Constraint Handling in RV-GOMEA

by
Johannes Koch
Student number 5610176

to obtain the degree of Master of Science
in Computer Science, track Artificial Intelligence Technology,
at the Delft University of Technology,
Faculty Electrical Engineering, Mathematics and Computer Science.

Thesis committee

Prof. dr. Peter A.N. Bosman	CWI, TU Delft, supervisor
Dr.ir. Sicco E. Verwer	TU Delft
Anton Bouter, PhD.	CWI
Renzo J. Scholman, MSc.	CWI

October 19, 2023

Delft University of Technology
Faculty EEMCS
Mekelweg 5
Delft, The Netherlands

Centrum Wiskunde &
Informatica
Evolutionary Intelligence Group
Science Park 123
Amsterdam, The Netherlands

Abstract

The Real-Valued Gene-pool Optimal Mixing Evolutionary Algorithm (RV-GOMEA) is a state-of-the-art algorithm for single-objective, real-valued optimization. As many practical applications are inherently constrained, evolutionary algorithms are equipped with constraint handling techniques to allow optimizing constrained problems. The approach currently in use with RV-GOMEA prioritizes solution feasibility over the objective value in all cases, pressuring the algorithm to find feasible solutions. However, this can be inefficient if the constrained optimum is located at the constraint boundary, as search is discouraged from exploring the search space close to infeasible solutions.

In this thesis, several well-known constraint handling techniques from literature are adapted for use with RV-GOMEA and evaluated on different benchmark problems, identifying the strengths and limitations of the various techniques. Furthermore, the inefficiency of the current technique is investigated in detail. Based on the insights gained, modifications to the existing techniques are proposed, leading to promising preliminary results.

Preface

As I reflect on the journey that led to this thesis, I can't help but feel grateful for the valuable lessons learned along the way, some of them the hard way. Initially, the plan was to implement a method from recent literature on the algorithm my thesis focuses on. The goal was to extend this to multi-objective optimization and finally to a real-world medical use case. However, it became clear that this method was not the way forward, and over time, the project became what it is now.

Despite the unexpected twists and turns, I thoroughly enjoyed the project and aim to reach my climbing goals during my next adventure.

Finally, I would like to express my heartfelt gratitude to everyone who supported and helped me throughout the project. In particular, I would like to thank my amazing supervisors Prof. dr. Peter Bosman, Anton Bouter, Ph.D. and MSc. Renzo Scholman who gave their best to keep me focused and provided support, motivation, and earnest feedback when needed. I also want to thank my partner, family, and friends for their unwavering support throughout this journey.

Johannes Koch
Delft, the Netherlands
October 19, 2023

Contents

Preface **v**

Contents **vii**

List of Figures **ix**

1 Introduction **1**

2 Background **3**
2.1 Constrained Optimization Problems 3
2.2 Evolutionary Algorithms 6
2.3 Constraint Handling Techniques 10

3 Existing Constraint Handling in RV-GOMEA **21**
3.1 Adapting Existing Techniques and Parameter Settings 22

4 Experiments and Results **29**
4.1 Benchmark Problems and Setup 29
4.2 CEC2006 34
4.3 bbob-constrained 38
4.4 Cone Problem 41
4.5 Effectiveness 45
4.6 Summary 52

5 Proposed Improvements **55**
5.1 Partially Infeasible Selection 55
5.2 Evolving Infeasible Solutions with a Partially Infeasible Population 59
5.3 Experiments 61

6 Discussion **75**

7 Conclusion and Future Work **77**

7.1 Conclusion 77
7.2 Future Work 78

Bibliography **81**

A Parameter Configuration **91**
A.1 Method 91
A.2 Parameters 92
A.3 Results 94

B CEC2006 Results **129**

C Effectiveness Results **143**

List of Figures

2.1 The optimization process of an EA. 7

3.1 The effect of stochastic acceptance during GOM on problem g07 from the CEC2006[38] benchmark problems over 25 runs. 23

3.2 The framework used for adapting the DPDE[25] approach to RV-GOMEA where r_f denotes the fraction of feasible solutions in the population. If both the percentage of feasible and infeasible solutions meet the threshold θ, the population is split into subpopulations. The feasible subpopulation optimizes the unconstrained objective value, while the infeasible population utilizes CDP. . 26

3.3 The effect of proportional selection on problem g07 from the CEC2006[38] benchmark problems over 25 runs. 26

3.4 The framework used for the second dual-population approach, based on [46]. If the main population has feasible solutions and there are enough infeasible solutions, the infeasible population is used. Using the traction strategy, the infeasible population explores the search space between the itself and the feasible population. 27

3.5 The effect of different traction strategies on problem g07 from the CEC2006[38] benchmark problems over 25 runs. 28

4.1 Different instances of the Cone problem with feasibility parameter $\theta = 90^\circ$ and $\theta = 45^\circ$ in 2D. 33

4.2 The results for the existing methods on the 2D instances of the bbob-constrained suite, grouped by the number of constraints, as described in Table 4.2. The number of total constraints is displayed above the figures. 39

4.3 The results for the existing methods on the 5D instances of the bbob-constrained suite. The 54 functions are grouped by the number of constraints and active constraints, as described in Table 4.2. The number of total constraints is displayed above the figures. 39

4.4 The results for the existing methods on the 10D instances of the bbob-constrained suite. The 54 functions are grouped by the number of constraints and active constraints, as described in Table 4.2. The number of total constraints is displayed above the figures. 40

4.5 The results for the existing methods on the 20D instances of the bbob-constrained suite. The 54 functions are grouped by the number of constraints and active constraints, as described in Table 4.2. The number of total constraints is displayed above the figures. 40

4.6 The median and interdecile range of the number of evaluations needed to reach the VTR of 10^{-10} on different Cone problem instances over 31 runs. The row corresponds to the block size used, and MP to the Marginal Product FOS. Solid lines indicate a success rate of ≥ 90%, dashed lines between 90% and 50%, dotted lines between 50% and 10% and less successful results are not shown. 43

4.7 The median and interdecile range of the number of evaluations needed to reach the VTR of 10^{-10} on different Cone problem instances over 31 runs. The row corresponds to the block size used, and MP to the Marginal Product FOS. Solid lines indicate a success rate of ≥ 90%, dashed lines between 90% and 50%, dotted lines between 50% and 10% and less successful results are not shown. 44

4.8 The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs. 47

4.9 The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs. 48

4.10 The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs. 51

5.1 The estimated distributions and AMS vectors of the CDP method using the Full FOS on the Cone 90° problem. The ellipses correspond to a Mahalanobis distance of 1, and the color gradient from white to green indicates the increasing generation, the first 24 generations are shown. 56

5.2 The outline of the proposed modification to RV-GOMEA. If the population is feasible, infeasible solutions encountered during the previous variation step are subject to a second selection in order to mitigate the bias towards feasibility induced by CDP. 57

5.3 The effect of different parameters on problem g07 from the CEC2006 problems. The remaining parameters are set to the best performing version in Table A.9. The values at the top/bottom of the bar correspond to the feasible rate and success rate respectively. 58

5.4 The outline of the proposed approach based on DP-O, where a partially feasible population is used instead of two subpopulations. In this mixed population, selection and variation is split by feasibility and model building is done jointly. Additionally, previously seen feasible solutions that were replaced by infeasible solutions, indicated by the dotted lines, are reconsidered during the added offspring selection step. 60
5.5 The results for the bbob-constrained suite, grouped by the number of constraints, as described in Table 4.2. The number of total constraints is displayed above the figures. 63
5.6 The results for the bbob-constrained suite. The 54 functions are grouped by the number of constraints and active constraints, as described in Table 4.2. The number of total constraints is displayed above the figures. 63
5.7 The results for the bbob-constrained suite. The 54 functions are grouped by the number of constraints and active constraints, as described in Table 4.2. The number of total constraints is displayed above the figures. 64
5.8 The results for the bbob-constrained suite. The 54 functions are grouped by the number of constraints and active constraints, as described in Table 4.2. The number of total constraints is displayed above the figures. 64
5.9 The median and interdecile range of the number of evaluations needed to reach the VTR of 10^{-10} on different Cone problem instances over 31 runs. The row corresponds to the block size used, and MP to the Marginal Product FOS. Solid lines indicate a success rate of $\geq 90\%$, dashed lines between 90% and 50%, dotted lines between 50% and 10% and less successful results are not shown. 67
5.10 The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs. 69
5.11 The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs. 70
5.12 The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs. 71
5.13 The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs. 72

A.1 The effect of the population size on the number of evaluations needed, for population sizes reaching the approximation target in at least 95% of 25 runs. . 95
A.2 The effect of the $b_{control}$ parameter of the EPS method over all runs performed. 102
A.3 The effect of the cp parameter of the EPS method over all runs performed. . . 103
A.4 The effect of the $b_{control}$ parameter of the IEPS method over all runs performed. 104
A.5 The effect of the cp parameter of the IEPS method over all runs performed. . . 105
A.6 The effect of the α parameter of the IEPS method over all runs performed. . . . 106
A.7 The effect of the β parameter of the IEPS method over all runs performed. . . . 107

A.8 The effect of the acceptance strategy of the SR method over all runs performed. 108
A.9 The effect of the P_f parameter of the SR method over all runs performed. . . . 109
A.10 The effect of the k parameter of the R-BS method over all runs performed. . . 110
A.11 The effect of the k parameter of the R-R method over all runs performed. . . . 111
A.12 The effect of the θ_s parameter of the R-R method over all runs performed. . . . 112
A.13 The effect of the θ_f parameter of the DP-O method over all runs performed. . 113
A.14 The effect of the proportional or full scheme for the subpopulations of the DP-O method over all runs performed. 114
A.15 The effect of the θ_s parameter of the DP-O method over all runs performed. . . 115
A.16 The effect of the different traction strategies for the DP-T method over all runs performed. CDP is no traction strategy, D2F corresponds to the distance to the feasible population and FSI to the injection of the feasible selection. 116
A.17 The effect of the τ_{min} parameter of the DP-T method over all runs performed. . 117
A.18 The effect of the "feasible selection injection percentile"-parameter of the DP-T method over all runs performed. 118
A.19 The effect of the proportional and full subpopulation schemes of the PI method over all runs performed. 119
A.20 The effect of the search strategy used in the infeasible (sub)populations of the PI method over all runs performed. 120
A.21 The effect of using a single joint distribution vs. a distribution per subpopulation for the PI method over all runs performed. 121
A.22 The effect of reconsidering feasible solutions lost during variation for the PI method over all runs performed. 122
A.23 The effect of the θ_i parameter of the PIS method over all runs performed. . . . 123
A.24 The effect of the P_f parameter of the PIS method over all runs using SR for either searching in the infeasible region or selecting the infeasible solutions used. 124
A.25 The effect of replacing feasible solutions or adding the infeasible solutions for PIS method over all runs performed. 125
A.26 The effect of pruning for the PIS method over all runs performed. 126
A.27 The effect of the strategy used to select the infeasible solutions for the PIS method over all runs performed. D2F corresponds to selection based on the proximity to the feasible solutions and NDS to non-dominated-sorting. 127
A.28 The effect of using SR in the infeasible region for the PIS method over all runs performed. 128

C.1 The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs. 144
C.2 The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs. 145

C.3 The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs. 146
C.4 The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs. 147
C.5 The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs. 148
C.6 The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs. 149
C.7 The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs. 150
C.8 The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs. 151
C.9 The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs. 152
C.10 The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs. 153
C.11 The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs. 154
C.12 The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs. 155
C.13 The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs. 156
C.14 The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs. 157
C.15 The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs. 158
C.16 The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs. 159

C.17 The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs. 160
C.18 The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs. 161
C.19 The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs. 162
C.20 The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs. 163
C.21 The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs. 164
C.22 The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs. 165

Chapter 1

Introduction

Optimization is an indispensable tool that underpins numerous aspects of our daily lives and enables the technologies we rely on. From scheduling public transport [12], designing computer chips [77] and finding neural network architectures [39] to cancer treatment planning [4], optimization algorithms serve as the workhorses that enable finding good solutions for complex problems. Naturally, many of these problems are inherently accompanied by constraints. For instance, solution candidates violating the laws of physics or not adhering to regulations are not useful in practice. Moreover, these constraints are often at odds with the goal, causing the best solutions to be found at the limit of what is feasible, the constraint boundaries. Due to the ubiquity of real-world problems involving constraints, constraint handling techniques are an active field of research [45, 35, 43, 50], where algorithmic advances enable downstream improvements for the practical use cases.

