulting from typical (median) runs of MOEAs.

Chapter 7

Design of Elevator Group Control

This chapter demonstrates the application of our suggested methodology in the design of elevator group control (EGC). In Section 7.1, we introduce the formulation of the EGC optimization problem (Vodopija, Stork, et al., 2022). Section 7.2 delves into the analysis of the feature space, while Section 7.3 provides an insight into the performance space. Finally, the insights from both perspectives are discussed in Section 7.4.

7.1 Problem Formulation

7.1.1 S-Ring Model

The S-Ring is a discrete, nontrivial event system to simulate EGC (Markon, 2015). It is highly adaptable and thus applicable to different use-cases while maintaining a low implementation effort. Its primary aim is to model the operation of an elevator system by simulating the handling of passenger traffic and passenger serving. It is intended to serve as a dynamic, fast to evaluate, and computationally inexpensive system for optimization purposes, i.e., finding solutions for EGC systems that serve passengers in the fastest, most energy-efficient, and most comfortable way. Due to its low computational cost, the S-Ring can quickly evaluate a broad variety of EGC systems as benchmarks for the proposed multiobjective optimization approach. In general, the S-Ring consists of three key elements: state-space representation, state transition table, and control policy. These elements are briefly explained in the rest of this section to equip the reader with general information on S-Ring models. Additional details can be found in (Markon, 2015; Markon et al., 2006).

7.1.2 State-Space Representation

An S-Ring is represented by its nodes $i \in \{1, \ldots, N_\mathrm{n}\}$. Specifically, each floor contributes two nodes, i.e., one for the upward and one for the downward direction. Exceptions are the first and the top floors that each contributes a single node. Consequently, $N_\mathrm{n} = 2n - 2$, where n is the number of floors in the EGC system. S-Ring nodes indicate the presence of passengers and elevator cars for a particular floor and direction. In more detail, the node state is determined by an ordered pair $s_i(t) = (c_i(t), e_i(t))$, where passenger state $c_i(t) \in \{0, 1\}$ indicates the presence of passengers and elevator state $e_i(t) \in \{0, 1\}$ the presence of an elevator car in node i at time $t \in \mathbb{N}$ (time is assumed to be discrete with unit interval steps). If $c_i(t) = 1$, we say the passenger state of node i is active. Similarly, $e_i(t) = 1$ indicates the elevator state of node i is active. The state of the whole S-Ring at time t is represented by a tuple $s(t) = (s_1(t), \ldots, s_{N_\mathrm{n}}(t))$ containing information on all node states. An example of an EGC system with the related S-Ring model is illustrated in Figure 7.1. During the simulation, the maximum number of nodes with the active elevator

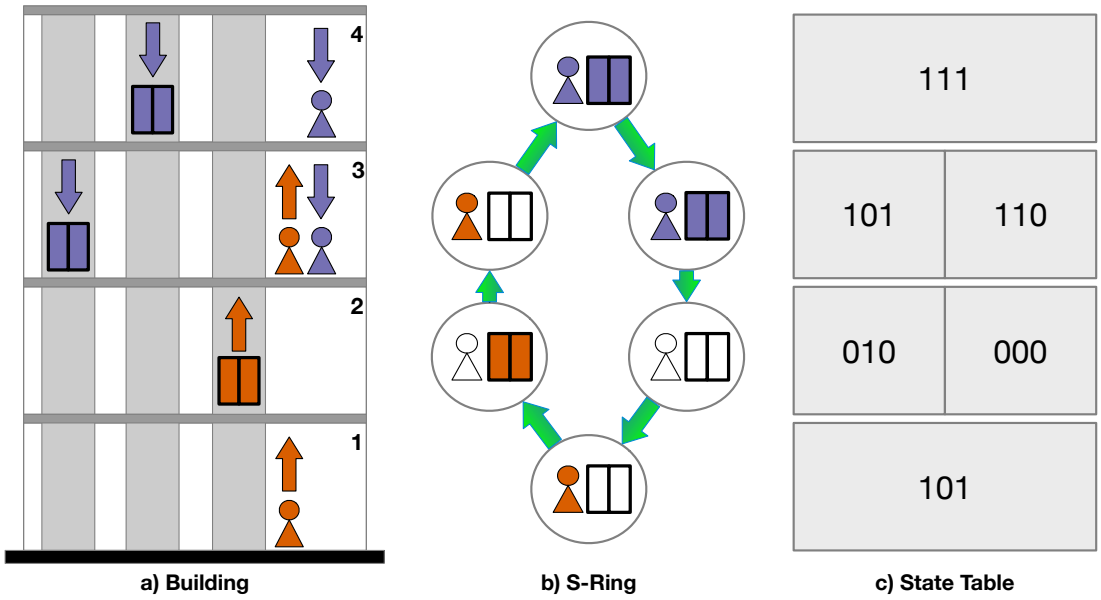


Figure 7.1: S-Ring model with related building and state table: a) A building with three elevator cars, four floors, and two waiting passengers for each direction. Upwards and downwards direction is possible for elevators and passengers and are colored red and blue, respectively. b) The S-Ring model for this building, with a single node for the first and the top floors and two nodes for the rest. c) The state encoding for each of the S-Ring nodes.

state is equal to the number of available elevator cars, m. Moreover, the number of nodes with the active passenger state dynamically changes over time and is influenced by the probability of a newly arriving passenger, p, and it is assumed to be identical for all floors and directions.

7.1.3 State Transition Table

The dynamics of an EGC system is expressed by state transition steps, $s(t) \rightarrow s(t+1)$. The fixed and dynamic rules for state transition steps are shown in the state transition table (Table 7.1), which is applicable for each node in the S-Ring model (Markon et al., 2006). For illustration, the node state $s_i(t) = (c_i(t), e_i(t))$ is extended by the next node state $s_{i+1}(t) = (c_{i+1}(t), e_{i+1}(t))$, forming a triplet $c_i(t), e_i(t), e_{i+1}(t)$, e.g., 101 for a waiting passenger (1xx), no elevator car in the current node (x0x) and a car in the next node (xx1). Moreover, the state transitions are influenced by the probability of arrival, p, or no arrival, $1-p$, of a new passenger. Some states do not require or allow a policy decision and lead to predetermined transitions. For example, if the next node is occupied by another elevator car, the car in the current node cannot move forward until the elevator car in the next node moves forward. In each node, it is first checked whether a new passenger arrived. Next, if the current node has an active elevator state, the controller determines whether the elevator car stops or continues to the next node. Finally, the indication of the passenger state is updated depending on whether or not the passengers were served.

State transition steps happen sequentially in all N_n nodes in a simulation procedure that scans the nodes in a counterclockwise direction. Note that only one node state is changed by performing a single state transition step. The S-Ring simulation is highly influenced by the number of the performed state transition steps, N_t. This number has to be large

Table 7.1: State transition table for the S-Ring. The table shows the node states and decisions to be made in certain node states. The transition probabilities are influenced by the probability of arrival, p, or no arrival, $1-p$, of a new passenger and the control policy. The states are encoded in the following way: $1**$ indicates a passenger present, $*1*$ an elevator in the current node, and $**1$ an elevator in the next node. Policy $\pi = 0$ indicates the elevator car passes the current node and $\pi = 1$ means the elevator car serves the current node. Some states have fixed transitions and do not require a policy decision.

Node state $s_i(t)$	Transition probability	Policy π	Node state $s_i(t+1)$
000	$1-p$		000
	p		100
100	1		100
010	$1-p$		001
	p	0	101
		1	010
110	1	0	101
		1	010
001	$1-p$		001
	p		101
101	1		101
011	1		011
111	1		011

enough for the simulation to converge.

7.1.4 Control Policy

The control policy π establishes the decision of either serving or passing a passenger for the required state transition step and is crucial for any optimization. In general, an optimized policy could be realized by a lookup table if n is small. However, for large n this becomes infeasible, and a perceptron neural network is usually utilized in the literature to perform the state-to-decision mapping (Markon et al., 2006). The perceptron has a simple, direct structure without hidden layers. It includes $2N_\mathrm{n}$ binary inputs where the first N_n inputs denote the passenger states and the remaining N_n inputs the elevator states. The inputs are weighted (therefore the length of the weight vector equals $2N_\mathrm{n}$) and directly connected to a binary output indicating the control policy decision. For a given setup of n, m, and p, the perceptron's size remains fixed, and the-state-to-decision mapping is only influenced by the perceptron weight vector $x \in [-1, 1]^{2N_\mathrm{n}}$.

7.1.5 Optimization Problem

In this subsection, we propose a multiobjective formulation of the EGC optimization problem. Specifically, we deal with two EGC objectives that are often studied in the literature, and both need to be minimized: the average number of S-Ring nodes with waiting passengers, and the total number of elevator stops (Hakonen et al., 2004; Sahin et al., 2013; Tyni & Ylinen, 2006). In contrast to previous studies, we do not combine the objectives into a single function but adopt the multiobjective perspective. Moreover, to make it possible to compare the performance of elevator systems of various configurations (determined by the number of floors, n, and the number of elevator cars, m), we consider

normalized objective function values.

The first objective, h_1, is the proportion of nodes with waiting passengers. It is expressed as the average number of nodes with waiting passengers during the S-Ring simulation, M_w, divided by the number of all nodes, N_n:

$$h_1 = \frac{M_\mathrm{w}}{N_\mathrm{n}}. \tag{7.1}$$

The second objective, h_2, is the proportion of elevator stops. It is equal to the total number of elevator stops observed during the entire S-Ring simulation, M_t, divided by the maximum possible number of elevator stops. The latter can be calculated as the number of elevator cars, m, multiplied by the number of EGC simulation cycles, which in turn corresponds to the number of state transition steps, N_t, divided by the number of nodes, N_n, therefore

$$h_2 = \frac{M_\mathrm{t}}{mN_\mathrm{t}/N_\mathrm{n}}. \tag{7.2}$$

Intuitively, the passengers' discomfort with long waiting times and long riding times due to many elevator stops does not increase linearly with time, but more drastically. To model this effect, we have additionally modified the original objectives as

$$f_1 = h_1^\alpha \quad \text{and} \quad f_2 = h_2^\beta, \tag{7.3}$$

where $\alpha, \beta \in [1, 2]$ are the objective function coefficients. The choice of their values is subjective, but the idea is to reflect the elevator system characteristics and the passenger preferences.