Many problems are complex and not fully understood, creating a need for optimization techniques that are able to perform well in scenarios where the problem at hand is a black box and there exists no problem specific approach. Evolutionary Algorithms (EAs) are a class of algorithms well suited for tackling these complex problems as they make few assumptions about the underlying problem, leading to a broad applicability [59]. One well-known example of EAs being applied to a real-world problem is [30], where antennas for aerospace applications are evolved that were used in NASA missions.

One such EA that is amongst the current state-of-the-art for large-scale optimization with limited problem knowledge is the Real-Valued Gene-pool Optimal Mixing Evolutionary Algorithm (RV-GOMEA)[8]. The multi-objective version of this algorithm is used in for real-world medical use cases [4, 8, 1]. These problems include constraints, however, constraint handling for RV-GOMEA has not been explicitly studied yet. This thesis aims to take a step towards closing the gap between state-of-the-art optimization and constraint handling techniques. Improvements in constraint handling of RV-GOMEA represent a step towards downstream improvements for the practical use cases the algorithm is used for.

When optimizing constrained problems, it is desirable to end up with solutions adhering to all constraints. Therefore, RV-GOMEA currently prioritizes such feasible solutions over solutions that do violate constraints. During optimization, a step of selecting the best solutions found so far is performed to identify a promising region for future search. Even when knowing very little about the problem at hand, it often holds true that making small changes to a solution tends to result in solutions of similar quality, making search in the vicinity of good solutions effective. However, for constrained problems, this can lead to ineffective search near constraint boundaries. By selecting and thus searching only near solutions adhering to the constraints, search is discouraged from approaching the constraint boundaries. Theoretical results show that making use of infeasible solutions throughout the optimization process can be the deciding factor between exponential or polynomial algorithm complexity [78]. As the best solutions are often located at the constraint boundary, efficient search near constraint boundaries is desirable [43, 25, 31].

This thesis aims to adapt and evaluate existing well-known constraint handling techniques within the context of RV-GOMEA. Additionally, the shortcomings of the constraint domination principle[17], the currently used approach, and other established methods[54, 64, 20, 25, 58, 22] will be analyzed. Building upon this analysis, novel approaches will be proposed, designed specifically for RV-GOMEA in order to facilitate efficient search near constraint boundaries. Finally, the effectiveness of the proposed approaches will be evaluated on various benchmark suites[38, 29] and problems, providing insights into their strengths and limitations in different optimization scenarios. By doing so, this thesis aims to improve the effectiveness of RV-GOMEA for constrained optimization problems and to pave the way towards possible downstream improvements for real-world use cases such as cancer treatment planning [4].

This is split up into the following main research questions:

1. How do existing constraint handling techniques perform when combined with RV-GOMEA?
2. Does the constraint domination principle, the currently used approach, perform efficient search near constraint boundaries, and if not, why?
3. How can existing approaches be improved upon to allow RV-GOMEA to search near constraint boundaries more effectively?

The structure of the thesis is as follows. First, the necessary background and terminology is introduced in Chapter 2. In Chapter 3, well-known constraint handling techniques from literature are adapted to RV-GOMEA, followed by experiments in Chapter 4. The experiments evaluate the different techniques on various benchmark problems, and the effectiveness of search near constraint boundaries is analyzed. Chapter 5 presents modifications to existing approaches designed to facilitate more efficient search near constraint boundaries. These new approaches and existing techniques are then empirically validated on several benchmark suites and problems. The work performed as well as the limitations are discussed in Chapter 6, before Chapter 7 concludes this thesis.

Chapter 2

Background

This chapter first presents the general problem formulation for single-objective constraint optimization problems (COPs), properties of such problems, and the notion of black-box optimization this thesis focuses on. Furthermore, the general working principles and terminology of evolutionary algorithms (EAs) are introduced, with special attention on the real-valued gene-pool optimal mixing evolutionary algorithm (RV-GOMEA). Finally, an overview of well-known constraint handling techniques for EAs is provided.

2.1 Constrained Optimization Problems

A constrained optimization problem (COP) consists of an objective function, a search space, and the constraints. Without loss of generality, this can be formulated as

$$
\begin{aligned}
\text{minimize} \quad & f(x) \\
\text{subject to} \quad & \\
& g_i(x) \leq 0 \quad \text{for } i = 1, 2, \ldots, p \\
& h_j(x) = 0 \quad \text{for } j = 1, 2, \ldots, q \\
\text{where} \quad & x \in \Omega
\end{aligned}
\tag{2.1}
$$

where $f : \Omega \to \mathbb{R}$ is the objective function and Ω is the set of all possible solutions, i.e. the search space. If both $p = q = 0$, the problem is unconstrained. The two types of constraint are inequality constraints $g_i(x)$ and equality constraints $h_j(x)$. A solution $x \in \Omega$ is called feasible if it satisfies all constraints, otherwise, it is infeasible. Furthermore, let $\mathcal{M} \subseteq \Omega$ be the subset containing all feasible solutions, the feasible set. Then the goal is to find or approximate the best feasible solution $x^* \in \mathcal{M}$, such that $f(x^*) \leq f(x)$ for all $x \in \mathcal{M}$. If $g_i(x^*) = 0$ holds, the constraint is called active. By extension, inequality constraints are always active, as $h_j(x^*) = 0$ must hold at the constrained optimum x^*.

For real-valued problems, a solution $\boldsymbol{x} \in \Omega$ is a vector of length l, and each value is called a decision variable or problem variable. Furthermore, the feasible set $\mathcal{M}$ is often subdivided into possibly multiple feasible regions, where a feasible region is a subset of $\mathcal{M}$ where all solutions are connected in the search space.

Domain The domain of the search space Ω can have many forms, often classified as either discrete, continuous, or mixed. Discrete search spaces consist of distinct values, whereas continuous search spaces are dense, i.e. no matter how close together two different values are, there is always another value in between them. Continuous search spaces are always infinite, while discrete search spaces can be finite. An example would be the set of possible next moves in a chess game for the discrete case, and all possible weights for a neural network in the continuous case. Mixed search spaces are a combination of discrete and continuous search spaces, for example, symbolic regression involves finding a mathematical expression consisting of a discrete term structure and continuous coefficients [67]. In this thesis, the focus lies on real-valued optimization, i.e. the search space can be stated as $\Omega \subseteq \mathbb{R}^l$.

2.1.1 Properties of Optimization Problems

Optimization problems have different properties, that in turn greatly impact the difficulty of finding good solutions and which algorithmic approaches are well-suited.

Convexity The objective function, as well as the feasible regions, can be convex or non-convex. If a feasible region is convex, any solution directly in between two different solutions within the feasible region must also be part of that feasible region. Similarly, convexity in the objective function states that the objective value of solutions on the line segment between any two distinct solutions in the search space must be less or equal to the objective value of those two solutions. How this property can be exploited for optimization is studied by the field of convex optimization [11].

Linearity A constraint or an objective function is called linear if it can be formulated as a summation of terms that only contain up to one decision variable raised to the power of 1. Non-linear problems on the other hand place no restriction on the structure of the objective function and constraints. If both the objective function and the constraints are linear, then convexity is implied, and the problem can be solved optimally in polynomial time for continuous decision variables [33].

Smoothness If the objective function or a constraint is continuously differentiable at least once in the search space, it is called smooth. This allows utilizing gradient-based techniques for efficiently finding at least local minima of the objective function or infeasibility. On the other hand, if a problem is non-smooth, then the objective or constraint function may contain irregularities such as discontinuities or sharp corners where the gradient is undefined.

Multi-Modality The objective function and constraints of a problem may have multiple possible local minima, where surrounding solutions have the same or worse objective value. Problems with a unique minimum are called uni-modal, and problems with multiple minima are called multi-modal. Since the global optimum is usually desired, this introduces an inherent tradeoff between exploring the whole search space and finding good local solutions. Without any further knowledge about the problem, an exhaustive search needs to be performed to guarantee finding the global optimum. However, this is often intractable and expensive compared to finding the nearest local optimum given an initial solution. Because of this, a distinction is made between local and global optimization, where global approaches such as evolutionary algorithms are more robust w.r.t. escaping local optima and finding the global optima.

2.1.2 Types of Optimization

In practice, not only the best solution but also the most efficient way of finding that optimum is desired. Depending on the type of problem at hand and how much is known about the relevant instances, different optimization approaches are needed and often classified as white-box, grey-box, and black-box optimization.

White Box Optimization The white-box optimization (WBO) approach assumes everything about the problem is known, often making it possible to solve the problem analytically or to provide optimality or complexity guarantees. Furthermore, by exploiting problem properties, efficient problem specific optimization approaches can be constructed.

Black Box Optimization Contrary to WBO, in a black box optimization (BBO) nothing about the problem is known other than an evaluation function, returning the objective value and feasibility for a given solution. Naturally, no optimality or complexity guarantees can be provided. Furthermore, the optimizers used are subject to the no free lunch theorems, stating that there cannot be an optimization algorithm that outperforms other algorithms on all problems [73]. Nonetheless, most practical problems satisfy the assumption that solutions close together in the search space tend to have similar objective values.For this type of problem, metaheuristics such as evolutionary algorithms are well-suited.

Grey Box Optimization Many real-world problems do not fully belong to either BBO or WBO, since there is some knowledge or a heuristic available but too limited for WBO approaches. In the case of this thesis, the relevant grey box optimization (GBO) scenario is that the evaluation function can be decomposed. This allows performing more efficient partial evaluations, where only a fraction of a full evaluation needs to be performed when only a few decision variables are changed.

Only the BBO and GBO scenarios are considered in this thesis, as the focus lies on RV-GOMEA, a state-of-the-art EA excelling at large-scale GBO optimization[8]. This algorithm is designed to learn and exploit the underlying structure often present in optimization problems, even though none of the previously introduced properties can be assumed.

2.2 Evolutionary Algorithms

This section provides an overview of the general working principles and terminology of Evolutionary Algorithms (EAs) for real-valued optimization. In addition, the model-based EA this thesis focuses on, RV-GOMEA, is introduced in detail.

2.2.1 General Working Principles of EAs

EAs are a class of algorithms that simulate and draw inspiration from the principles of natural evolution. Naturally, the terminology used in EA literature often reflects their biological origin. For instance, solutions are referred to as *individuals*, and the objective function is known as the *fitness function*. The main idea behind EAs is to improve the fitness of a population of individuals through repeated application of selection and variation. Note that improving fitness w.r.t. to the introduced COPs translates to decreasing the objective value.

Selection A key concept of evolutionary biology is natural selection, famously known as "Survival of the fittest". In EAs, selection generally chooses a limited number of individuals with above average fitness, allowing them to contribute to the next generation. Selection introduces an implicit pressure towards finding better solutions and encourages the propagation of promising combinations of genes linked to better fitness, often referred to as *building blocks*.

Variation While selection steers the optimization process towards better solutions, variation is responsible for generating new individuals by modifying the existing population. This process mimics the concept of reproduction and genetic diversity in natural evolution. EAs typically employ two main operators for variation: *recombination* and *mutation*. Recombination typically combines the parameters of multiple *parent* solutions into new *offspring* solutions. Mutation, on the other hand, introduces small random changes to individual variables, mimicking genetic mutations in nature and maintaining diversity within the population.

Generations The optimization process typically progresses through a series of generations. Starting from an initial population, each generation involves performing selection and variation, thereby forming the next population. Figure 2.1 depicts this principle. After variation, the individuals need to be evaluated using the fitness function before selection can be performed. This is repeated until a termination condition is met. Common termination conditions include achieving a certain fitness threshold, convergence of the population towards a single solution or fitness value, and reaching a predefined computational budget. The computational budget is often defined in the number of evaluations or computation time.

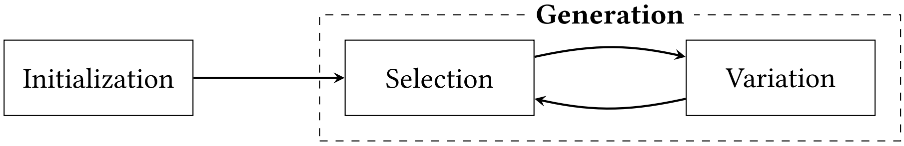


Figure 2.1: The optimization process of an EA.