In (Vodopija et al., 2018), we observed that many obtained solutions allow for a large number of elevator car skips. It sometimes happened that a passenger was skipped more than 20 times, which made an EGC policy impractical. In this work we therefore introduce a constraint that renders all solutions with a large number of elevator skips infeasible. The corresponding constraint is expressed as the maximum number of elevator skips, M_s, that has to be less than or equal to a predefined number, M, therefore:

$$g = M_\mathrm{s} \leq M. \tag{7.4}$$

The resulting constrained multiobjective optimization problem (CMOP) can be mathematically formulated as

$$\begin{aligned} \text{minimize} \quad & f_m(x), \quad m = 1, 2 \\ \text{subject to} \quad & g(x) \leq M \end{aligned} \tag{7.5}$$

where $x = (x_1, \ldots, x_D)^\mathrm{T}$ is a perceptron weight, $f_1, f_2 : [-1, 1]^D \to \mathbb{R}$ are the two objective functions, $g : [-1, 1]^D \to \mathbb{R}$ is the constraint function, and $[-1, 1]^D$ the decision space of dimension $D = 2N_\mathrm{n}$. Additionally, $f_m(x)$ is an objective value and $v(x) = \max(g(x) - M, 0)$ constraint violation.

7.1.6 Test Elevator System Configurations

Table 7.2 summarizes the characteristics of the test elevator system configurations used to assess the performance of multiobjective optimization approaches in our experiments. Nine configurations were carefully selected to cover a wide range of elevator systems operating in various kinds of buildings. The number of floors determines the building size. It varies among 5, 20, and 50. The probability of newly arriving passengers takes values among 0.01, 0.05, and 0.1. The smallest probability reflects the elevator systems operating in

Table 7.2: Characteristics of the test elevator system configurations: number of floors n, number of elevator cars m, probability of a newly arriving passenger p, number of nodes in the S-Ring representation N_{n}.

Config.	n	m	p	N_{n}
C1	5	1	0.01	8
C2	5	2	0.05	8
C3	5	3	0.10	8
C4	20	4	0.01	38
C5	20	6	0.05	38
C6	20	8	0.10	38
C7	50	10	0.01	98
C8	50	15	0.05	98
C9	50	20	0.10	98

Table 7.3: ELA features used to characterize test EGC configurations.

ρ_{F}	Feasibility ratio
ρ_{f_1}	Correlation between g and f_1
ρ_{f_2}	Correlation between g and f_2
$H_{\max}$	Maximum information content
ε_s	Settling sensitivity
M_0	Initial partial information
$(\rho_{\partial F})_{\mathrm{med}}$	Median ratio of feasible boundary crossings

environments with low passenger traffic, such as residential buildings. On the other hand, the probability of 0.1 is related to high passenger traffic observed in commercial buildings or parking garages. The number of elevator cars was selected based on building size and passenger traffic. The idea was to reflect real and meaningful elevator systems. The parameters α, β were set to 1.5 and M to 4 for all test configurations.

7.2 Feature Space Analysis

7.2.1 Experimental Setup

To analyze the feature space of the EGC optimization problem, we refer to seven features listed in Table 7.3. All these features have been previously utilized and described in Section 4.1. In this subsection, we focus on the parameter values used for their computation.

The initial sample size for space-filling design, $|X_{\mathrm{S}}|$, was set to 10 000. Then, the Spearman's rank correlation coefficient was used to express the relationship between the constraint and the objectives, ρ_{f_1} and ρ_{f_2}. The initial sample size, X_{I}, for the information content features was set to 6 000. Finally, the number of steps for the random walks, X_{R}, was set to 1 000, each with maximum step size, δ, of 1% of the range of the domain for each test problem. Thirty independent runs were conducted to obtain the values for $(\rho_{\partial F})_{\mathrm{med}}$.

7.2.2 Results

Table 7.4 shows the resulting feature values for the test EGC configurations. A look at the feasibility ratios across configurations reveals that they are rather small, ranging between 0.06 and 0.32. This indicates that, although the feasibility ratio could possibly affect the

Table 7.4: Resulting feature values for test EGC configurations.

Config.	ρ_{F}	ρ_{f_1}	ρ_{f_2}	$H_{\max}$	ε_s	M_0	$(\rho_{\partial F})_{\mathrm{med}}$
C1	0.32	0.97	−0.95	0.73	3.03	0.48	0.18
C2	0.21	0.97	−0.93	0.75	3.35	0.50	0.21
C3	0.23	0.96	−0.94	0.74	3.53	0.49	0.20
C4	0.06	0.93	−0.97	0.81	2.32	0.59	0.17
C5	0.11	0.52	−0.97	0.81	2.50	0.59	0.25
C6	0.18	0.66	−0.98	0.82	2.65	0.57	0.35
C7	0.07	0.73	−0.98	0.80	2.00	0.62	0.45
C8	0.09	0.50	−0.96	0.82	2.22	0.60	0.57
C9	0.08	0.62	−0.95	0.81	2.34	0.61	0.69

performance of algorithms, its impact is not substantial enough to severely impact their efficiency.

The correlation between the constraint and the first objective function, ρ_{f_1}, is positive across configurations, with values ranging from 0.5 to 0.97. This positive correlation suggests that algorithms are expected to find good solutions for this objective.

On the other hand, the correlation between the constraint and the second objective function, ρ_{f_2}, is strongly negative, with values close to -1. This implies that algorithms may encounter difficulty finding good solutions for this objective.

Turning attention to the local structure of the problems, one can observe high values for information content features across all configurations. This observation indicates that the structure of the feasible region is complex, which could make it challenging for algorithms to find high-quality solutions. Moreover, the values for $(\rho_{\partial F})_{\mathrm{med}}$ are also relatively high, suggesting that the feasible region is heavily disjointed. This adds another layer of complexity for the algorithms.

In conclusion, the results indicate that while algorithms are expected to perform well in finding solutions for the first objective, f_1, they may face considerable hurdles due to the complex local structure and the strong negative correlation between the constraint and the second objective, f_2.

7.3 Performance Space Analysis

7.3.1 Experimental Setup

Based on the constrained multiobjective formulation of the EGC optimization problem, the experimental evaluation aimed at finding sets of feasible trade-off solutions representing Pareto front approximations. For this purpose, we used three MOEAs equipped with their default CHTs as described in Sections 2.3 and 2.4: NSGA-II, DEMO, and MOEA/D. The CHT used in NSGA-II and DEMO was the CDP, while penalty function was considered in MOEA/D.

The applied MOEAs involve various parameters that effect their operation. The choice of parameter values can profoundly impact algorithm performance. Thus, in many real-world applications, it is mandatory to select an adequate set of algorithm parameter values to solve the given problem efficiently. Besides, the adequate choice of parameter values allows for a more sound and robust comparison of various algorithms. This is because an algorithm's chance to be inferior due to an inappropriate choice of parameter values is highly reduced.

Table 7.5: Tuned algorithm parameter values for MOEAs aggregated over test elevator system configurations of the same size: total number of solution evaluations n_f, population size n_p, number of generations n_g, crossover probability p_c, mutation probability p_m, and scaling factor s_f.

Config.	NSGA-II				DEMO				MOEA/D			
	n_p	n_g	p_c	p_m	n_p	n_g	p_c	s_f	n_p	n_g	p_c	p_m
C1–C3	120	84	0.78	0.46	328	31	0.29	0.22	204	50	0.77	0.50
C4–C6	240	167	0.79	0.16	264	152	0.55	0.34	120	334	0.59	0.32
C7–C9	180	556	0.91	0.17	432	232	0.58	0.42	408	246	0.70	0.11

In the literature, various approaches to tuning or controlling algorithm parameters have been proposed. However, one of the most reliable and efficient methods used for real-world problems remains sequential model-based parameter optimization. This approach falls in the group of parameter tuning methods and is one of the most frequently used approaches when dealing with computationally demanding evaluations. In our work, Sequential Parameter Optimization (SPO) (Bartz-Beielstein et al., 2005) was used to tune the algorithm parameters.

The target function optimized by SPO was the MOEA performance measured as the obtained cumulative hypervolume given the algorithm parameter settings. The tuned parameters were: the population size, $n_\mathrm{p} \in \{48, \ldots, 500\}$, the crossover probability, $p_\mathrm{c} \in [0, 1]$, and the mutation probability, $p_\mathrm{m} \in [0, 1]$. Specifically, the scaling factor, $s_\mathrm{f} \in [0, 2]$, was tuned instead of the mutation probability in DEMO.

For each algorithm, the number of generations, n_g, was determined as the number of solution evaluations, n_f, divided by the population size:

$$n_\mathrm{g} = \left\lceil \frac{n_\mathrm{f}}{n_\mathrm{p}} \right\rceil . \tag{7.6}$$

Therefore, n_g was not considered as a decision variable (algorithm parameter) by the tuning process. The number of allowed solution evaluations changed proportionately with the number of floors: $n_\mathrm{f} = 2000n$. Note that $n_\mathrm{g} n_\mathrm{p}$ may be greater than n_f. If this happened, the run was stopped after n_f solution evaluations.

Additionally, we used the following SPO configuration: Gaussian process regression for building surrogate models, classic differential evolution as the optimizer, Latin hypercube sampling to initiate the MOEA parameter values, and 25 target function evaluations (MOEA runs) without repetitions. The tuned algorithm parameter values aggregated over configurations of the same size are summarized in Table 7.5.

Finally, the candidate solutions in the algorithm runs were evaluated by the S-Ring simulation with a predefined number of simulation cycles. This was 100 000 for all test elevator system configurations. As a consequence, the number of state transition steps was equal to $N_\mathrm{t} = 100\,000 N_\mathrm{n}$, which was sufficient for the convergence of all S-Ring models used in this study.

7.3.2 Results

The algorithms were assessed from the point of view of effectiveness—quality of results. To assess the algorithm effectiveness, every algorithm was run 31 times, each time with a new randomly initialized population of solutions and the tuned parameter values from Table 7.5. The cumulative hypervolume of the Pareto front approximation and the cumulative inverted generational distance plus (IGD^+) were used to measure the quality of results. Cumulative

Table 7.6: Average cumulative hypervolume values for all tested algorithms on the test elevator system configurations.