2.2.2 The Real-Valued Gene-pool Optimal Mixing Evolutionary Algorithm

EAs are suitable for a wide range of optimization problems due to their simplicity and robustness. However, while the recombination of "fit" parents likely leads to improvements, inefficient mixing of building blocks can lead to exponential increases in the population size or time needed to solve a problem [65]. Hence, identifying and effectively mixing linked building blocks is crucial for achieving efficient and scalable EAs. Various so-called Model-based Evolutionary Algorithms (MBEAs) aim to do precisely this by modeling and exploiting linkage information [21]. One class of MBEAs that explicitly models problem dependencies are Gene-pool Optimal Mixing Evolutionary Algorithms (GOMEAs) [8, 21]. These algorithms have shown impressive performance on benchmarks [66, 8] and several practical use cases, such as optimizing radiotherapy treatment plans [8], symbolic regression [66] and radiotherapy dose reconstruction [68]. In this thesis, the focus lies on constrained real-valued optimization, hence the real-valued GOMEA variant, RV-GOMEA, is introduced. First, the main components are described before the full algorithm is presented.

Modeling Linkage A core component of RV-GOMEA is the modeling of linkage information. To model building blocks, or *linkage*, a structure capable of capturing which variables belong together is needed. For this, RV-GOMEA uses a structure called *family of subsets* (FOS), denoted $\mathcal{F}$. The FOS $\mathcal{F}$ is a set that contains all the modeled linkage information. Each subset $\mathcal{F}_i \in \mathcal{F}$ represents one presumed building block as the set of the corresponding decision variable indices in the solution vector $\boldsymbol{x} \in \mathbb{R}^l$. Formally, $\mathcal{F} := \{\mathcal{F}_i \mid \mathcal{F}_i \subseteq \mathcal{I} \land \mathcal{F}_i \neq \emptyset \text{ for } i = 0, 1, \ldots\} \subset \wp(\mathcal{I})$, where $\mathcal{I} := \{0, 1, \ldots, l-1\}$ is the set of decision variable indices and $\wp(\mathcal{I})$ is the power set of all indices. Generally, every decision variable is assumed to be of importance and thus part of at least one linkage set, i.e. $\forall i \in \mathcal{I}$ there exists a subset $\mathcal{F}_j \in \mathcal{F}$ such that $i \in \mathcal{F}_j$. The linkage model can either be learned during optimization or supplied by the user, potentially constructed using available knowledge about the problem [8].

During evolution, variation is applied to the decision variables represented by each FOS subset separately. By doing so, the effect of a single building block on the fitness of an individual can be isolated. This eliminates cases where during variation improvements in some building blocks can mask disruption of other building blocks. The full FOS, i.e. $\mathcal{F}_{\text{Full}} := \{\mathcal{I}\}$, varies all decision variables together. On the other end, the univariate structure $\mathcal{F}_{\text{Univariate}} := \{\{i\} \mid i \in \mathcal{I}\}$ treats all variables independently. Such a FOS structure where the problem is fully decomposed into disjoint FOS subsets is called *marginal* and referred to as *marginal product* FOS.

If the FOS is learned, this is done by performing hierarchical clustering using the estimated correlation matrix of the previously selected solutions as an indication of similarity. This results in a dendrogram, which is then directly used as the FOS structure for the next generation. This FOS structure is referred to as *linkage tree* (LT) [8].

Distribution Estimation Apart from the linkage modeling, the recombination operators of another EA, AMaLGaM [7], are used in RV-GOMEA [9]. AMaLGaM works by combining the information of all selected solutions into a multivariate normal distribution. New offspring solutions are then sampled from this distribution. Since the distribution is estimated from solutions with high fitness, sampling from the distribution likely corresponds to performing search in regions with high fitness.

In RV-GOMEA, variation is applied per linkage set, and thus a separate distribution $\mathcal{N}(\hat{\boldsymbol{\mu}}_i, \hat{\Sigma}_i)$ is maintained for each FOS subset $\mathcal{F}_i \in \mathcal{F}$. Let $\mathcal{S}$ be the set of selected solutions. Each distribution is then obtained using the maximum likelihood estimate of the selection $\mathcal{S}$, where $\hat{\boldsymbol{\mu}}_i \in \mathbb{R}^{|\mathcal{F}_i|}$ is the selection mean and $\hat{\Sigma}_i \in \mathbb{R}^{|\mathcal{F}_i| \times |\mathcal{F}_i|}$ the selection covariance. Let $j, k \in \mathcal{F}_i$, then the $|\mathcal{F}_i|$-dimensional multivariate normal distribution is estimated using:

$$\left(\hat{\boldsymbol{\mu}}_i\right)_j = \frac{1}{|\mathcal{S}|} \sum_{\boldsymbol{x} \in \mathcal{S}} \boldsymbol{x}_j \tag{2.2}$$

$$\left(\hat{\Sigma}_i\right)_{(j,k)} = \frac{1}{|\mathcal{S}|} \sum_{\boldsymbol{x} \in \mathcal{S}} \left(\boldsymbol{x}_j - \left(\hat{\boldsymbol{\mu}}_i\right)_j\right)\left(\boldsymbol{x}_k - \left(\hat{\boldsymbol{\mu}}_i\right)_k\right) \tag{2.3}$$

During variation, the corresponding distribution is used to sample new values for each decision variable in the FOS subset.

Anticipated Mean Shift The selection $\mathcal{S}$ consists of the best solutions in the population, and hence tends to approximate the contours of the objective function. Subsequently, the estimated distributions will focus the search along the contours as well. Over multiple generations, this results in rapidly shrinking distributions. This is desired and efficient if the best solutions are located near the distribution mean. However, if the best solutions are located perpendicular to the contours, e.g. for slope-like fitness landscapes, then this effectively halts the optimization progress. To prevent this and to allow the distribution to reorient itself perpendicular to the contours, AMS was introduced by [6]. Let $\mu_{\mathcal{F}_i}^{Shift}$ be the difference of the current and previous selection means, representing the direction the mean has shifted. After sampling new solutions for subset $\mathcal{F}_i$, a fraction $\frac{1}{2}\tau$ of the individuals are shifted by $\delta^{AMS}\mu_{\mathcal{F}_i}^{Shift}$ with $\delta^{AMS} = 2$. Furthermore, after all FOS subsets have been subject to variation, the same $\frac{1}{2}\tau$ solutions are shifted by $\delta^{AMS}\mu^{Shift}$ to allow a shift in all decision variables.

Note that both AMaLGaM and RV-GOMEA do employ other mechanisms, such as periodical re-evaluations to mitigate accumulating numerical errors due to hardware limitations, Adaptive Variance Scaling (AVS), Standard Deviation Ratio (SDR), or Forced Improvements (FI) [9, 8, 7]. For instance, AVS and SDR together prevent premature shrinking of the distribution by adaptively scaling the covariance when improvements are found far from the estimated distribution mean, as explained in [5] and [6]. Compared with AMS, AVS and SDR prevent the center bias of the normal distribution when it is counterproductive, while AMS allows the distribution to reorient itself.

Gene-pool Optimal Mixing In RV-GOMEA, variation is applied per FOS subset for all solutions, as shown in Algorithm 1. Improvements are accepted, otherwise, there is a probability of $p^{accept} = 0.05$ of keeping non-improvements. This procedure of varying one building block at a time and only accepting improving changes is called *gene-pool optimal mixing* (GOM) [21].

Compared to more traditional EAs, GOM has two key differences. First, the purpose of selection is effectively split into two steps. Selection in RV-GOMEA guides the optimization and determines where search is performed, but the decision of whether a change survives is made during GOM, per individual.Furthermore, if all decision variables are varied at once, this is generally noisy. It is not possible to know which change of which decision variable led to an improvement. In addition, an improvement in one building block may be hidden by other variables simultaneously deteriorating, leading to a good building block potentially not surviving selection, or bad ones surviving. RV-GOMEA mitigates this by varying and accepting changes per FOS subset, denoising this effect to an extent at the expense of additional evaluations. Instead of one evaluation per offspring, an evaluation is needed per modified subset to see the effect of the partial change. This makes RV-GOMEA especially suited for GBO settings permitting partial evaluations, where there is minimal computational overhead over doing one full evaluation [8].

Algorithm 1: The gene-pool optimal mixing procedure.

procedure *GOM*:
    Input: Solution $\boldsymbol{x}$, FOS subset $\mathcal{F}_i$
    $\boldsymbol{y} \leftarrow \boldsymbol{x}$
    $\boldsymbol{y}[\mathcal{F}_i] \leftarrow \mathcal{N}\left(\hat{\boldsymbol{\mu}}_{\mathcal{F}_i}, \hat{\Sigma}_{\mathcal{F}_i}\right)$
    if $\boldsymbol{x} \in$ *AMS solutions* **then**
        $\boldsymbol{y}[\mathcal{F}_i] \leftarrow \boldsymbol{y}[\mathcal{F}_i] + \delta^{AMS} \hat{\boldsymbol{\mu}}^{Shift}_{\mathcal{F}_i}$
    evaluate $\boldsymbol{y}$
    if $\boldsymbol{y}$ *is better than* $\boldsymbol{x}$ *or* $\mathcal{U}(0,1) < p^{accept}$ **then**
        $\boldsymbol{x} \leftarrow \boldsymbol{y}$

The Full Algorithm Together, these concepts make up the main building blocks of RV-GOMEA as shown in Algorithm 2. The algorithm operates on a population $\mathcal{P}$ of size n. The set of selected solutions $\mathcal{S} \subset \mathcal{P}$ is obtained using *truncation selection*, i.e., the best $\lfloor \tau n \rfloor$ individuals in $\mathcal{P}$ are chosen, with $\tau = 0.35$. These selected solutions are then used for learning the linkage structure of the problem and to estimate Gaussian distributions to sample new building blocks from. Furthermore, the best $n^{elitist} = 1$ solutions are preserved and not varied throughout the generation. All other solutions are subject to variation. First, GOM is performed followed by the Anticipated Mean Shift (AMS), which is applied to a subset of the population.

Algorithm 2: The general structure of RV-GOMEA as per [9].

procedure *RVGOMEA*:
    $\mathcal{P} \leftarrow$ evaluated initial population
    $t \leftarrow 0$
    while *not terminated* **do**
        $\mathcal{S} \leftarrow \lfloor \tau n \rfloor$ best individuals in $\mathcal{P}$
        $\mathcal{F} \leftarrow$ learn FOS from $\mathcal{S}$ `// (only if the FOS is learned)`
        for $\mathcal{F}_i \in \mathcal{F}$ **do** `// estimate FOS distributions`
            $\hat{\boldsymbol{\mu}}_{\mathcal{F}_i}(t), \hat{\Sigma}_{\mathcal{F}_i}(t) \leftarrow$ maximum-likelihood estimate of $\mathcal{S}$
            $\hat{\boldsymbol{\mu}}^{Shift}_{\mathcal{F}_i}(t) \leftarrow \hat{\boldsymbol{\mu}}_{\mathcal{F}_i}(t) - \hat{\boldsymbol{\mu}}_{\mathcal{F}_i}(t-1)$
        for $\mathcal{F}_i \in \mathcal{F}$ **do** `// perform GOM`
            for $\boldsymbol{x} \in \mathcal{P} \setminus n^{elitist}$ *best solutions* **do**
                GOM($\boldsymbol{x}$, $\mathcal{F}_i$)
        for $\boldsymbol{x} \in$ *AMS solutions* **do** `// perform full AMS`
            $\boldsymbol{y} \leftarrow \boldsymbol{x} + \delta^{AMS} \hat{\boldsymbol{\mu}}^{Shift}$
            evaluate $\boldsymbol{y}$
            if $\boldsymbol{y}$ *is better than* $\boldsymbol{x}$ *or* $\mathcal{U}(0,1) < p^{accept}$ **then**
                $\boldsymbol{x} \leftarrow \boldsymbol{y}$
        $t \leftarrow t + 1$

Population Sizing The size of the population is an important parameter. Too small population sizes are more likely to get stuck in local optima since the solutions do not represent the fitness landscape well enough. Large population sizes represent the problem well and are less subject to the stochasticity of an EA but at the cost of efficiency. Naturally, the right population size depends on multiple factors such as the problem, the dimensionality, and the linkage model used. There has been extensive testing of the relation between these different aspects, leading to population size recommendations for the different FOS types [5, 6, 7]. These recommended sizes are used in this thesis.