Config.	NSGA-II	DEMO	MOEA/D
C1	1.19 ± 0.00	1.19 ± 0.00	1.19 ± 0.00
C2	1.10 ± 0.00	1.10 ± 0.00	1.10 ± 0.00
C3	1.05 ± 0.00	1.05 ± 0.00	1.05 ± 0.00
C4	1.16 ± 0.00	1.16 ± 0.00	1.15 ± 0.01
C5	0.95 ± 0.00	0.92 ± 0.01	0.92 ± 0.02
C6	0.79 ± 0.00	0.76 ± 0.01	0.77 ± 0.01
C7	1.03 ± 0.01	1.02 ± 0.02	0.98 ± 0.04
C8	0.69 ± 0.02	0.61 ± 0.02	0.63 ± 0.02
C9	0.63 ± 0.01	0.59 ± 0.02	0.60 ± 0.02

Table 7.7: Average cumulative IGD^+ values for all tested algorithms on the test elevator system configurations.

Config.	NSGA-II	DEMO	MOEA/D
C1	0.0007 ± 0.0002	0.0007 ± 0.0003	0.0006 ± 0.0002
C2	0.0022 ± 0.0006	0.0023 ± 0.0006	0.0021 ± 0.0005
C3	0.0025 ± 0.0005	0.0031 ± 0.0008	0.0036 ± 0.0018
C4	0.0059 ± 0.0026	0.0074 ± 0.0004	0.0088 ± 0.0040
C5	0.0090 ± 0.0025	0.0183 ± 0.0049	0.0195 ± 0.0064
C6	0.0094 ± 0.0025	0.0254 ± 0.0038	0.0180 ± 0.0059
C7	0.0118 ± 0.0047	0.0211 ± 0.0163	0.0476 ± 0.0034
C8	0.0221 ± 0.0110	0.0569 ± 0.0130	0.0635 ± 0.0085
C9	0.0118 ± 0.0041	0.0271 ± 0.0072	0.0287 ± 0.0108

means that all the nondominated feasible solutions found in the entire run were used for hypervolume and IGD^+ calculation. Given $f_1, f_2 \in [0, 1]$, the reference point for hypervolume calculation was set to $(1.1, 1.1)^{\mathrm{T}}$. Additionally, all the nondominated feasible solutions found over all runs and algorithms were used as a reference front for calculating IGD^+. The means of cumulative hypervolume and IGD^+ values averaged over 31 runs are shown in Tables 7.6 and 7.7, respectively.

As we can see, for all three MOEAs the obtained hypervolume and IGD^+ values are very similar on C1–C4. In contrast, NSGA-II outperforms DEMO and MOEA/D on C5–C9 concerning IGD^+. Moreover, NSGA-II performs better than DEMO and MOEA/D on C5, C6, C8 and C9 with respect to hypervolume, while on C7 only MOEA/D is outperformed by NSGA-II. Finally, only negligible differences are observed between DEMO and MOEA/D performance on C5, C6, C8 and C9 with respect to hypervolume and IGD^+.

The statistical analysis confirms these findings. According to the Friedman test, we observe statistically significant differences in MOEA performance concerning both hypervolume ($p < 0.01$) and IGD^+ ($p < 0.01$). Post hoc analysis with the Wilcoxon signed-rank test and the Benjamini–Hochberg procedure to adjust p-values shows that NSGA-II performance is superior to those of DEMO and MOEA/D, while no statistically significant differences are observed between DEMO and MOEA/D performance. The adjusted p-values of pairwise comparisons are shown in Table 7.8.

The results can also be analyzed by visualizing the obtained Pareto front approximations. Figure 7.2 shows Pareto front approximations for the test configurations resulting from

Table 7.8: Adjusted p-values resulting from post hoc analysis with Wilcoxon signed-rank test and Benjamini–Hochberg procedure. Asterisks (*) indicate statistically significant differences (at the significance level of 0.05) in algorithm performances.

Pair	Hypervolume	IGD^+
NSGA-II vs. DEMO	0.0056*	0.0078*
NSGA-II vs. MOEA/D	0.0056*	0.0279*
DEMO vs. MOEA/D	0.9102	0.2031

typical runs of the MOEAs. In more detail, all the runs corresponding to a given test elevator configuration are sorted based on the obtained cumulative hypervolume, and the front obtained in the median run is shown in the figure. We can see that the Pareto front approximations produced by NSGA-II are better in convergence for C5, C6, and C8 than those obtained by the other MOEAs. In contrast to the statistical results, it is hard to see any difference in MOEA performance by only viewing the fronts for test configurations C7 and C9.

Figure 7.3 shows box plots for the cumulative hypervolume obtained before and after parameter tuning on test elevator configurations C7 and C9. The box plots on the left show the results for the ten runs used in the initial phase of SPO. These parameter values were sampled using the LHS design of experiment method and denote the results before tuning. The box plots on the right side show the hypervolume achieved in 31 runs with the tuned parameter values from Table 7.5.

The results show that NSGA-II is less sensitive to parameter tuning than DEMO and MOEA/D. While the performance of the latter two algorithms varies considerably, NSGA-II is robust over various parameter settings. In particular, the results obtained before tuning NSGA-II are already concentrated around the median hypervolume values. The improvements in the tuning process for NSGA-II are negligible compared to the improvements obtained for DEMO and MOEA/D. For example, on C7, the hypervolume values obtained before tuning NSGA-II are almost as good as those obtained after tuning it. Moreover, the results obtained before tuning NSGA-II are already better than the results obtained after tuning MOEA/D. We can also see that DEMO sometimes performs better than NSGA-II but at the expense of performing worse in most runs.

The box plots for C9 show that NSGA-II performs better than DEMO and MOEA/D even with random parameter settings. Additionally, it is worth noting that NSGA-II using tuned parameters produces three outlier runs: one run achieving much better results and two runs achieving much worse results than the median run. Test elevator configuration C9 is the hardest to optimize, and it seems that even the best performing algorithm can sometimes underperform on it.

In Figure 7.4, the ERDs aggregated over 31 runs are shown for C7 and C9 and all the three MOEAs. Although NSGA-II and DEMO achieve comparable hypervolumes on C7, NSGA-II is more efficient. Similar behavior is observed on other test elevator configurations. This further proves that NSGA-II is the best performing algorithm on the test elevator system configurations used in this study. Interestingly, the algorithms can rapidly locate the feasible region and the ideal zone. However, they subsequently run into difficulties in locating high-quality solutions, as indicated by the nearly straight lines in the middle of the graphs.

To summarize, NSGA-II performs at least as well as DEMO and MOEA/D on C1–C4 and C7, and outperforms both algorithms on C5, C8 and C9. Its efficiency and robustness are also superior to other MOEAs (Figures 6.3 and 7.4).

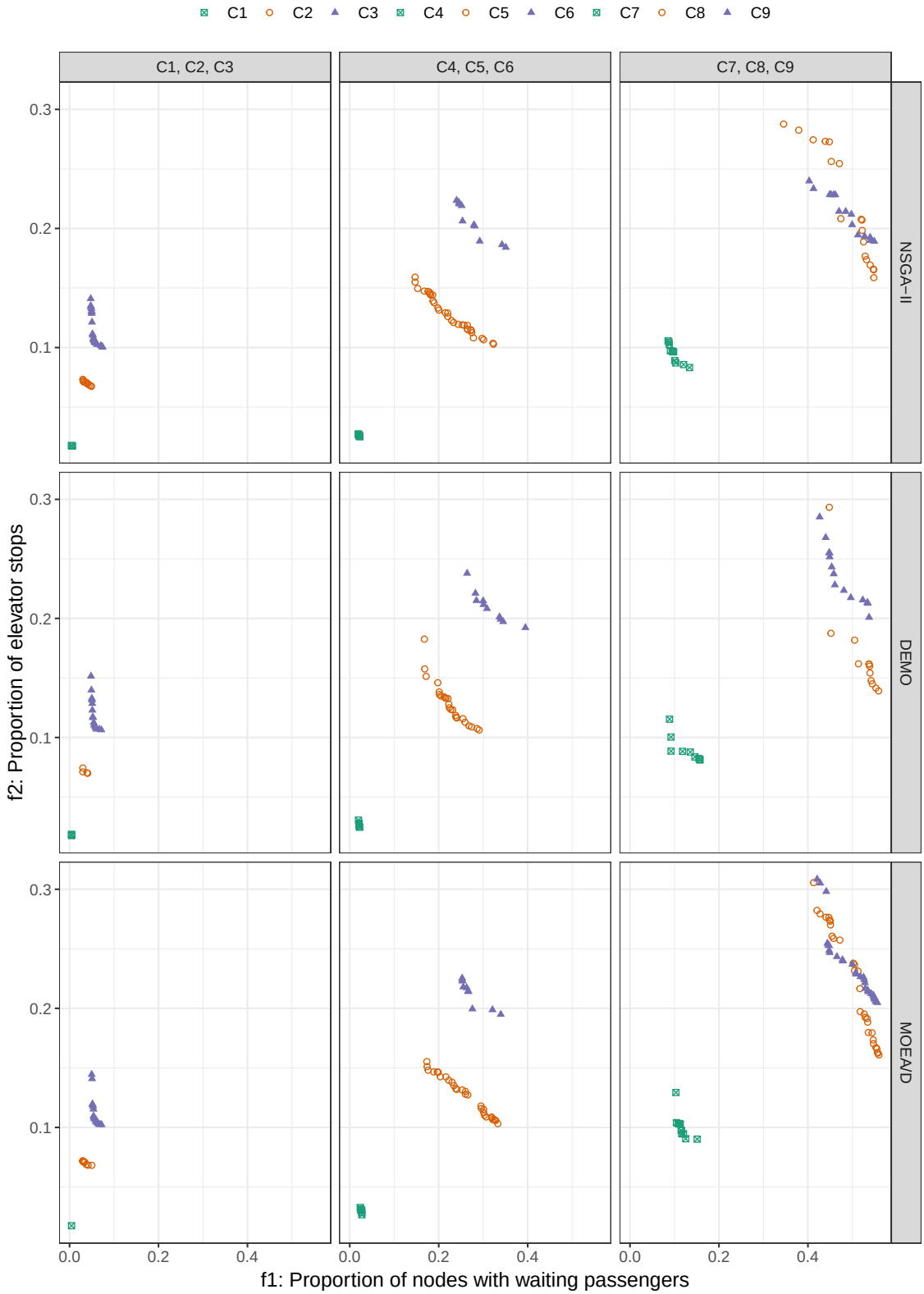


Figure 7.2: Pareto front approximations for the test elevator system configurations.