2.3 Constraint Handling Techniques

This section will first introduce the topic of constraint handling techniques (CHTs) for EAs and provide an overview of the general types of techniques used. Then, common approaches of dealing with multiple constraints and equality constraints are discussed. Finally, well-known constraint handling techniques from literature as well as current areas of research are presented.

2.3.1 Taxonomy and Properties

EAs are generally unconstrained optimizers, however, many real-world problems involve constraints. Thus, how to best handle constraints with EAs has been subject to considerable research over the years and is an active field [45, 13, 55, 43, 14, 50, 35]. The approaches that have emerged can roughly be classified into the following categories based on their operating principles:

- *Penalty functions*[20]. These methods aim to turn COPs into unconstrained optimization problems by adding a penalty term to the fitness of infeasible solutions.
- *Separation of objective and constraints*[17, 61, 22, 54]. These approaches generally prioritize feasible over infeasible solutions and are also referred to as *feasibility oriented*.
- *Repair operators*[57, 55, 58]. These approaches attempt to turn infeasible solutions into feasible ones, often using similar feasible solutions.
- *Segregational methods*[25, 46]. This category involves approaches that handle feasible and infeasible solutions differently, often inspired by multi-objective and coevolutionary optimization concepts.
- *Other methods*[41]. Ensemble methods and other hybrid techniques making use of multiple CHT concepts fall into this category, as well as problem specific methods.

In [20], the different CHTs are divided into two groups, penalty functions that handle the constraints at the fitness level and approaches that handle the constraints in the search space by biasing towards feasible solutions.

For practical problems, there is often at least some available knowledge about the problem at hand, in turn allowing problem specific constraint handling approaches or at least making assumptions about properties of the problem which can subsequently be exploited. This thesis does not focus on a particular problem, hence these types of approaches and necessary assumptions are not further considered. Furthermore, in [13] the following desired properties of CHTs are identified:

- *Generality.* In a BBO setting, the properties of the problem are generally not known, hence CHTs should ideally work with any kind of problem.
- *Parameter robustness.* Many CHTs introduce new parameters that need to be tuned for each problem in order to achieve good performance. Ideally, the method should work out of the box, without the need for fine-tuning parameters.
- *Well-known limitations.* There likely is no CHT that is superior on all problems. Hence, an understanding of the limitations as well as when a CHT is applicable is important for achieving excellent performance on a specific problem.

- *Efficiency.* In order to make solving large-scale problems and settings where fitness evaluations are expensive tractable, effective use of the obtained information is crucial.

These properties are conflicting, for instance, an approach making specific assumptions about a problem may be very efficient but not generally applicable.

2.3.2 Handling of Multiple and Equality Constraints

Real-world problems often involve several constraints, however, most CHTs considered work on a notion of overall feasibility rather than individual constraints. Hence, a way of aggregating the different constraint violations into a single *constraint value* $v(\boldsymbol{x})$ representing the *feasibility* of a solution $\boldsymbol{x}$ is needed. In general, this aggregation is done in a way that the constraint value is > 0 if a solution is infeasible and ≤ 0 otherwise.

Commonly, equality constraints are rewritten as inequalities using the absolute difference to a tolerance threshold δ, i.e. $h_j(\boldsymbol{x}) = 0$ becomes $\left|h_j(\boldsymbol{x})\right| - \delta \leq 0$. This turns a problem with p inequality and q equality constraints into a problem with $p+q$ inequality constraints. Formally, let $\Psi : \Omega \rightarrow \mathbb{R}$ be defined as the function indicating the constraint violation for each of the $p+q$ constraints:

$$\Psi_i(\boldsymbol{x}) = \begin{cases} g_i(\boldsymbol{x}) & \text{if } i \leq p \\ \left|h_{i-p+1}(\boldsymbol{x})\right| - \delta & \text{otherwise} \end{cases} \tag{2.4}$$

Then one common way to aggregate the constraints is to simply sum the individual constraint violations as follows:

$$v(\boldsymbol{x}) = \sum_{i=1}^{p+q} \max\{0, \Psi_i(\boldsymbol{x})\} \tag{2.5}$$

Note that negative constraint values, i.e. satisfied conditions with leeway, are masked using $\max\{0, \ldots\}$ to ensure that such constraints do not compensate for actual constraint violations. This approach is also used in this thesis. However, to satisfy the assumptions of the Augmented Lagrangian approach introduced later in this chapter, the constraint value for feasible individuals is allowed to be negative:

$$v_{modified}(\boldsymbol{x}) = \begin{cases} v(\boldsymbol{x}) & \text{if } v(\boldsymbol{x}) > 0 \\ \sum_{i=1}^{p+q} \Psi_i(\boldsymbol{x}) & \text{otherwise} \end{cases} \tag{2.6}$$

Other methods include using the mean of the constraint violations, considering only the value of the most violated constraint or a combination. As the numerical value of the individual constraints is potentially scaled unequally, ranking based approaches have also been used [41, 32, 15, 16]. In [24], an entirely different approach is used, where the feasibility is defined relative to another solution instead of an absolute approach. Approaches considering individual constraints also exist, e.g. [19], but are not further considered in this thesis. Furthermore, all these aggregation methods assume well-conditioned constraint violation values, which may not be the case in a BBO setting. In [37] the influence of using Equation (2.6) compared to a binary constraint value is explored, and no significant difference is found. However, they note that the feasible region of the used benchmark problems may just have been too large to observe significant differences.

For equality constraints, both a dynamic and adaptive setting for the parameter δ have been proposed in [27]. Starting from a permissive initial threshold, usually based on the constraint violation of the initial population, the value of δ is decreased during optimization to the target threshold. The dynamic approach decreases the threshold based on the remaining computational budget, and [81] shows that by using such an approach, thresholds as low as 10^{-15} can be reached while static settings for δ fail around 10^{-7} for the considered problems. While [32] also reports performance gains using a scheme with exponential decrease, the fragility of the parameter setting is stressed, and a static value is used. While these are relevant considerations for constrained optimization with equality constraints in general, the focus of this thesis lies on search near constraint boundaries and therefore a static δ set to the target threshold of the respective benchmark problem is used.

2.3.3 Penalty Functions

Penalty functions are some of the earliest CHTs used for EAs based on mathematical programming approaches [13, 43], where a penalty term is added to the fitness function and this new fitness function is optimized. Since this turns the constrained problem into an unconstrained problem, this technique allows the use of an unconstrained optimization algorithm for constrained problems. However, it is important that this new fitness landscape mirrors the constrained problem and that the optima stay the same. Otherwise, the solutions obtained from optimizing this new problem may not correspond to good solutions in the original constrained problem. In literature, various types of penalty functions have been introduced, ranging from static penalty functions, and death penalty, to dynamic and adaptive penalty functions [50, 14]. In the case of a static penalty function, a commonly used form is

$$P_{\alpha,\beta}(\boldsymbol{x}) = f(\boldsymbol{x}) + \alpha \sum_{i=1}^{p+q} \max\{0, \Psi_i(\boldsymbol{x})\}^{\beta} \tag{2.7}$$

where α and β are parameters controlling the strength of the applied penalty. Naturally, these parameters are problem dependent and need to be set correctly. If the penalty is set too low, then there may be an infeasible point with better fitness than the constrained optimum according to the penalized objective function. On the other hand, if the penalty is too high such that infeasible solutions are effectively always discarded, this is called the death penalty and has been found ineffective [13, 43].

Dynamic penalties increase the penalty as evolution progresses, but this also needs the setting of problem specific parameters. Furthermore, increasing the penalty factor too slowly may lead to the optimization converging in the infeasible region [14, 50].

Adaptive penalty functions adjust the penalty term based on the constraint values of the current population. These generally also need parameters that control the adaptation, which again are problem dependent [50].

In [43] it is reported that most uses of penalty functions in recent literature are of the adaptive type, as the other types require time-consuming parameter tuning as the optimal parameter values cannot be known before optimization in a BBO setting.

Augmented Lagrangian One recent adaptive penalty function is applying the Augmented Lagrangian method[26] to EAs [20, 19].The Lagrangian method works by incorporating the constraints into the objective function through the use of so-called Lagrangian multipliers. However, the convergence speed of this approach can be slow. To avoid this, additional penalty terms are added that allow for penalizing constraint violations more aggressively. This allows for a back-and-forth process where infeasible solutions first get overly penalized through the penalty term to pressure search towards the feasible region. The penalty term for feasible solutions is decreased and over time, the Lagrangian multipliers are adapted towards the value of the penalty weights. With suitable update rules, the Lagrangian multipliers converge towards the "correct" penalty to make the unconstrained optimum of the modified fitness function feasible, while the penalty term vanishes [26].For a single constraint, this penalty function is defined as follows

$$H_{\gamma,\omega}(\boldsymbol{x}) = f(\boldsymbol{x}) + \begin{cases} \gamma v(\boldsymbol{x}) + \frac{\omega}{2}(v(\boldsymbol{x}))^2 & \text{if } v(\boldsymbol{x}) \geq \frac{-\gamma}{\omega} \\ -\frac{\gamma^2}{2\omega} & \text{otherwise} \end{cases} \tag{2.8}$$

where $\gamma \in \mathbb{R}^+$ corresponds to the Lagrangian multiplier and $\omega \in \mathbb{R}^+$ to the penalty weight, both generally positive and initialized as $\gamma = 0$ and $\omega = 1$ [20]. After every generation, the parameters are updated as per Algorithm 3. Using the first setting proposed in [19], the update is controlled by the parameters $d_\gamma = 5$, $\chi = 2^{1/5l}$, $k_1 = 3$ and $k_2 = 5$.

Algorithm 3: The update rule for the Augmented Lagrangian method, where $\hat{\boldsymbol{\mu}}$ corresponds to the selection mean and t to the current generation.

procedure *UpdateAL*:

  if $t > 0$ **and** $v(\hat{\boldsymbol{\mu}}_t) > -\frac{\gamma}{\omega}$ **then**

    $\gamma \leftarrow \max\left[0, \gamma + \frac{\omega}{d_\gamma} v(\hat{\boldsymbol{\mu}}_t)\right]$

$$\omega \leftarrow \begin{cases} \omega\chi^{1/4} & \textbf{if } \omega\left(v(\hat{\boldsymbol{\mu}}_t)\right)^2 < k_1 \frac{\left|H_{\gamma,\omega}(\hat{\boldsymbol{\mu}}_t) - H_{\gamma,\omega}(\hat{\boldsymbol{\mu}}_{t-1})\right|}{l} \\ & \textbf{or} \\ & k_2\left|v(\hat{\boldsymbol{\mu}}_t) - v(\hat{\boldsymbol{\mu}}_{t-1})\right| < \left|v(\hat{\boldsymbol{\mu}}_{t-1})\right| \\ \omega\chi^{-1} & \textbf{otherwise} \end{cases}$$

2.3.4 Separation of Objective and Constraints

When doing constrained optimization, the goal is to find high-quality feasible solutions. This means that given a feasible solution and an infeasible solution with a better objective value, the feasible solution is preferred. This preference can be directly used to steer the search towards feasible solutions. To do so, selection is modified to incorporate this preference of feasibility over the objective. Generally, this is done by changing the comparison operator used to determine the "fittest" individuals during selection, such that feasible and infeasible individuals are compared separately. This is the underlying idea of the Constraint Domination Principle (CDP) [17], the ϵ-constrained method[64] and Stochastic Ranking[54].

Constraint Domination Principle This method introduced by Deb [17] directly encodes the desired preference into the following comparison operator used to sort the individuals during selection as follows:

1. Between two feasible individuals, the one with better objective value is preferred.
2. Between a feasible and an infeasible individual, the feasible one is preferred.
3. Between two infeasible individuals, the one with lower constraint value is preferred.

Formally, this is stated in Equation (2.9). Note that this relation is irreflexive ($\boldsymbol{x}_1 \nprec \boldsymbol{x}_1$), asymmetric ($\boldsymbol{x}_1 \prec \boldsymbol{x}_2 \implies \boldsymbol{x}_2 \nprec \boldsymbol{x}_1$) and transitive ($\boldsymbol{x}_1 \prec \boldsymbol{x}_2 \wedge \boldsymbol{x}_2 \prec \boldsymbol{x}_3 \implies \boldsymbol{x}_1 \prec \boldsymbol{x}_3$), i.e. a strict total order. Hence, any comparison based sorting algorithm can be used.

$$\boldsymbol{x}_1 \prec \boldsymbol{x}_2 \iff \begin{cases} f(\boldsymbol{x}_1) < f(\boldsymbol{x}_2) & v(\boldsymbol{x}_1), v(\boldsymbol{x}_1) \leq 0 \\ v(\boldsymbol{x}_1) < v(\boldsymbol{x}_1) & \text{otherwise} \end{cases} \tag{2.9}$$

This method is a direct encoding of the optimization goal, parameter free, not making assumptions about a particular problem, and hence generally applicable.

ϵ-Constrained Method While desired at the end of the optimization, throughout the evolution it may not be ideal to always prefer feasible solutions. For instance, recombination between feasible and slightly infeasible solutions with good objective fitness may lead to better feasible solutions faster [78]. To this end, the ϵ-constrained method loosens the notion of what is considered to be feasible when compared to CDP. Solutions that violate the constraints by up to a threshold ϵ are still considered feasible. Initially this ϵ is set to a large value, often based on the constraint violations found in the initial population, and then shrinks during the optimization, either dynamically or adaptively. As introduced by [64], the comparison operator used is as follows:

$$\boldsymbol{x}_1 \prec_\epsilon \boldsymbol{x}_2 \iff \begin{cases} f(\boldsymbol{x}_1) < f(\boldsymbol{x}_2) & v(\boldsymbol{x}_1), v(\boldsymbol{x}_2) \leq \epsilon \\ f(\boldsymbol{x}_1) < f(\boldsymbol{x}_2) & v(\boldsymbol{x}_1) = v(\boldsymbol{x}_2) \\ v(\boldsymbol{x}_1) < v(\boldsymbol{x}_1) & \text{otherwise} \end{cases} \tag{2.10}$$

For setting the ϵ value throughout the optimization, various different update rules were proposed in literature [64, 62, 61, 79, 60, 22]. Often these are dependent on the remaining computational budget, where either a maximum number of evaluations or a time limit is assumed. The percentage of the budget used can then be defined as

$$b_{used} = \max\left\{\frac{\#\text{evaluations performed}}{\text{evaluation budget}}, \frac{\text{time elapsed}}{\text{time budget}}\right\} \tag{2.11}$$

The update rule for ϵ introduced in [61] uses

$$\begin{aligned} \epsilon_0 &= v(\boldsymbol{x}_\theta) \\ \epsilon_t &= \begin{cases} \epsilon_0 \cdot \left(1 - \frac{b_{used}}{b_{control}}\right)^{cp} & b_{used} < b_{control} \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{2.12}$$

where $v(\boldsymbol{x}_\theta)$ corresponds to the constraint violation of the θ-th individual when sorted by constraint violation.

Recommended settings for the parameters are $\theta = 0.2n$, $b_{control} \in [0.1, 0.8]$ and $cp \in [2, 10]$ [61, 62]. The ϵ-constrained method is frequently used in literature [43, 50, 14] and various dynamic and adaptive update rules have been proposed [60, 79, 10, 22]. In [22] modifications are proposed to increase performance. When starting in the feasible region without a single infeasible solution, the original update rule would set $\epsilon_0 = 0$ and effectively becomes CDP. In addition to deferring the initialization of ϵ_0 until an infeasible solution is found, the proposed update rule allows increasing the ϵ threshold if the population is above a user-defined feasibility threshold to improve search close to constraint boundaries:

$$\begin{aligned} \epsilon_0 &= \begin{cases} v(\boldsymbol{x}_{\theta_i}) & v(\boldsymbol{x}_{\theta_i}) > 0 \\ \infty & \text{otherwise} \end{cases} \\ \epsilon_t &= \begin{cases} \epsilon_0 \cdot \left(1 - \frac{b_{used}}{b_{control}}\right)^{cp} & b_{used} < b_{control} \text{ and } r_f < \alpha \\ (1+\beta) \cdot v_{max} & b_{used} < b_{control} \text{ and } r_f \geq \alpha \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{2.13}$$

where θ_i corresponds to the θ-th infeasible individual when sorted by constraint violation, r_f to the ratio of feasible solutions and v_{max} to the biggest constraint value in the current population. If there are no infeasible solutions at the start of the optimization, the comparison $\prec_\epsilon$ effectively ignores constraints and ϵ_0 is set during the first generation where infeasible solutions are encountered. The authors recommend $\alpha = 0.8 \in [0, 1]$ and $\beta = 0.1 \in [0, 1]$ for the additional parameters[22]. With the original formulation in [61], however, ϵ_0 would be set to 0 if the initial population is fully feasible and the CHT effectively becomes CDP.

Stochastic Ranking Another way of loosening the strict CDP was introduced in [54]. The proposed method uses randomness for the trade-off between feasibility and objective fitness. To allow high quality infeasible solutions to be selected, the introduced comparison stochastically compares either using CDP or using the objective value:

$$\boldsymbol{x}_1 \prec_{P_f} \boldsymbol{x}_2 \iff \begin{cases} f(\boldsymbol{x}_1) < f(\boldsymbol{x}_2) & v(\boldsymbol{x}_1), v(\boldsymbol{x}_1) \leq 0 \text{ or } \mathcal{U}(0,1) \leq P_f \\ v(\boldsymbol{x}_1) < v(\boldsymbol{x}_1) & \text{otherwise} \end{cases} \tag{2.14}$$

where P_f is a parameter corresponding to the probability of considering the solutions as feasible. The recommended setting is $P_f \in [0.4, 0.5]$, such that search is biased towards feasibility. Since the comparison operator is stochastic, efficient sorting algorithms that rely on transitive comparison operators cannot be used. Thus, starting from a random order, bubble-sort is used in order to determine the best individuals during selection. While this CHT only requires one additional parameter, one known weakness of the method is that due to the stochastic nature of selection, it is sometimes possible that the feasible region is not found [50].

2.3.5 Repair Operators

Repair methods perform unconstrained optimization while modifying infeasible solutions to make them feasible. This approach to CHTs is predominantly found in combinatorial optimization [55, 45]. For example, if the search space consists of permutations, invalid genomes can be repaired to ensure that the solution is within the search space. Nonetheless, there have been several adaptions to real-valued settings [58, 57, 36]. These approaches generally need feasible solutions, and the repair is performed by performing a line search between the infeasible solution and the closest feasible solution [36]. It also is possible to simply resample infeasible solutions, however, that does not guarantee a successful repair [58]. In [57], the performance between linear and binary search between the infeasible solution and feasible donor have been investigated, leading to the mixed conclusion that the performance of the different strategies was problem dependent and no strategy outperformed other strategies significantly.

2.3.6 Segregational Methods

When optimizing COPs, balancing feasibility and the objective value is necessary. Hence, a COP can be viewed as a multi-objective problem where the two objectives feasibility and objective fitness should be optimized at the same time. These two objectives can be conflicting, motivating the use of approaches that handle feasible and infeasible solutions separately. In contrast to the feasibility oriented approaches that separate objective and constraints, the approaches in this category typically consider and evolve the feasible and infeasible solutions separately.

Methods handling constraints in this manner were researched two decades ago [44, 51, 27] and have recently found new attention in literature [25, 46, 70, 47, 76, 31]. These methods often employ either multi-objective concepts to keep promising infeasible solutions in the population or co-evolution, where feasible and infeasible individuals constitute two populations that evolve in parallel and exchange information.

The authors of [27] propose using an approach that first selects a user defined fraction of the population using feasible solutions only. The remaining slots are filled by ranking based on a penalized fitness value.

In [44], there is a low chance for promising individuals to survive to the next generation.

A multi-objective inspired approach is presented in [51]. The next population is selected separately from the feasible and infeasible solutions, where the overall target feasibility of the population is a user defined parameter. Infeasible solutions are ranked by both objective fitness and constraint violation using non-dominated sorting. The authors show better performance when compared to the multi-objective algorithm NSGA-II.

Recently, dual-population approaches have received increased attention. The commonality of these approaches is that one population is oriented towards prioritizing feasible individuals, while the second population is more permissive w.r.t. the constraints and there is some form of information exchange.

In [25], the individuals are split into populations based on the number of feasible individuals. If there are enough feasible and infeasible individuals, those are treated as separate populations. The feasible solutions optimize towards better objective values ignoring the constraints, while the infeasible solutions prioritize feasibility. Otherwise, one joint population optimizes towards feasibility if there are too many infeasible solutions or better objective values in case of too many feasible individuals.

Another approach proposed in [46], where only selection is performed with two populations. One population is geared towards feasible solutions, and the other one uses an adaptive penalty function to allow keeping infeasible solutions with a good objective value. Variation is performed jointly, recombining solutions from both populations.

A different strategy for constrained evolutionary optimization is presented in [31] where two populations serve as archive of the best feasible and least constrained infeasible solutions respectively. Recombination then is performed only for the feasible solutions, where nearby solutions from both archives are used for recombination, and offspring solutions can replace solutions in both archives.

In [47], [70] and [76], two populations evolve separately, but selection includes the offspring of the other population. Furthermore, both [47] and [70] make use of a two-stage scheme. The first phase of [47] is exploration, where one population is feasibility oriented while the second population only considers the objective value. Then, the following exploitation stage aims to guide the infeasible population back to the feasible population in order to obtain high quality feasible solutions. In [70] on the other hand, only the feasibility oriented population is used once it starts to converge.

2.3.7 Other Methods

The research on evolutionary constraint handling is active and very diverse approaches have been proposed that do not fit into the aforementioned types of approaches.

Ensemble Methods For different problems, different CHTs and parameter settings may be required. In [52], an approach for constraint handling using a segregated selection is introduced. To avoid fragile parameter settings, the same CHT is used with different parameters, such that each method is used to select half of the offspring. Similarly, to avoid the problem of finding a suitable CHT, [41] proposes an ensemble method utilizing multiple CHTs at once. Four populations with different constraint handling techniques are used, such that all populations select from the shared offspring.

Hybrid Approaches One common theme in recent literature on CHTs is that aspects of different CHTs are combined. For instance, equality constraints are often handled using an ϵ-constrained approach while a different CHT such as a dual-population scheme is used for the overall constraint violation [25, 27].

Multi-Stage Methods Another approach that combines different CHTs are multi-stage approaches. Often the focus is on the tradeoff between exploring the infeasible regions and exploiting slightly infeasible solutions to find better feasible solutions [47, 70]. An approach that first spends evaluations to determine the feasible regions for each constraint is used in [24], followed by a CHT that prioritizes constraints with small feasible regions. In [72], the first stage aims to learn the correlation between constraints and objective value to then exploit this information to guide the trade-off between feasibility and objective value. Other strategies perform optimization using different selection strategies and objectives depending on the number of feasible individuals in the population [71, 25].

Multi-Objective Approaches In addition to the introduced approaches that use multi-objective and co-evolutionary concepts, handling constraints in a multi-objective manner is subject to research [56, 74]. In addition, many recent proposed CHTs are designed for multi-objective settings [51, 46, 70, 76, 47, 31].

Surrogate Models For problems where fitness evaluations are expensive, recent approaches make use of learned surrogate models of the objective and constraint landscape. This can greatly reduce the number of fitness evaluations needed at the computation expense of learning a model. For COPs, this is often combined with other CHTs, and thus primarily an approach to tackle expensive real-world problems rather than an approach to constraint handling on its own [63, 80, 53].

Chapter 3

Existing Constraint Handling in RV-GOMEA

In this chapter, several well-known CHTs from literature are implemented to be used with RV-GOMEA. First, the chosen techniques are introduced. However, some approaches cannot be applied directly to RV-GOMEA as these methods were proposed using different EAs. Where necessary, the CHTs are adapted, and the modifications made are documented. Experiments are performed on the CEC2006[38] and bbob-constrained[29] to determine and compare the performance of the different CHTs when used with RV-GOMEA. Furthermore, as a measure of evolution efficiency, the likelihood of sampling the constrained optimum throughout evolution is analyzed for both constrained and unconstrained problems.