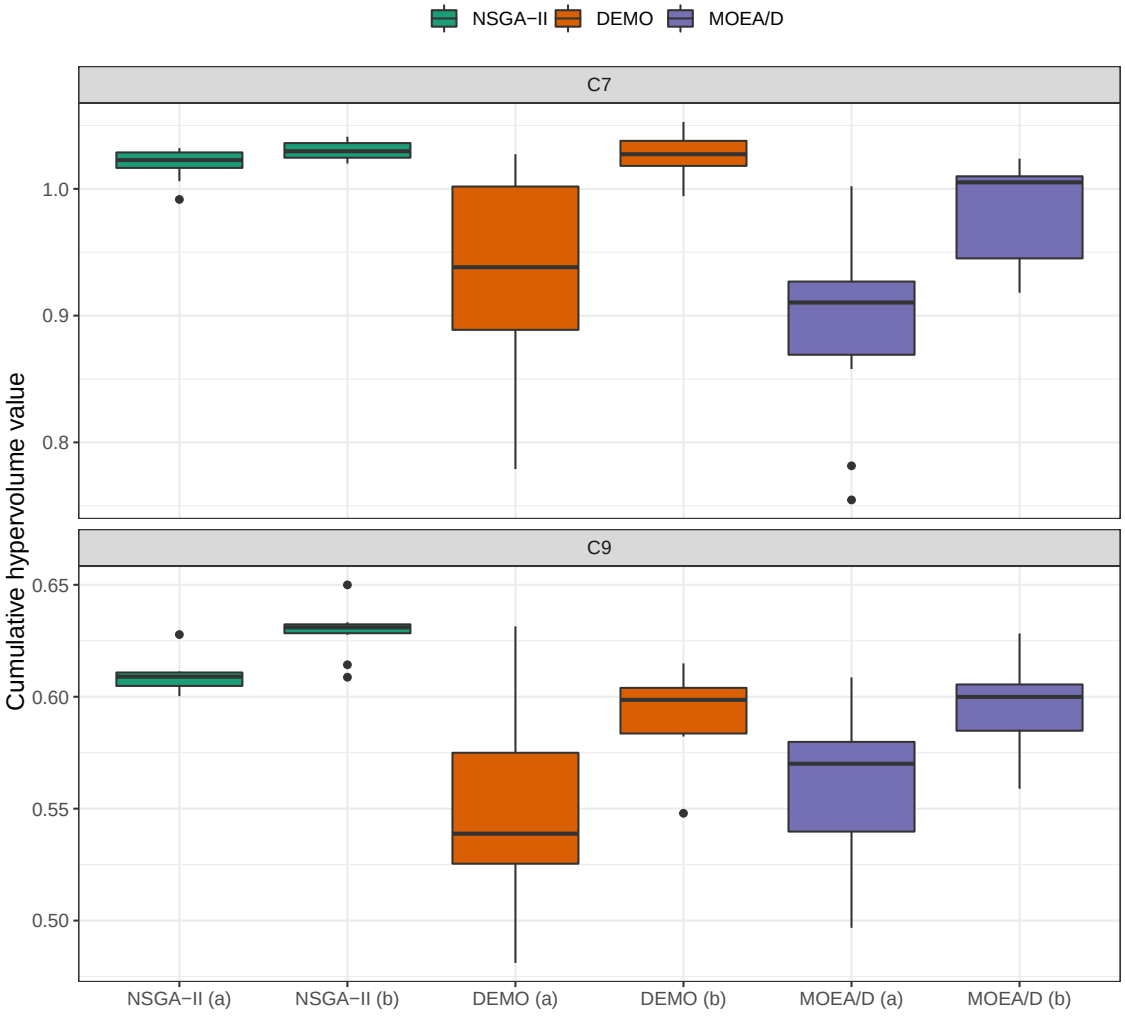
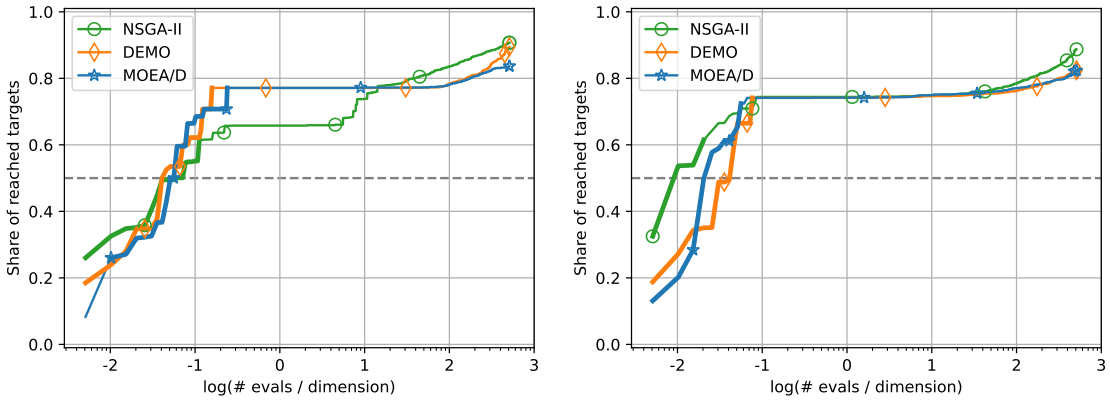


Figure 7.3: Box plots of cumulative hypervolume values obtained by MOEAs. The cumulative hypervolume values obtained (a) before and (b) after tuning are shown for each algorithm.



(a) C7 (b) C9

Figure 7.4: ERDs for the test elevator system configurations C7 and C9 and all the three MOEAs.

Finally, we investigated how NSGA-II performs concerning both objectives. In other words, we analyzed the spread of the obtained Pareto front approximations. We created two constrained single-objective optimization problems, where the two objectives, f_1 and f_2, were subject to minimization, separately. Ordered pairs of solutions for both objectives represent approximations of ideal points. To obtain these approximations, the best feasible solutions found in 31 runs of classic differential evolution (DE) (Price et al., 2005) were recorded.

The results are shown in Figure 7.5, where both the optimized solutions found by DE and the Pareto front approximations obtained in the median runs of NSGA-II are depicted. From the figure it is evident that it is harder for NSGA-II to find optimal solutions with respect to the second objective, e.g., for C7 and C8. This was expected since this objective is strongly negatively correlated with the constraint (see Figure 7.6). On the other hand, we can see that NSGA-II can always find near-optimal solutions concerning the first objective. In conclusion, a typical run of NSGA-II can find a satisfactorily spread Pareto front approximation for each test elevator configuration.

7.4 Putting the Two Perspectives Together

In this section, we offer an analysis of the EGC optimization problem by concurrently examining both the feature and performance spaces. In particular, we focus on the characterization of the EGC optimization problem with respect to the difficulty of handling individual elevator system configurations and correlations between the objectives and the constraint. As previously observed, the algorithms exhibit similar performances on configurations C1–C4, suggesting these elevator configurations are relatively simple to solve. For instance, configurations C1 and C2 depict elevator systems in smaller buildings with minimal passenger flow. Simple EGC strategies, such as always serving a passenger, are already effective. On the other hand, the results indicate that the test configurations C5–C9 are harder to solve than C1–C4.

In Section 7.2, we see that the correlation between the constraint and the first objective function is positive for all configurations. Specifically, in the case of C2 the first objective is strongly positively correlated ($\rho_{f_1} = 0.97$) with the constraint. The EGC policy that always serves a passenger is a reasonable choice for an elevator system such as C2. This

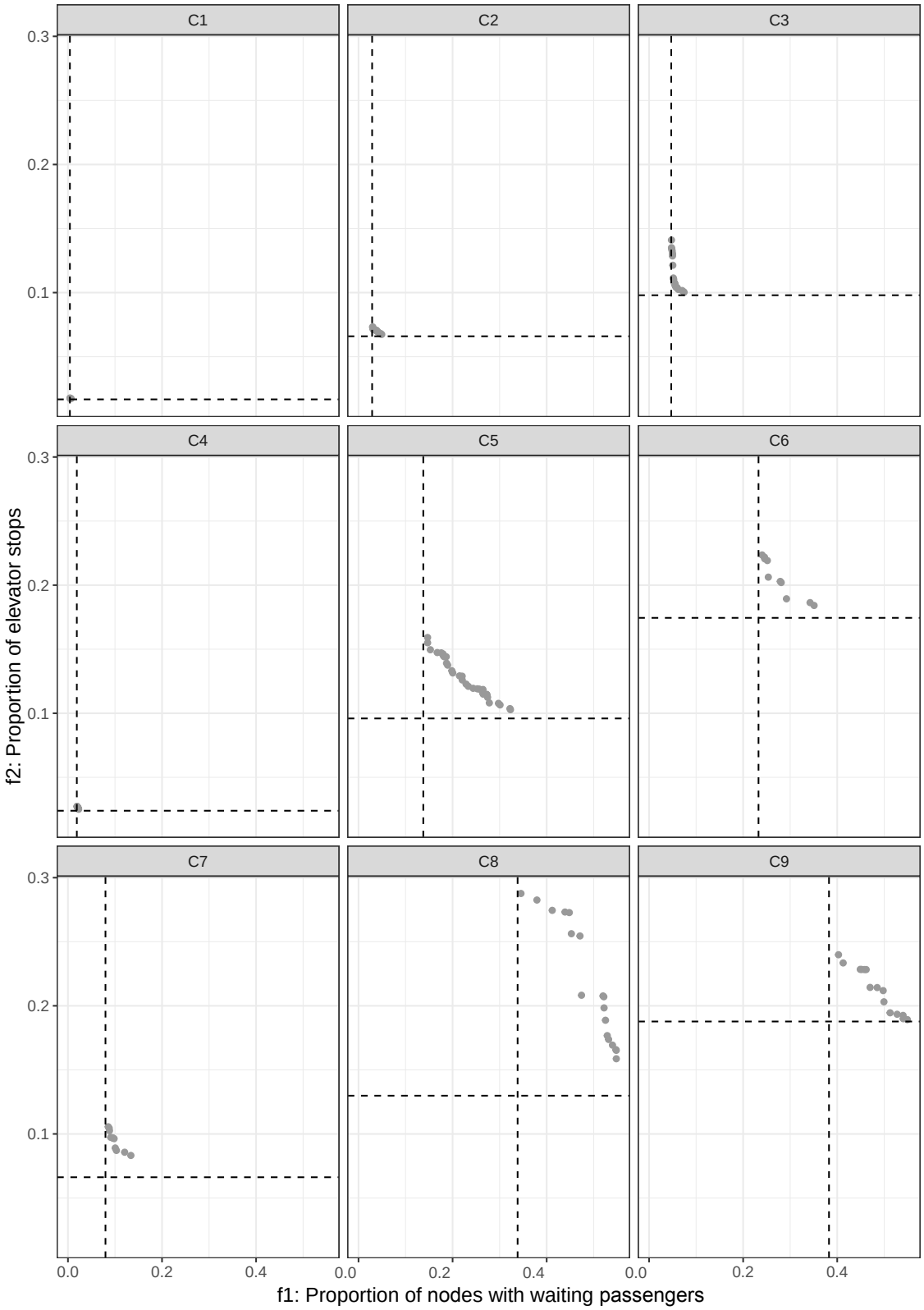


Figure 7.5: Pareto front approximations resulting from typical runs of NSGA-II. The dashed lines show the minimum objective values found by DE, where each objective was optimized separately.