There are numerous literature surveys on constraint handling for EAs [43, 14, 50] documenting the different ideas explored, but they do not provide concrete recommendations regarding which approach to use for a given problem and how to set the parameters [3]. In literature, benchmark problem suites such as CEC2006[38] are commonly used to determine the effectiveness and compare different methods. However, new techniques are often proposed together with minor tweaks of the underlying optimizer, or using different EAs altogether. These changes add noise, making it hard to tell if good benchmark results are due to well-performing CHTs or due to changes in the underlying EA. In addition, for CHTs that are not transforming the fitness function, e.g. through penalty functions, it is often not clear if they can be applied to other EAs and what performance can be expected.

Because of this, in this thesis, CHTs of each type introduced in Section 2.3 are adapted to RV-GOMEA in order to allow more straightforward comparison of the different types of approaches, their advantages, and limitations. The following techniques are used:

- *CDP*[17]. This is the currently used default CHT in RV-GOMEA and thus presents a natural baseline to compare other techniques against.
- *ϵ-Constrained Method*[22, 61]. This approach is well-known and often used in literature[14, 43]. As several versions have been proposed, the original method[61] and a more recent formulation [22] are used.
- *Stochastic Ranking*[54]. This is another well-known technique with competitive per-

formance and has only one additional parameter[14].

- *Augmented Lagrangian Penalty*[19]. This technique is included as representation for the penalty function approaches and has shown promising results when used with CMA-ES[19].
- *Repair Methods*[58, 57]. Repair methods are often combined with EAs using Gaussian distributions[58, 35, 57]. In this thesis, repair using binary search between a feasible parent and an infeasible sample was implemented. This always leads to feasible solutions if enough iterations of binary search are performed. Furthermore, repair by resampling infeasible solutions has been implemented.
- *Dual-Population Method.* These approaches have shown promising performance in recent literature [25, 31, 46, 70, 76]. However, in literature different EAs are used and due to the information exchange between populations, the constraint handling is often tightly coupled to with the optimizer used. Thus, two different dual-population approaches have been adapted for RV-GOMEA, resulting in approaches similar to [25] and [46].

Note that this does not include techniques from Section 2.3.7 as those are either combinations of individual CHTs or designed for more specific optimization settings. Moreover, the field is very active and an exhaustive comparison of all established and recent methods would go beyond the scope of this thesis.

3.1 Adapting Existing Techniques and Parameter Settings

Many techniques such as penalty functions do not depend on the optimization algorithm used and can be implemented by changing the comparison operator used throughout optimization. However, other techniques such as dual-population schemes are tightly coupled to the optimizer used and thus need to be adapted for RV-GOMEA. In this section, the necessary changes for the selected CHTs are described. The parameter tuning performed, and the best parameter settings found are reported in Appendix A.

3.1.1 Constraint Domination Principle

As the CDP method is currently used in RV-GOMEA for constrained problems, no modification is necessary. The current implementation uses this parameter free comparison operator instead of preferring better objective fitness only as per Algorithm 2.

3.1.2 Stochastic Ranking

Similar to CDP, stochastic ranking can be implemented by changing the comparison operator in Algorithm 2. However, as described in Section 2.3.4, due to the stochastic comparison operator bubble sort is used for selection. Additionally, commonly selection both steers where search is performed and which solutions survive, whereas in RV-GOMEA selection is only for where search is performed and GOM decides if a particular GOM sample is accepted. Because of this difference, two versions of stochastic ranking are tested, one where only selection uses the stochastic operator and one where it also is used for GOM acceptance.

Not using the stochastic operator leads to a decline in the number of infeasible solutions in the population, as every feasible GOM sample constitutes an improvement according to CDP, no matter the objective value. Once there are no more infeasible solutions in the population, the method effectively becomes CDP. On the other hand, making GOM stochastic has a higher chance of high quality infeasible solutions surviving, but there also is a chance that good feasible solutions are lost by accepting infeasible changes. This goes against the working principles of GOM, as it is possible for a new infeasible solution to replace a feasible solution that has accumulated several improvements already.

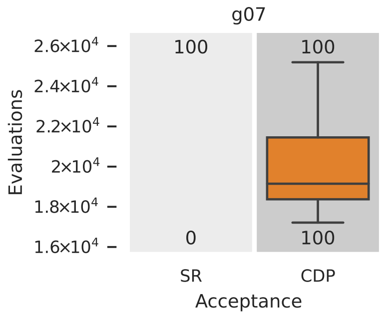


Figure 3.1: The effect of stochastic acceptance during GOM on problem g07 from the CEC2006[38] benchmark problems over 25 runs.

The clear performance difference, visible in Figure 3.1, indicates that also making the GOM acceptance stochastic does not work well with RV-GOMEA. The light gray bar and number at the top of each column indicate the percentage of runs where at least one feasible solution was found. The darker gray bar and number at the bottom correspond to the rate of successfully reaching the approximation target. Clearly, the stochastic acceptance does not perform well. Hence, only the version without will be considered in the upcoming experiments.

3.1.3 ϵ-Constrained Method

For the ϵ-constrained method two variants were implemented, the original version suggested in [62] and an improved method proposed in [22]. The changes necessary to RV-GOMEA are straightforward and as described in literature, instead of CDP as solution comparison operator, the ϵ comparison operator is used. The only additions needed are the initialization and update rules for the CHT. The update is performed after the first generation and before selection is performed.

3.1.4 Augmented Lagrangian Penalty Function

To implement this method, new parameters need to be added and initialized to the described values before the first generation. Then the comparison operator used in RV-GOMEA is adjusted to use Equation (2.8). Furthermore, starting with the second generation, the comparison operator is updated as per Algorithm 3. For the parameter settings, the first setting proposed in [19] is used. Note that the CHT is implemented using the previously introduced constraint violation measure and not individual constraints as per AL-single in [20].

3.1.5 Repair Operators

For the repair of infeasible solutions, two methods from literature are adjusted to RV-GOMEA using binary search and resampling. Both methods are implemented in addition to using CDP. Due to how RV-GOMEA works, this is done during the GOM procedure, between sampling new candidate solutions and accepting improvements.

Binary Search For repairing infeasible solutions, generally an infeasible offspring solution is paired with the closest feasible solution in parameter space. Then linear or binary search is performed in-between the two solutions, always leading to feasible solutions if enough steps are performed [58]. In RV-GOMEA however, the offspring corresponds to a GOM sample tied to a specific parent solution. Hence, the parent solution is used for repair and thus repair is only possible for infeasible GOM samples with feasible parents.

Typically, the closest feasible solution is used [36]. However, as RV-GOMEA can perform partial modifications depending on the used FOS, doing so is not possible. If the violated constraint depends on decision variables that are not in the currently considered FOS subset, different values in two feasible solutions can lead to different constraint boundaries. Consider the example where the violated constraint is $x_0 + x_1 \leq 42$, there are two parents $p_1 = [40, 1, \dots]$ and $p_1 = [35, 3, \dots]$, the current FOS subset is $\mathcal{F}_i = \{1\}$ and a sample $s_{p_1,\mathcal{F}_i} = 3.5$ for parent p_1. The sample with value 3.5 is infeasible, as only values less than 2 ($= 42 - x_{0_{p_1}}$) are feasible w.r.t. the constraint. Then it is entirely possible that p_2 is the closest feasible solution to $s_{p_1,\mathcal{F}_i}$ due to the values of the remaining decision variables. In this case, p_2 cannot be used for repair using binary search, as that would search for x_1 in the range between 3 and 3.5 which contains no feasible solution when combined with p_1. Hence, to ensure that repair is possible, not the closest feasible solution but the feasible parent solution is used to repair infeasible samples.

Furthermore, since GOM generally only accepts improvements, only promising infeasible solutions are repaired to save evaluations. A solution is promising, if it is likely to be selected and thus has a chance of influencing where search is performed in subsequent generations. Feasible solutions with better objective fitness than the best solution before the generation have a high chance of being selected in the following generation. Thus, the objective value of the current elite is used as a threshold infeasible samples need to exceed in order to be repaired. This repair method is described in Algorithm 4 and has a single parameter, the number of binary search steps performed. There is a tradeoff for this parameter, as fewer steps save evaluations, while more steps lead to solutions closer to the constraint boundary.

Resampling A simple way to repair infeasible solutions is to simply resample when a GOM sample turns out to be infeasible. In this case, it is not necessary for the parent to be feasible or the sample to be promising, but the distribution used for sampling needs to have at least some probability density in the feasible region. To ensure this is the case in RV-GOMEA, a new parameter is introduced, setting a minimum threshold of the selection that has to be feasible. A feasible selection generally leads to the estimated distributions having probability density in the feasible region as well. The chance of a successful repair,

Algorithm 4: GOM sample repair using binary search.

procedure *RepairUsingBinarySearch*:
  Input: Parent solution $\boldsymbol{x}$, GOM sample solution $\boldsymbol{y}$, FOS subset $\mathcal{F}_i$, comparison operator $\prec$ and number of repair steps k
  if $f(\boldsymbol{y}) < f(\boldsymbol{x}_{elite})$ **and** $v(\boldsymbol{y}) > 0$ **and** $v(\boldsymbol{x}) = 0$ **then**
    $low, high \leftarrow 0, 1$
    $\vec{\boldsymbol{d}} \leftarrow \boldsymbol{y}[\mathcal{F}_i] - \boldsymbol{x}[\mathcal{F}_i]$
    for $step \in \{1, \ldots, k\}$ **do**
      $mid \leftarrow low + \frac{high - low}{2}$
      $\boldsymbol{z} \leftarrow \boldsymbol{x}[\mathcal{F}_i] + mid \cdot \vec{\boldsymbol{d}}$
      evaluate $\boldsymbol{z}$
      $low, high \leftarrow \begin{cases} mid, high & \textbf{if} \, v(\boldsymbol{z}) = 0 \\ low, mid & \textbf{otherwise} \end{cases}$
      if $\boldsymbol{z} \prec \boldsymbol{y}$ **then**
        $\boldsymbol{y} \leftarrow \boldsymbol{z}$
        if $\boldsymbol{y} \prec \boldsymbol{x}_{elite}$ **then break**

i.e. resampling feasible values, then corresponds to the number of retries and the likelihood of sampling a feasible partial solution. Furthermore, infeasible solutions that would be accepted even without repair are not repaired.

3.1.6 Dual-Population Approaches

For the dual population approach, also two versions are adapted to RV-GOMEA. These need to be adapted, as the previously introduced approaches are specific to the used EA.

"Oscillating" Dual-Population Approach This version corresponds to the method proposed in [47], where the population is split into a feasible and infeasible subpopulation if possible. The feasible population aims to reach better objective fitness, while the infeasible population aims to find feasible solutions. Since both subpopulations aim to cross the constraint boundary, the individuals in this approach "oscillate" between being feasible and infeasible.

However, a few modifications are necessary, since the proposed implementation specific to differential evolution is not compatible with RV-GOMEA. In differential evolution, generally, three solutions other than the parent solution are needed to generate an offspring solution. Because of this, the approach presented in [25] makes it possible to split the population into two subpopulations once at least three feasible and infeasible solutions are in the population. For a given parent, two of the three donor solutions are picked from the same subpopulation, and the last one from the other subpopulation.

However, for RV-GOMEA more solutions are required. Two versions of managing the split into subpopulations are explored. Either, the selection size is kept constant and at least τn feasible and infeasible solutions are needed, or the selection size is adapted such

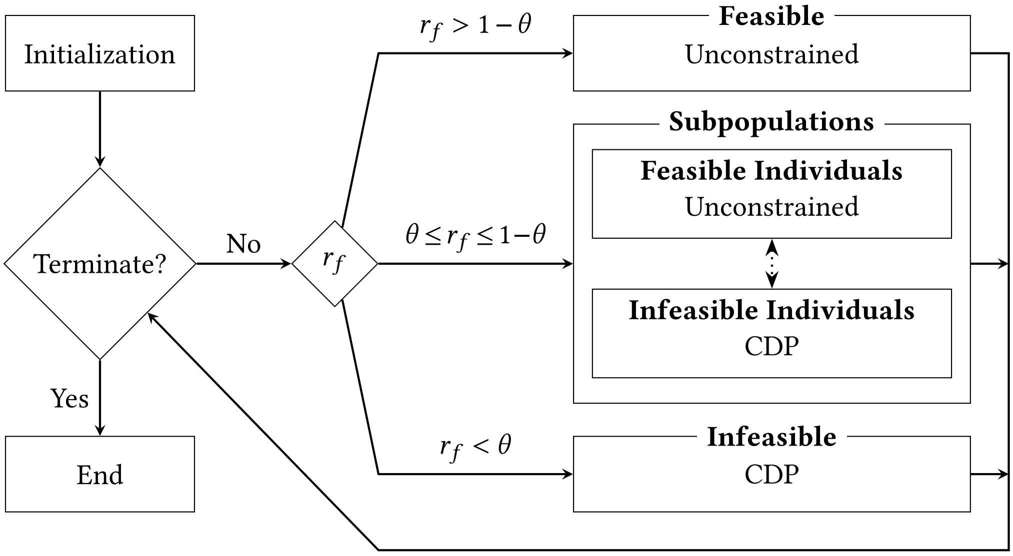


Figure 3.2: The framework used for adapting the DPDE[25] approach to RV-GOMEA where r_f denotes the fraction of feasible solutions in the population. If both the percentage of feasible and infeasible solutions meet the threshold θ, the population is split into subpopulations. The feasible subpopulation optimizes the unconstrained objective value, while the infeasible population utilizes CDP.

that selection is performed proportional to the size of each subpopulation. In the first case, the selection pressure decreases as fewer individuals compete for selection slots. The second case leads to fewer selected solutions, leading to a more focused search due to fewer solutions being used for distribution estimation. However, this potentially leads to an inaccurate estimated distribution if the size of the subpopulation is too small. This is the case in Figure 3.3, where the number of evaluations clearly increases if the proportional scheme is used.

A new parameter $\theta \in [\tau, \frac{1}{2}]$ is introduced as the threshold of feasible and infeasible solutions needed to split the population. If the number of feasible solutions is below θn, then evolution is performed using CDP only to prioritize finding more feasible solutions. Likewise, if there are less than θn infeasible solutions, the constraints are ignored to find more infeasible solutions. Only when both thresholds are satisfied, the population is split into two populations. This is shown in Figure 3.2.

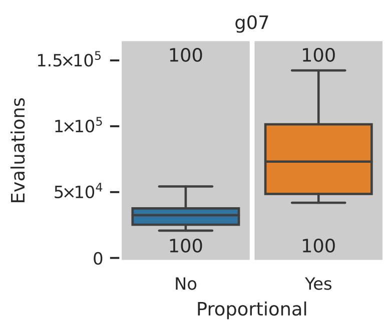


Figure 3.3: The effect of proportional selection on problem g07 from the CEC2006[38] benchmark problems over 25 runs.

Another core aspect of the method proposed in [47] is the information sharing strategy, recombining solutions from both populations when generating the offspring. In RV-GOMEA, the GOM samples come from the estimated distributions. Hence, a possible information sharing strategy is including individuals from the other subpopulation when estimating the distribution. This is determined by a new parameter $\theta_s \in [0, 1]$, controlling the number of solutions added from the other subpopulation in addition to the selection. The value of this parameter is expressed as a percentage of the selection size, i.e. at most half of the solutions used for distribution estimation are drawn from the other population. Additionally, these solutions are not chosen randomly but are drawn from the selection of the other subpopulation.

Dual-Population Approach with Traction Strategy The second dual population approach follows the ideas outlined in the second stage of [46] where the feasible population respects the constraints instead. With the first dual population approach, feasible solutions only consider the objective value and accept infeasible GOM samples that lead to better objective value. This cannot happen with the second dual population approach. Another difference is that two full populations are used. This is the general scheme also followed by [70, 48, 76, 47], however, as the focus of this thesis lies on approximating the boundary from within the feasible region, there are differences in the objectives used for the two populations. These approaches use a penalty function for the less feasible population and a feasibility oriented CHT for the main population, while the implemented version uses CDP for both populations and the solutions in the infeasible population are the solutions that correspond to optimizing the unconstrained problem[1]. Infeasible solutions

[1] This is the case if the Full FOS is used or if the subsets can be optimized fully disjoint, otherwise the infeasible solutions correspond to each improving infeasible GOM sample, not the accumulation of all improving

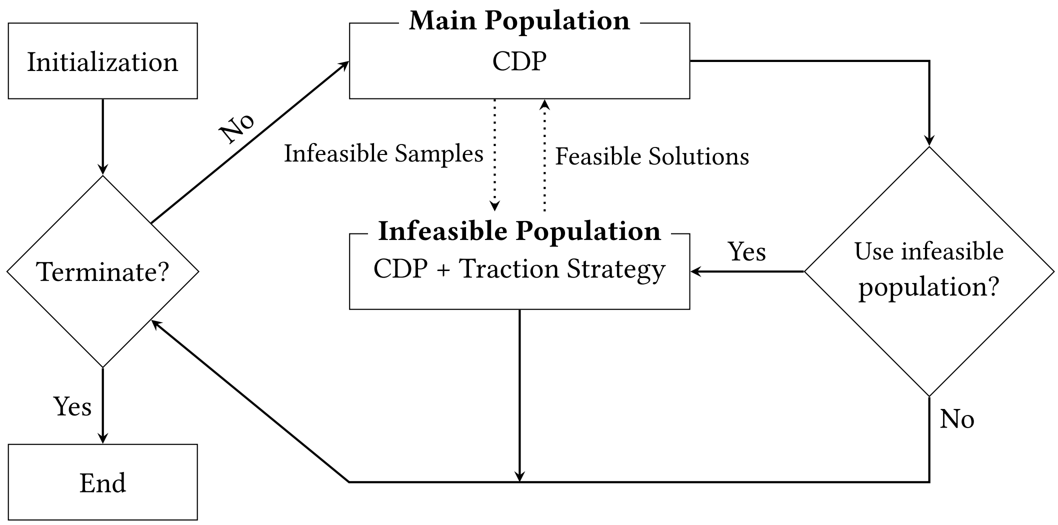


Figure 3.4: The framework used for the second dual-population approach, based on [46]. If the main population has feasible solutions and there are enough infeasible solutions, the infeasible population is used. Using the traction strategy, the infeasible population explores the search space between the itself and the feasible population.

found throughout evolution are added to the infeasible population, which tries to become feasible. The second population only is used if the main population has feasible solutions, as otherwise, both populations would optimize towards the same goal. Because the second population aims to become feasible, there is no need to migrate the GOM samples, and it suffices to migrate the feasible individuals from the infeasible to the main population. Figure 3.4 depicts how the approach works.

Additionally, several traction strategies are explored to tether the two populations together. Since the infeasible population optimizes using CDP, effectively only the constraint violation factor is improved. Individuals that have been in the infeasible population over multiple generations are likely closer to the constraint boundary than newly added solutions, as these have been subject to GOM towards the feasible region already. In selection, this gives an advantage to older solutions. However, the main population is searching close to the newly added infeasible solutions. If the main population is traversing the search space, the focus of the search in the second population will thus lag behind the main population. This decreases the effectiveness of the second population and the information sharing strategy. To prevent this, the selection of the second, infeasible population is adjusted in one of the following ways:

- Rather than selecting based on the constraint violation, the selection is based on the distance to the feasible selection. Individuals close to feasible individuals also tend to have lower constraint violation values.

- A second strategy is to add a few feasible solutions to the selection of the second population, with the idea of increasing the likelihood of exploring between the constraint boundary and the feasible population. In [47], a similar strategy is used.

Figure 3.5 shows the effectiveness of the two traction strategies. The FSI column corresponds to injecting feasible solutions into the infeasible population and shows that there is a clear advantage to ensuring that the infeasible population performs search close to the feasible main population.

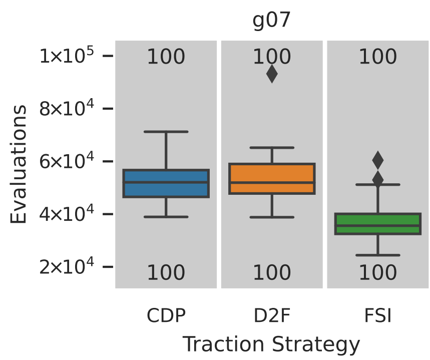


Figure 3.5: The effect of different traction strategies on problem g07 from the CEC2006[38] benchmark problems over 25 runs.

infeasible samples into one solution as normally done by RV-GOMEA.

Chapter 4

Experiments and Results

In the previous chapter, different approaches for constraint handling have been introduced and adapted to RV-GOMEA. In this chapter, experiments are performed in order to find out how these approaches perform and compare. Additionally, the experiments should give an indication of the strengths and weaknesses of the different approaches. First, the experimental setup is explained, followed by the results for the different benchmark suites and problems.

4.1 Benchmark Problems and Setup

In this subsection, the benchmark suites and problems used in the experiments are introduced. First, the CEC2006[38] suite is introduced, followed by the bbob-constrained suite[29]. Furthermore, a new problem constructed to test aspects not covered by these benchmark suites is introduced.

For the sake of brevity, the different CHTs are henceforth abbreviated in the following results as follows: Augmented Lagrangian (AL), Constraint Domination Principle (CDP), ϵ-Constrained (EPS), improved ϵ-Constrained (IEPS), Stochastic Ranking (SR), the two dual-population adaptions using “oscillating” subpopulations and a traction strategy as DP-O and DP-T respectively and the repair methods using binary search and resampling as R-BS and R-R.

4.1.1 CEC2006

This benchmark suite introduced in [38] consists of 24 COPs and provides clear setup instructions to allow comparing results across literature using the benchmark suite. The problems have between 2 and 24 dimensions and a wide range of objective functions, equality, and inequality constraints as well as differently sized feasible regions. These properties are displayed in Table 4.1. The recommended experiment setup allocates an evaluation budget of 500 000 evaluations, with 25 runs per problem. For the equality constraints, constraint violations lower than the threshold $\epsilon = 10^{-4}$ are considered feasible. For all problems, the objective value of the constrained optimum $\boldsymbol{x}_{opt}$ is known and feasible except for problem 20 which will not be considered in this thesis for this reason. A run

Problem	l	Type of function	% Feasible	LI	NI	LE	NE	A
g01	13	quadratic	0.0111%	9	0	0	0	6
g02	20	nonlinear	99.9971%	0	2	0	0	1
g03	10	polynomial	0.0000%	0	0	0	1	1
g04	5	quadratic	52.1230%	0	6	0	0	2
g05	4	cubic	0.0000%	2	0	0	3	3
g06	2	cubic	0.0066%	0	2	0	0	2
g07	10	quadratic	0.0003%	3	5	0	0	6
g08	2	nonlinear	0.8560%	0	2	0	0	0
g09	7	polynomial	0.5121%	0	4	0	0	2
g10	8	linear	0.0010%	3	3	0	0	6
g11	2	quadratic	0.0000%	0	0	0	1	1
g12	3	quadratic	4.7713%	0	1	0	0	0
g13	5	nonlinear	0.0000%	0	0	0	3	3
g14	10	nonlinear	0.0000%	0	0	3	0	3
g15	3	quadratic	0.0000%	0	0	1	1	2
g16	5	nonlinear	0.0204%	4	34	0	0	4
g17	6	nonlinear	0.0000%	0	0	0	4	4
g18	9	quadratic	0.0000%	0	13	0	0	6
g19	15	nonlinear	33.4761%	0	5	0	0	0
g20	24	linear	0.0000%	0	6	2	12	16
g21	7	linear	0.0000%	0	1	0	5	6
g22	22	linear	0.0000%	0	1	8	11	19
g23	9	linear	0.0000%	0	2	3	1	6
g24	2	linear	79.6556%	0	2	0	0	2

Table 4.1: The properties of the 24 problems in the CEC benchmark suite. l corresponds to the problem dimensionality, % Feasible denotes the estimated ratio of feasible solutions in the search space ($|\mathcal{M}|/|\Omega|$), LI/LE corresponds to the number of linear inequalities/equalities, NI/NE to the number of non-linear inequalities/equalities and A to the number of constraints active at the constrained optimum.

is considered feasible if a feasible solution is found, and successful if a feasible solution $\boldsymbol{x}$ with $f(\boldsymbol{x}) - f(\boldsymbol{x}_{opt}) \leq 10^{-4}$ is found. Reported are the best, median, and worst result and the mean objective value and the standard deviation over the 25 repetitions. Additionally, the feasible rate, success rate, and success performance are reported for each problem. The feasible and success rates correspond to the number of feasible or successful runs divided by the number of repetitions (25). The success performance is defined in [38] as

$$\text{Success Performance} := \frac{\text{mean \# of evaluations until success} \times \text{\# of total runs}}{\text{\# of successful runs}} \tag{4.1}$$

By following these instructions outlined in [38], the results obtained by RV-GOMEA can be used to compare with results from literature and hopefully give an indication of how well the different CHTs work on different optimizers. Notably, CEC2006 is the only benchmark suite used in this thesis where the initialization is performed fully randomly and a substantial aspect of solving the problems is finding the feasible region in the first place. For the other benchmark suites, the focus lies on approximating the constraint boundary when starting in the feasible region.