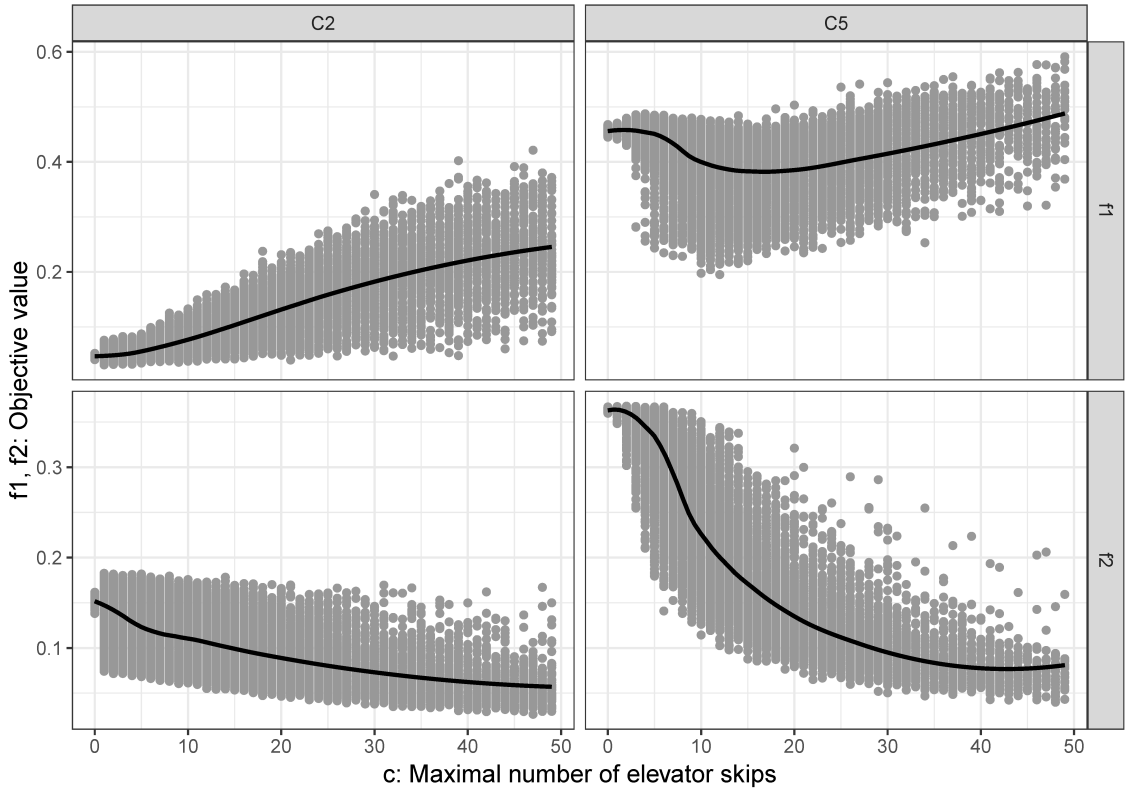


Figure 7.6: A detailed analysis of the correlations between the objective and constraint values, conducted on a sample of 100 000 randomly generated solutions.

renders the test configuration C2 easy to solve since optimizing the first objective already improves the constraint violation.

In C5, we still observe a positive correlation ($\rho_{f_1} = 0.52$) between the objective and the constraint for $10 \leq g \leq 50$, but while this positive correlation persists until $g \leq 4$ in C2, this is not true for C5 (see Figure 7.6). This makes this constraint harder to satisfy than the constraint involved in C2.

As already seen in Section 7.2, the second objective is in both cases strongly negatively correlated with the constraint ($\rho_{f_2} = -0.93, -0.97$). This explains why the examined MOEAs have difficulty finding good solutions for this objective. The results for other test elevator configurations are not shown here since C1, C3, and C4 behave similarly to C2, and C6–C9 behave almost identically to C5.

Chapter 8

Conclusions

This final chapter presents the concluding remarks. A summary of the thesis is given in Section 8.1, while Section 8.2 recalls its original contributions. The chapter ends with ideas for future work in Section 8.3.

8.1 Summary

8.1.1 Feature Space

In this thesis, we have investigated CMOPs from the feature space perspective. Firstly, we have extended the concepts of fitness and violation landscapes to constrained multiobjective optimization and proposed a mathematical formulation to describe multimodality in the violation landscape. Secondly, four ELA techniques, namely space-filling design, information content, random walks, and adaptive walks, have been adapted for CMOPs. Next, we have used the resulting features to analyze eight artificial test suites and compare them against the RCM suite, which consists of real-world problems based on physical models. In particular, the artificial test suites have been assessed with respect to the representativeness of various RCM characteristics. Additionally, by contrasting the test suites against a set of all the considered CMOPs, we have also assessed whether the test suites contain problems with various characteristics. The effectiveness of the proposed features in predicting algorithm performance has also been assessed on two MOEAs. Finally, we have discussed the advantages and limitations of the existing artificial test suites, as well as the sensitivity and limitations of our methodology. The experimental results have shown that the proposed ELA techniques are adequate to characterize constrained multiobjective optimization problem landscapes with up to five decision variables.

Our findings indicate that the artificial test problems fail to satisfactorily represent some real-world problem characteristics. In particular, the sizes of feasible components, correlations between the objectives and the overall constraint violation, feasibility ratios, and ruggedness of the violation landscapes are not well represented by the existing artificial test suites. Moreover, the predominant source of complexity in novel artificial test problems is very often violation multimodality. This can lead to biased interpretations and conclusions since many other relevant characteristics are ignored. All this clearly suggests that further advances in landscape analysis for CMOPs are required to overcome these drawbacks.

Additionally, the experimental analysis on feature-based MOEA performance prediction has shown the great potential of the proposed ELA features. The most valuable features are those derived from adaptive walks and space-filling design. However, the results indicate that the synergy gained from all the feature groups yields especially prospective results.

This work is an important step towards enhancing the understanding of test suites

developed for constrained multiobjective optimization. It has revealed that all the studied artificial test suites have certain advantages and limitations and showed that no "perfect" test suite exists.

Based on the findings from this study, we are able to affirm the validity of Hypothesis 1 and Hypothesis 3.

8.1.2 Performance Space

This thesis has also presented an investigation of the existing artificial test CMOPs from the performance space perspective. Firstly, we have proposed a performance assessment methodology capable of simultaneously monitoring both the front approximation and constraint satisfaction. This methodology is an extension of the approach used in the COCO platform for unconstrained bi-objective optimization problems. Next, the resulting performance methodology has been used to analyze and contrast eight artificial test suites. Specifically, the test suites have been assessed with respect to the effectiveness of differentiating between three well-known MOEAs. Finally, the thesis has discussed the advantages and drawbacks of the existing artificial test suites and discloses some limitations of the proposed methodology.

The experimental results show that the CF, DC-DTLZ, and especially MW suites have the greatest potential in differentiating the three MOEAs. They all include multiple CMOPs that can separate the MOEAs in both front approximation and constraint satisfaction. Additionally, our findings indicate that half of the artificial test suites fail to satisfactorily differentiate among the three MOEAs. This suggests that CMOPs from those suites provide limited information for the algorithm designer and are thus of little value for the benchmarking purpose. Finally, we saw that the predominant source of complexity in artificial test CMOPs is the front approximation.

With the findings from this study, we are able to validate and confirm Hypothesis 2.

8.1.3 Real-World Case Studies

The methodology proposed for analyzing CMOPs from the two perspectives has been showcased on two real-world problems: the design of cyclone dust separators and the design of elevator group control. Both of these problems have been characterized using seven ELA features that were proposed in this thesis. We have also provided some information about the complexity based on the values of these features. Already, the characterization with ELA features indicated that designing cyclone dust separators is a less demanding problem. This conclusion was later confirmed when we studied their performance space. There, we observed that the algorithms faced more difficulties. In this manner, we have demonstrated that we can analyze and predict the complexity of a given CMOP by considering its landscape.

Furthermore, we have once again observed that one of the most challenging properties of a CMOP is its negative correlation between the objectives and constraints. This additionally emphasizes that this aspect should be better represented by artificial test suites, and further research on this topic is needed.

8.2 Original Contributions

Let us recall the original scientific contributions of the thesis:

- An exact mathematical formulation for the constrained multiobjective optimization problem landscape, including a formal definition for the violation multimodality.

The rigorous mathematical formulation facilitates further consideration of landscape analysis in constrained multiobjective optimization.

- The extension of ELA features to CMOPs, as well as the design of novel features specialized for CMOPs. Special attention is devoted to ELA features for describing violation multimodality as this has not been studied previously.
- The design of a novel method for algorithm performance assessment specialized for constrained multiobjective optimization. It simultaneously monitors both the performance in approximating the Pareto front and constraint satisfaction. The method improves the state of the art, as it provides holistic information on the MOEA's performance.
- A critical investigation of the current test suites of CMOPs with respect to the feature and the performance spaces. The obtained results can be used by benchmark designers to select or generate appropriate CMOP instances based on the characteristics they want to explore.

8.3 Future Work

8.3.1 Feature Space

We are aware that the presented work for characterizing CMOPs has a shortcoming. The proposed approach to describe violation multimodality is not sufficiently scalable, and thus only low-dimensional problems were addressed in this regard. Similarly, these features cannot be used to characterize expensive real-world CMOPs as their derivation requires many function evaluations. Nevertheless, the results show that the proposed ELA features include relevant information for predicting MOEA performance, thus their exploration is relevant and should be further considered. For this reason, we suggest to extend the proposed methodology to large-scale and expensive optimization problems, as well as to further analyze the behavior of constraint handling techniques subject to various problem characteristics.

8.3.2 Performance Space

The work concerning the performance space perspective is not focused on the task of problem selection. A natural continuation would be to extend the proposed methodology to incorporate this important aspect. In particular, we suggest an extension of our methodology that would enable the identification and selection of a custom suite of existing CMOPs producing the highest differences in the performance space. Additionally, the potential of the proposed methodology in studying algorithm behavior while solving additional real-world problems should also be addressed. Measuring algorithm performance in this case is especially challenging as in a real-world scenario the Pareto front is usually unknown.

References

Barth, W. (1956). Design and layout of the cyclone separator on the basis of new investigations. *Brenn. Wärme Kraft*, *8*(4), 1–9.

Bartz-Beielstein, T., Doerr, C., van den Berg, D., 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., & Weise, T. (2020). Benchmarking in optimization: Best practice and open issues. *arXiv preprint arXiv:2007.03488v2*, 1–50. https://doi.org/10.48550/arXiv.2007.03488

Bartz-Beielstein, T., Lasarczyk, C. W., & Preuss, M. (2005). Sequential parameter optimization. *2005 IEEE Congress on Evolutionary Computation (CEC)*, 773–780. https://doi.org/10.1109/CEC.2005.1554761

Blank, J., & Deb, K. (2020). Pymoo: Multi-objective optimization in Python. *IEEE Access*, *8*, 89497–89509. https://doi.org/10.1109/ACCESS.2020.2990567

Bosman, P., & Thierens, D. (2003). The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Trans. Evol. Comput.*, *7*(2), 174–188. https://doi.org/10.1109/TEVC.2003.810761

Breiman, L. (2001). Random forests. *Mach. Learn.*, *45*, 5–32. https://doi.org/10.1023/A:1010933404324

Campelo, F., Batista, L. S., & Aranha, C. (2020). The MOEADr package: A component-based framework for multiobjective evolutionary algorithms based on decomposition. *J. Stat. Softw.*, *92*(6), 1–39. https://doi.org/10.18637/jss.v092.i06

Coker, A. K. (1993). Understand cyclone design. *Chem. Eng. Pro.*, *89*(12).

Cortes, C., & Gil, A. (2007). Modeling the gas and particle flow inside cyclone seperators. *Prog. Energy Combust. Sci.*, *33*, 409–452. https://doi.org/10.1016/j.pecs.2007.02.001

Daolio, F., Liefooghe, A., Verel, S., Aguirre, H., & Tanaka, K. (2017). Problem features versus algorithm performance on rugged multiobjective combinatorial fitness landscapes. *Evol. Comput.*, *25*(4), 555–585. https://doi.org/10.1162/evco_a_00193

Deb, K., Thiele, L., Laumanns, M., & Zitzler, E. (2002). Scalable multi-objective optimization test problems. *Proceedings of the 2002 Congress on Evolutionary Computation (CEC)*, 825–830. https://doi.org/10.1109/CEC.2002.1007032

Deb, K., & Jain, H. (2014). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: Solving problems with box constraints. *IEEE Trans. Evol. Comput.*, *18*(4), 577–601. https://doi.org/10.1109/TEVC.2013.2281535

Deb, K., Pratap, A., & Meyarivan, T. (2001). Constrained test problems for multi-objective evolutionary optimization. *International Conference on Evolutionary Multi-Criterion Optimization (EMO 2001)*, 284–298. https://doi.org/10.1007/3-540-44719-9_20

Dust distributions [Accessed: 2020, September 1]. (n.d.). https://www.ksl-staubtechnik.de/en/pruefentesten/fractions/

Eiben, A. E., & Smith, J. E. (2003). *Introduction to evolutionary computing* (1st ed.). Springer. https://doi.org/10.1007/978-3-662-05094-1

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *International Conference on Knowledge Discovery and Data Mining (KDD-96)*, 226–231.

Fan, Z., Li, W., Cai, X., Huang, H., Fang, Y., You, Y., Mo, J., Wei, C., & Goodman, E. (2019). An improved epsilon constraint-handling method in MOEA/D for CMOPs with large infeasible regions. *Soft Comput.*, *23*(23), 12491–12510. https://doi.org/10.1007/s00500-019-03794-x

Fan, Z., Li, W., Cai, X., Li, H., Wei, C., Zhang, Q., Deb, K., & Goodman, E. (2019). Difficulty adjustable and scalable constrained multiobjective test problem toolkit. *Evol. Comput.*, *28*(3), 339–378. https://doi.org/10.1162/evco_a_00259

Filipič, B., & Vodopija, A. (2021). Multiobjective optimization in the presence of constraints. *2021 IEEE Congress on Evolutionary Computation (CEC)*. https://dis.ijs.si/filipic/cec2021tutorial/

Fonseca, C. M. (1995). *Multiobjective genetic algorithms with application to control engineering problems* (Doctoral dissertation). University of Sheffield.

Gimbun, J., Chuah, T. G., Choong, T. S. Y., & Fakhru'l-Razi, A. (2005). A CFD study on the prediction of cyclone collection efficiency. *Int. J. Comput. Methods Eng. Sci. Mech.*, *6*(3), 161–168. https://doi.org/10.1080/15502280590923649

Grimme, C., Kerschke, P., Aspar, P., Trautmann, H., Preuss, M., Deutz, A. H., Wang, H., & Emmerich, M. (2021). Peeking beyond peaks: Challenges and research potentials of continuous multimodal multi-objective optimization. *Comput. Oper. Res.*, *136*. https://doi.org/10.1016/j.cor.2021.105489

Hakonen, H. M., Rong, A., & Lahdelma, R. (2004). Multiobjective optimization in elevator group control [17 pages]. *European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2004)*.

Hansen, N., Auger, A., Brockhoff, D., & Tušar, T. (2022). Anytime performance assessment in blackbox optimization. *IEEE Trans. Evol. Comput.*, *26*(6), 1293–1305. https://doi.org/10.1109/TEVC.2022.3210897

Hansen, N., Auger, A., Ros, R., Mersmann, O., Tušar, T., & Brockhoff, D. (2021). COCO: A platform for comparing continuous optimizers in a black-box setting. *Optim. Methods Softw.*, *36*(1), 114–144. https://doi.org/10.1080/10556788.2020.1808977

Hoffmann, A., & Stein, L. (2007). *Gas cyclones and swirl tubes. Principles, design and operation.* Springer.

Jain, H., & Deb, K. (2014). An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, Part II: Handling constraints and extending to an adaptive approach. *IEEE Trans. Evol. Comput.*, *18*(4), 602–622. https://doi.org/10.1109/TEVC.2013.2281534

Kerschke, P., & Grimme, C. (2017). An expedition to multimodal multi-objective optimization landscapes. *International Conference on Evolutionary Multi-Criterion Optimization (EMO 2017)*, 329–343. https://doi.org/10.1007/978-3-319-54157-0_23

Kerschke, P., & Trautmann, H. (2019). Comprehensive feature-based landscape analysis of continuous and constrained optimization problems using the R-package Flacco. *Applications in Statistical Computing: From Music Data Analysis to Industrial Quality Improvement*, 93–123. https://doi.org/10.1007/978-3-030-25147-5_7

Kerschke, P., Wang, H., Preuss, M., Grimme, C., Deutz, A., Trautmann, H., & Emmerich, M. (2016). Towards analyzing multimodality of continuous multiobjective landscapes. *International Conference on Parallel Problem Solving from Nature (PPSN XIV)*, 962–972. https://doi.org/10.1007/978-3-319-45823-6_90

Kumar, A., Wu, G., Ali, M. Z., Luo, Q., Mallipeddi, R., Suganthan, P. N., & Das, S. (2021). A benchmark-suite of real-world constrained multi-objective optimization problems

and some baseline results. *Swarm Evol. Comput.*, *67*. https://doi.org/10.1016/j.swevo.2021.100961

Lee, A. (2015). pyDOE: Design of experiments for Python [v. 0.3.8]. https://pypi.org/project/pyDOE/

Li, H., & Zhang, Q. (2009). Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *IEEE Trans. Evol. Comput.*, *13*(2), 284–302. https://doi.org/10.1109/TEVC.2008.925798

Li, J. P., Wang, Y., Yang, S., & Cai, Z. (2016). A comparative study of constraint-handling techniques in evolutionary constrained multiobjective optimization. *2016 IEEE Congress on Evolutionary Computation (CEC)*, 4175–4182. https://doi.org/10.1109/CEC.2016.7744320

Li, K., Chen, R., Fu, G., & Yao, X. (2019). Two-archive evolutionary algorithm for constrained multiobjective optimization. *IEEE Trans. Evol. Comput.*, *23*(2), 303–315. https://doi.org/10.1109/TEVC.2018.2855411

Li, Z., Zisheng, Z., & Kuotsung, Y. (1988). Study of structure parameters of cyclones. *Chem. Eng. Res. Des.*, *66*, 114–120.

Liefooghe, A., Daolio, F., Verel, S., Derbel, B., Aguirre, H., & Tanaka, K. (2020). Landscape-aware performance prediction for evolutionary multiobjective optimization. *IEEE Trans. Evol. Comput.*, *24*(6), 1063–1077. https://doi.org/10.1109/TEVC.2019.2940828

Liefooghe, A., Verel, S., Lacroix, B., Zăvoianu, A.-C., & McCall, J. (2021). Landscape features and automated algorithm selection for multi-objective interpolated continuous optimisation problems. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '21)*, 421–429. https://doi.org/10.1145/3449639.3459353

Liu, Z.-Z., & Wang, Y. (2019). Handling constrained multiobjective optimization problems with constraints in both the decision and objective spaces. *IEEE Trans. Evol. Comput.*, *23*(5), 870–884. https://doi.org/10.1109/TEVC.2019.2894743

Löffler, F. (1988). *Staubabscheiden*. Thieme. https://doi.org/10.1002/star.19890410313

Ma, Z., & Wang, Y. (2019). Evolutionary constrained multiobjective optimization: Test suite construction and performance comparisons. *IEEE Trans. Evol. Comput.*, *23*(6), 972–986. https://doi.org/10.1109/TEVC.2019.2896967

Ma, Z., & Wang, Y. (2021). Shift-based penalty for evolutionary constrained multiobjective optimization and its application. *IEEE Trans. Cybern.*, 1–13. https://doi.org/10.1109/TCYB.2021.3069814

Ma, Z., Wang, Y., & Song, W. (2021). A new fitness function with two rankings for evolutionary constrained multiobjective optimization. *IEEE Trans. Syst. Man Cybern. Syst.*, *51*(8), 5005–5016. https://doi.org/10.1109/TSMC.2019.2943973

Malan, K. M., Oberholzer, J. F., & Engelbrecht, A. P. (2015). Characterising constrained continuous optimisation problems. *2015 IEEE Congress on Evolutionary Computation (CEC)*, 1351–1358. https://doi.org/10.1109/CEC.2015.7257045

Malan, K. M., & Engelbrecht, A. P. (2014). A progressive random walk algorithm for sampling continuous fitness landscapes. *2014 IEEE Congress on Evolutionary Computation (CEC)*, 2507–2514. https://doi.org/10.1109/CEC.2014.6900576

Markon, S. (2015). A solvable simplified model for elevator group control studies. *2015 IEEE 4th Global Conference on Consumer Electronics (GCCE)*, 56–60. https://doi.org/10.1109/GCCE.2015.7398739

Markon, S., Kise, H., Kita, H., & Bartz-Beielstein, T. (2006). *Control of traffic systems in buildings*. Springer. https://doi.org/10.1007/1-84628-449-X

Mersmann, O., Bischl, B., Trautmann, H., Preuss, M., Weihs, C., & Rudolph, G. (2011). Exploratory landscape analysis. *Proceedings of the 13th Annual Conference on*

Genetic and Evolutionary Computation (GECCO '11), 829–836. https://doi.org/10.1145/2001576.2001690

Mezura-Montes, E., & Coello Coello, C. A. (2011). Constraint-handling in nature-inspired numerical optimization: Past, present and future. *Swarm. Evol. Comput.*, *1*(4), 173–194. https://doi.org/10.1016/j.swevo.2011.10.001

Moré, J. J., & Wild, S. M. (2009). Benchmarking derivative-free optimization algorithms. *SIAM J. Optim.*, *20*(1), 172–191. https://doi.org/10.1137/080724083

Muñoz, M. A., Kirley, M., & Halgamuge, S. K. (2015). Exploratory landscape analysis of continuous space optimization problems using information content. *IEEE Trans. Evol. Comput.*, *19*(1), 74–87. https://doi.org/10.1109/TEVC.2014.2302006

Muñoz, M. A., Sun, Y., Kirley, M., & Halgamuge, S. K. (2015). Algorithm selection for black-box continuous optimization problems: A survey on methods and challenges. *Inf. Sci.*, *317*, 224–245. https://doi.org/10.1016/j.ins.2015.05.010

Muschelknautz, E., & Brunner, K. (1967). Untersuchungen an Zyklonen. *Chem. Ing. Tech.*, *39*(9–10), 531–538.

Nash, S. G. (1984). *User's guide for TN/TNBC: Fortran routines for nonlinear optimization* (Technical Report No. 397). Mathematical Sciences Department, The Johns Hopkins University. Baltimore, MD.

Pedregosa, F., & et al. (2011). Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.*, *12*, 2825–2830.

Picard, C., & Schiffmann, J. (2021). Realistic constrained multiobjective optimization benchmark problems from design. *IEEE Trans. Evol. Comput.*, *25*(2), 234–246. https://doi.org/10.1109/TEVC.2020.3020046

Poursoltan, S., & Neumann, F. (2014). A feature-based analysis on the impact of linear constraints for ϵ-constrained differential evolution. *2014 IEEE Congress on Evolutionary Computation (CEC)*, 3088–3095. https://doi.org/10.1109/CEC.2014.6900572

Poursoltan, S., & Neumann, F. (2015). A feature-based comparison of evolutionary computing techniques for constrained continuous optimisation. *Neural Information Processing*, 332–343. https://doi.org/10.1007/978-3-319-26555-1_38

Prager, R. P. (2019). Pflacco: A Python interface of the R package flacco [v. 0.4]. https://pypi.org/project/pflacco/

Price, K. V., Storn, R. M., & Lampinen, J. A. (2005). *Differential evolution*. Springer. https://doi.org/10.1007/3-540-31306-0

Robič, T., & Filipič, B. (2005). DEMO: Differential evolution for multiobjective optimization. *Proceedings of the 3rd International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005)*, *3410*, 520–533. https://doi.org/10.1007/978-3-540-31880-4_36

Sahin, Y. G., Uzunbayir, S., Akcay, B., & Yildiz, E. (2013). Real-time monitoring of elevators to reduce redundant stops and energy wastage. *Proceedings of the 2013 International Conference on Systems, Control and Informatics*, 264–269.

Salcedo, R., & Cândido, M. (2007). Global optimization of reverse-flow gas cyclones: Application to small-scale cyclone design. *Sep. Sci. Technol.*, *36*(12), 2707–2731.

Schäpermeier, L., Grimme, C., & Kerschke, P. (2021). To boldly show what no one has seen before: A dashboard for visualizing multi-objective landscapes. *International Conference on Evolutionary Multi-Criterion Optimization (EMO 2021)*, 632–644. https://doi.org/10.1007/978-3-030-72062-9_50

Stadler, P. F. (2002). Fitness landscapes. *Biological evolution and statistical physics* (pp. 183–204). Springer.

Tanabe, R., & Oyama, A. (2017). A note on constrained multi-objective optimization benchmark problems. *2017 IEEE Congress on Evolutionary Computation (CEC)*, 1127–1134. https://doi.org/10.1109/CEC.2017.7969433

Tyni, T., & Ylinen, J. (2006). Evolutionary bi-objective optimisation in the elevator car routing problem. *Eur. J. Oper. Res.*, *169*, 960–977. https://doi.org/10.1016/j.ejor.2004.08.027

van der Maaten, L. (2014). Accelerating t-SNE using tree-based algorithms. *J. Mach. Learn. Res.*, *15*(1), 3221–3245.

van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *J. Mach. Learn. Res.*, *9*, 2579–2605.

Van Rossum, G., & Drake, F. L. (2009). *Python 3 reference manual*. CreateSpace.

Verel, S., Liefooghe, A., Jourdan, L., & Dhaenens, C. (2013). On the structure of multi-objective combinatorial search space: MNK-landscapes with correlated objectives. *Eur. J. Oper. Res.*, *227*(2), 331–342. https://doi.org/10.1016/j.ejor.2012.12.019

Virtanen, P., & et al. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nat. Methods*, *17*, 261–272. https://doi.org/10.1038/s41592-019-0686-2

Vodopija, A., Breiderhoff, B., Naujoks, B., & Filipič, B. (2021). Design of cyclone dust separators: A constrained multiobjective optimization perspective. *2021 IEEE Congress on Evolutionary Computation (CEC)*, 1983–1990. https://doi.org/10.1109/CEC45853.2021.9504991

Vodopija, A., & Filipič, B. (2019). CmoPy: Constrained multiobjective optimization in Python. *Slovenian Conference on Artificial Intelligence: Proceedings of the 22nd International Multiconference (IS 2019)*, 65–68.

Vodopija, A., Stork, J., Bartz-Beielstein, T., & Filipič, B. (2018). Model-based multiobjective optimization of elevator group control. *International Conference on High-Performance Optimization in Industry, HPOI 2018 : Proceedings of the 21st International Multiconference Information Society (IS 2018)*, 43–46.

Vodopija, A., Stork, J., Bartz-Beielstein, T., & Filipič, B. (2022). Elevator group control as a constrained multiobjective optimization problem. *Appl. Soft Comput.*, *115*, 108277. https://doi.org/10.1016/j.asoc.2021.108277

Vodopija, A., Tušar, T., & Filipič, B. (2021). Analyzing the diversity of constrained multi-objective optimization test suites. *Slovenian Conference on Artificial Intelligence: Proceedings of the 24th International Multiconference (IS 2021)*, 51–54.

Vodopija, A., Tušar, T., & Filipič, B. (2022). Characterization of constrained continuous multiobjective optimization problems: A feature space perspective. *Inf. Sci.*, *607*, 244–262. https://doi.org/10.1016/j.ins.2022.05.106

Vodopija, A., Tušar, T., & Filipič, B. (2024). Characterization of constrained continuous multiobjective optimization problems: A performance space perspective [accepted for publication, arXiv preprint arXiv.2302.02170]. *IEEE Trans. Evol. Comput.* https://doi.org/10.48550/arXiv.2302.02170

Vodopija, A., Janko, V., Luštrek, M., & Filipič, B. (2020). Constrained multiobjective optimization for the design of energy-efficient context recognition systems. *Bioinspired Optimization Methods and Their Applications: 9th International Conference (BIOMA 2020)*, 308–320. https://doi.org/10.1007/978-3-030-63710-1_24

Vodopija, A., Oyama, A., & Filipič, B. (2019). Ensemble-based constraint handling in multiobjective optimization. *Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO '19)*, 2072–2075. https://doi.org/10.1145/3319619.3326909

Weiß, S. (1986). *Verfahrenstechnische Berechnungsmethoden, 3: Mechanisches Trennen in fluider Phase*. VCH Verl.-Ges., Weinheim.

Zhang, Q., & Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evol. Comput.*, *11*(6), 712–731. https://doi.org/10.1109/TEVC.2007.892759

Zhang, Q., Zhou, A., Zhao, S., Suganthan, P. N., Liu, W., & Tiwari, S. (2008). *Multiobjective optimization test instances for the CEC 2009 special session and competition* (Technical report CES-487). The School of Computer Science and Electronic Engieering, University of Essex, Colchester, UK.

Zhang, Q., Liu, W., & Li, H. (2009). The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances. *Proceedings of the 2009 Congress on Evolutionary Computation (CEC)*, 203–208. https://doi.org/10.1109/CEC.2009.4982949

Zhu, Q., Zhang, Q., & Lin, Q. (2020). A constrained multiobjective evolutionary algorithm with detect-and-escape strategy. *IEEE Trans. Evol. Comput.*, *24*(5), 938–947. https://doi.org/10.1109/TEVC.2020.2981949

Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Trans. Evol. Comput.*, *3*(4), 257–271. https://doi.org/10.1109/4235.797969

Bibliography

Publications Related to the Thesis

Journal Articles

Vodopija, A., Stork, J., Bartz-Beielstein, T., & Filipič, B. (2022). Elevator group control as a constrained multiobjective optimization problem. *Appl. Soft Comput.*, *115*, 108277. https://doi.org/10.1016/j.asoc.2021.108277

Vodopija, A., Tušar, T., & Filipič, B. (2022). Characterization of constrained continuous multiobjective optimization problems: A feature space perspective. *Inf. Sci.*, *607*, 244–262. https://doi.org/10.1016/j.ins.2022.05.106

Vodopija, A., Tušar, T., & Filipič, B. (2024). Characterization of constrained continuous multiobjective optimization problems: A performance space perspective [accepted for publication, arXiv preprint arXiv.2302.02170]. *IEEE Trans. Evol. Comput.* https://doi.org/10.48550/arXiv.2302.02170

Conference Papers

Vodopija, A., Breiderhoff, B., Naujoks, B., & Filipič, B. (2021). Design of cyclone dust separators: A constrained multiobjective optimization perspective. *2021 IEEE Congress on Evolutionary Computation (CEC)*, 1983–1990. https://doi.org/10.1109/CEC45853.2021.9504991

Vodopija, A., & Filipič, B. (2019). CmoPy: Constrained multiobjective optimization in Python. *Slovenian Conference on Artificial Intelligence: Proceedings of the 22nd International Multiconference (IS 2019)*, 65–68.

Vodopija, A., Stork, J., Bartz-Beielstein, T., & Filipič, B. (2018b). Model-based multiobjective optimization of elevator group control. *International Conference on High-Performance Optimization in Industry: Proceedings of the 21st International Multiconference Information Society (IS 2018)*, 43–46.

Vodopija, A., Tušar, T., & Filipič, B. (2021). Analyzing the diversity of constrained multiobjective optimization test suites. *Slovenian Conference on Artificial Intelligence: Proceedings of the 24th International Multiconference (IS 2021)*, 51–54.

Vodopija, A., Janko, V., Luštrek, M., & Filipič, B. (2020). Constrained multiobjective optimization for the design of energy-efficient context recognition systems. *Bioinspired Optimization Methods and Their Applications: 9th International Conference (BIOMA 2020)*, 308–320. https://doi.org/10.1007/978-3-030-63710-1_24

Vodopija, A., Oyama, A., & Filipič, B. (2019). Ensemble-based constraint handling in multiobjective optimization. *Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO '19)*, 2072–2075. https://doi.org/10.1145/3319619.3326909

Other Publications

Journal Articles

Baert, A., Clays, E., Bolliger, L., De Smedt, D., Lustrek, M., Vodopija, A., Bohanec, M., Puddu, P. E., Ciancarelli, M. C., Schiariti, M., Derboven, J., Tartarisco, G., Pardaens, S., & on behalf of the HeartMan consortium. (2018). A personal decision support system for heart failure management (HeartMan): Study protocol of the HeartMan randomized controlled trial. *BMC Cardiovasc. Disord.*, *18*(1), 186. https://doi.org/10.1186/s12872-018-0921-2

Bohanec, M., Tartarisco, G., Marino, F., Pioggia, G., Puddu, P. E., Schiariti, M. S., Baert, A., Pardaens, S., Clays, E., Vodopija, A., & Luštrek, M. (2021). HeartMan DSS: A decision support system for self-management of congestive heart failure. *Expert Syst. Appl.*, *186*, 115688. https://doi.org/https://doi.org/10.1016/j.eswa.2021.115688

Clays, E., Puddu, P. E., Luštrek, M., Pioggia, G., Derboven, J., Vrana, M., De Sutter, J., Le Donne, R., Baert, A., Bohanec, M., Ciancarelli, M. C., Dawodu, A. A., De Pauw, M., De Smedt, D., Marino, F., Pardaens, S., Schiariti, M. S., Valič, J., Vanderheyden, M., ... Tartarisco, G. (2021). Proof-of-concept trial results of the HeartMan mobile personal health system for self-management in congestive heart failure. *Sci. Rep.*, *11*(1), 5663. https://doi.org/10.1038/s41598-021-84920-4

Janko, V., Reščič, N., Vodopija, A., Susič, D., De Masi, C., Tušar, T., Gradišek, A., Vandepitte, S., De Smedt, D., Javornik, J., Gams, M., & Luštrek, M. (2023). Optimizing non-pharmaceutical intervention strategies against COVID-19 using artificial intelligence. *Front. Public Health*, *11*, 1–11. https://doi.org/10.3389/fpubh.2023.1073581

Luštrek, M., Bohanec, M., Cavero Barca, C., Ciancarelli, M. C., Clays, E., Dawodu, A. A., Derboven, J., De Smedt, D., Dovgan, E., Lampe, J., Marino, F., Mlakar, M., Pioggia, G., Puddu, P. E., Rodríguez, J. M., Schiariti, M., Slapničar, G., Slegers, K., Tartarisco, G., ... Vodopija, A. (2021). A personal health system for self-management of congestive heart failure (HeartMan): Development, technical evaluation, and proof-of-concept randomized controlled trial. *JMIR Med. Inform.*, *9*(3), e24501. https://doi.org/10.2196/24501

Conference Papers

Andova, A., Vodopija, A., Cork, J., Tušar, T., & Filipič, B. (2023). An attempt at predicting algorithm performance on constrained multiobjective optimization problems. *Slovenian Conference on Artificial Intelligence: Proceedings of the 26th International Multiconference Information Society (IS 2023)*, 44–47.

Andova, A., Vodopija, A., Krömer, P., Uher, V., Tušar, T., & Filipič, B. (2022). Initial results in predicting high-level features of constrained multi-objective optimization problems. *Slovenian Conference on Artificial Intelligence: Proceedings of the 25th International Multiconference Information Society (IS 2022)*, 7–10.

Bohanec, M., Dovgan, E., Maslov, P., Vodopija, A., Lustrek, M., Puddu, P., Schiariti, M., Ciancarelli, M., Baert, A., Pardaens, S., & Clays, E. (2017). Designing a personal decision support system for congestive heart failure management. *Slovenian Conference on Artificial Intelligence: Proceedings of the 20th International Multiconference Information Society (IS 2017)*, 67–70.

Levstek, A., Silan, D., & Vodopija, A. (2020). Speech analysis as a diagnostic method for the detection of Parkinson's disease. *Slovenian Conference on Artificial Intelligence:*

Proceedings of the 23rd International Multiconference Information Society (IS 2020), 59–62.

Luštrek, M., Dovgan, E., Vodopija, A., Bohanec, M., Baert, A., S., P., & Clays, E. (2016). Aplikacija za pomoč pri telesni vadbi bolnikov s srčnim popuščanjem. *Workshop Electronic and Mobile Health: Proceedings of the 19th International Multiconference Information Society (IS 2016)*, 61–62.

Luštrek, M., Reščič, N., Janko, V., Susič, C., D. De Masi, Vodopija, A., Marinko, M., Tušar, T., Dovgan, E., Cigale, M., Gradišek, A., & Gams, M. (2021). Forecasting trends and optimizing the intervention plans against the COVID-19 pandemic: The XPRIZE competition and beyond. *Insieme Project Workshop: Proceedings of the 24th International Multiconference Information Society (IS 2021)*, 52–55.

Reščič, N., Janko, V., Susič, C., D. De Masi, Vodopija, A., Marinko, M., Tušar, T., Dovgan, E., Cigale, M., Gradišek, A., Gams, M., & Luštrek, M. (2021). Finding efficient intervention plans against Covid-19: Second place at the XPRIZE pandemic response challenge. *Proceedings of the XV International Conference ETAI 2021*, 139–143.

Tušar, T., Sever, N., Vodopija, A., & Filipič, B. (2022). A study of the performance of a fieldwork scheduling algorithm. *Slovenian Conference on Artificial Intelligence: Proceedings of the 25th International Multiconference Information Society (IS 2022)*, 43–46.

Tušar, T., Vodopija, A., & Filipič, B. (2023). Visual exploration of the effect of constraint handling in multiobjective optimization. *International Conference on Evolutionary Multi-Criterion Optimization (EMO 2023)*, 3–16. https://doi.org/10.1007/978-3-031-27250-9_1

Vodopija, A., Cvetković, B., Luštrek, M., Rudel, D., & Balorda, Z. (2016). Napovedovanje hospitalizacij pri bolnikih s srčnim popuščanjem iz podatkov telespremljanja. *Workshop Electronic and Mobile Health: Proceedings of the 19th International Multiconference Information Society (IS 2016)*, 89–92.

Vodopija, A., Mlakar, M., & Luštrek, M. (2017). Predictive models to improve the wellbeing of heart-failure patients. *Workshop on Advanced Predictive Model in Healthcare, 16th Conference on Artificial Intelligence in Medicine (AIME 2017)*, 21–24.

Vodopija, A., Tušar, T., & Filipič, B. (2018). Comparing black-box differential evolution and classic differential evolution. *Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO '18)*, 2072–2075. https://doi.org/10.1145/3205651.3208309

Biography

Aljoša Vodopija grew up in Portorož. He finished his primary and secondary education in Piran. In 2009, he moved to Ljubljana to attend the Faculty of Mathematics and Physics at the University of Ljubljana. In 2013, he received a B.Sc. degree in Mathematics with the thesis entitled *Links and Khovanov Homology*. He continued his Master's study in the same program and received his M.Sc. in Mathematics for the thesis entitled *Mortality Forecasting for Small Populations* in 2017.

During his Master's study, he joined the Department of Intelligent Systems at the Jožef Stefan Institute, Ljubljana. There he collaborated on the EU-funded project named HeartMan (Personal Decision Support System for Heart Failure Management) and on smaller national projects.

After finishing his Master's study, he continued working at the Jožef Stefan Institute as a young researcher under the supervision of Prof. Dr. Bogdan Filipič. At the same time, he enrolled in the Ph.D. program in Information and Communication Technologies at the Jožef Stefan International Postgraduate School, Ljubljana. His research involves optimization, machine learning, and model-driven decision support systems. His doctoral research focused on constrained multiobjective optimization with evolutionary algorithms.

While working on his doctoral thesis, he still contributed to the ongoing projects developing solutions for engineering design problems, including the optimization of elevator systems and finding optimal landing sites for lunar landers. He was also a member of the team that won the second place at the XPRIZE competition *Pandemic Response Challenge* in 2021. Additionally, he has also been involved in the organization of the International Multiconference Information Society (IS) and has been a member of the Slovenian Artificial Intelligence Society (SLAIS) since 2017.