4.1.2 bbob-constrained

This benchmark suite is part of the COCO platform[29] and contains 54 problems in dimensions $l \in \{2, 3, 5, 10, 20, 40\}$ constructed using originally unconstrained functions with added inequality constraints. This suite varies the number of constraints, also proportional to the dimensionality. Additionally, non-linear transformations and rotations are applied. In Table 4.2, the number of constraints, objective function, and transformation of the 54 problems are described, and the unconstrained base functions are listed in Table 4.3. For these problems, an initial solution in the feasible region is provided. The problem description includes an instruction on generating a feasible initial population by repeatedly sampling from a Gaussian distribution with the first feasible solution as mean. Hence, this benchmark suite tests the capability of an algorithm to traverse a single feasible region with the constrained optimum located at the constraint boundary. By construction, there is always an active constraint. To assess algorithm performance, the COCO platform can be used[29]. The number of function evaluations is recorded for multiple target values on 15 instances of the same problem with different problem parameters. This data is then used to compute the average runtimes and the empirical distribution function of runtimes (ECDF), aggregated over target values, subclasses of problems, or all problems[28]. In this thesis, the dimensions $l \in \{2, 5, 10, 20\}$ are used. The computational budget was set to $10^6 l$

Number of constraints			1	3	9	$9 + \lfloor 3l/4 \rfloor$	$9 + \lfloor 3l/2 \rfloor$	$9 + \lfloor 9l/2 \rfloor$
Number of active constraints			1	2	6	$6 + \lfloor l/2 \rfloor$	$6 + l$	$6 + 3l$
Objective	T	c_{scal}	Function IDs					
f_{sphere}	-	10	1	2	3	4	5	6
$f_{\text{ellipsoid}}$	T_{osz}	10^{-4}	7	8	9	10	11	12
f_{linear}	-	10	13	14	15	16	17	18
$f_{\text{elli\_rot}}$	T_{osz}	10^{-4}	19	20	21	22	23	24
f_{discus}	T_{osz}	10^{-4}	25	26	27	28	29	30
$f_{\text{bent\_cigar}}$	$T_{\text{asy}}^{0.5}$	10^{-4}	31	32	33	34	35	36
$f_{\text{diff\_powers}}$	-	10^{-2}	37	38	39	40	41	42
$f_{\text{rastrigin}}$	$T_{\text{asy}}^{0.2} \circ T_{\text{osz}}$	10	43	44	45	46	47	48
$f_{\text{rast\_rot}}$	$T_{\text{asy}}^{0.2} \circ T_{\text{osz}}$	10	49	50	51	52	53	54

Table 4.2: The identifiers, objective functions, number of constraints and transformations for the 54 problems in the bbob-constrained suite. T corresponds to a non-linear transformation of the search space and "-" means no transformation. Again, l is the number of decision variables.

Function	Formulation	Transformations
Sphere	$f_{\text{sphere}}(x) = z^\top z + f_{\text{uopt}}$	$z = x - x^{\text{uopt}}$
Separable ellipsoid	$f_{\text{ellipsoid}}(x) = \sum_{i=1}^{l} 10^{6\frac{i-1}{l-1}} z_i^2 + f_{\text{uopt}}$	$z = x - x^{\text{uopt}}$
Linear slope	$f_{\text{linear}}(x) = \sum_{i=1}^{l} 5\,\lvert s_i\rvert - s_i z_i + f_{\text{uopt}}$	$s_i = \text{sign}\left(x_i^{\text{uopt}}\right) 10^{\frac{i-1}{l-1}}$ $z_i = \begin{cases} x_i & \text{if } x_i^{\text{uopt}} x_i < 5^2 \\ x_i^{\text{uopt}} & \text{otherwise} \end{cases}$ for $i = 1, \ldots, l$
Rotated ellipsoid	$f_{\text{elli\_rot}}(x) = \sum_{i=1}^{l} 10^{6\frac{i-1}{l-1}} z_i^2 + f_{\text{uopt}}$	$z = R\left(x - x^{\text{uopt}}\right)$
Discus	$f_{\text{discus}}(x) = 10^6 z_1^2 + \sum_{i=2}^{l} z_i^2 + f_{\text{uopt}}$	$z = R\left(x - x^{\text{uopt}}\right)$
Bent cigar	$f_{\text{bent\_cigar}}(x) = z_1^2 + 10^6 \sum_{i=2}^{l} z_i^2 + f_{\text{uopt}}$	$z = R\left(x - x^{\text{uopt}}\right)$
Different powers	$f_{\text{diff\_powers}}(x) = \sqrt{10^6 \sum_{i=1}^{l} \lvert z_i\rvert^{2+4\frac{i-1}{n-1}}} + f_{\text{uopt}}$	$z = R\left(x - x^{\text{uopt}}\right)$
Rastrigin	$f_{\text{rastrigin}}(x) = 10\left(n - \sum_{i=1}^{l} \cos(2\pi z_i)\right) + z^\top z + f_{\text{uopt}}$	$z = x - x^{\text{uopt}}$
Rotated Rastrigin	$f_{\text{rast\_rot}}(x) = 10\left(n - \sum_{i=1}^{l} \cos(2\pi z_i)\right) + z^\top z + f_{\text{uopt}}$	$z = R\left(x - x^{\text{uopt}}\right)$

Table 4.3: The objective functions from the COCO platform[29] used by bbob-constrained. The vector $\boldsymbol{x}^{\text{uopt}} \in \mathbb{R}^l$ is randomly sampled and determines the unconstrained optimum of the function with $f_{\text{uopt}}) := f(\boldsymbol{x}^{\text{uopt}})$. The rotations $R \in \mathbb{R}^{l \times l}$ were randomly sampled and are fixed for each instance. The non-linear transformations mentioned in Table 4.2 are applied after rotation.

evaluations with a time limit of 60 seconds. This time limit is restrictive for the larger dimensions, however, it was necessary to ensure timely completion of individual runs.

4.1.3 Cone Problem

There are existing constrained benchmark suites using both constructed and real-world problems, but none of these benchmark suites allow changing the size of the feasible region or decomposing constrained problems in line with the previously introduced GBO setting. Even though the contribution of a single variable to the objective value may be separable, constraints can create linkage and prevent decomposition. To mitigate these shortcomings, a new problem based on existing benchmark problems and a new constraint formulation is introduced, hereafter called the Cone problem. In Figure 4.1, the fitness landscape of different instances is shown. In 2D, the problem corresponds to a single linear constraint or a constrained optimum located at the intersection of two linear constraints, depending on θ.

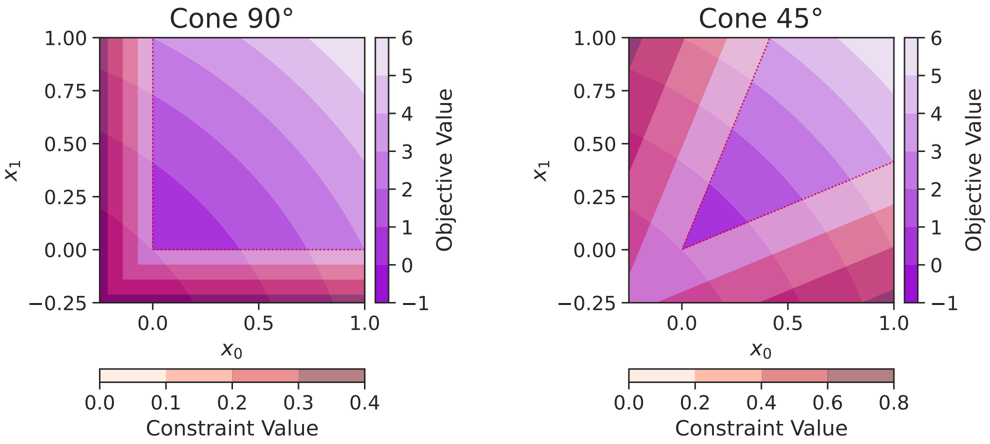


Figure 4.1: Different instances of the Cone problem with feasibility parameter $\theta = 90°$ and $\theta = 45°$ in 2D.

The sphere function $f(\boldsymbol{x}) := x^T x = \sum_{i=1}^{l} x_i^2$ is used for the objective value. The optimum of the unconstrained problem is shifted to $\mathbf{-1}^l$ and the constrained optimum is placed at $\mathbf{0}$ and set to 0 instead:

$$f_{Cone}(\boldsymbol{x}) := \sum_{i=1}^{l} x_i^2 + 2x_i \tag{4.2}$$

The constraint is a l-dimensional extension of a 2D cone parameterized by an angle $\theta \in (0°, 180°]$, corresponding to how large the feasible region should be. The tip of the cone is located at the feasible optimum, i.e. $\mathbf{0}^l$, and the feasible region grows away from the direction of the unconstrained optimum[1]. The constraint value is defined as a single linear inequality and computes the distance from the feasible region in parameter space:

$$g_{Cone}(\boldsymbol{x}) := \begin{cases} -\boldsymbol{x} & l = 1 \\ -length & \theta = 180° \\ \frac{\boldsymbol{x} - \hat{\mathbf{1}} * length}{\|\boldsymbol{x} - \hat{\mathbf{1}} * length\|} - length \cdot \tan\theta & \text{otherwise} \end{cases} \tag{4.3}$$

where $length := \boldsymbol{x}^T \hat{\mathbf{1}}$ corresponds to the distance away from the tip of the cone and $\hat{\mathbf{1}} := \frac{\{1\}^l}{\|\{1\}^l\|}$. The first case allows having 1-dimensional problems, the second case is necessary because $\tan 180°$ is not defined, and the third case effectively computes the signed distance from the boundary of the cone at the given length from the tip. Within the feasible region, the constraint value will be negative and outside there is a positive constraint value

[1] This construction works using the Sphere function but not necessarily for other functions. The transformation $\tilde{f}(\boldsymbol{x}) = max\{f(\boldsymbol{x}_{opt}), f(\boldsymbol{x})\} + v(\boldsymbol{x})$ would work for any function to ensure that at least one constrained optimum is at $\boldsymbol{x}_{opt}$ regardless of where the unconstrained optimum and the local optima are placed. However, this potentially leads to loss of precision due to the limitations of floating point numbers if the scales of the objective and constraint value near the optimum differ substantially.

indicating the shortest distance to the feasible region. If $\theta = 180°$, then this amounts to a single linear constraint. With increased dimensionality and smaller angles, however, this is a non-linear constraint. Note that through the constraint function, all decision variables are linked, although the objective function is separable. In order to construct a decomposable problem, the Cone problem can be repeated, creating a problem consisting of linked *blocks*, where each block corresponds to a Cone problem instance. Furthermore, while the angle θ determines the size of the feasible region, the relationship between the angle and the ratio of feasible to infeasible solutions is not constant w.r.t. increasing problem size. As the problem dimension grows larger, the ratio of feasible solutions shrinks with constant θ where $\theta < 180°$.

As for the experimental setting, the effect of decreasing the size of the feasible region, increasing the problem dimensionality, and decomposition using a marginal product FOS are explored to identify the strengths and weaknesses of the CHTs in settings not covered by the other experiments used. For $l \in \{2, 10, 20, 40, 80\}$, $\theta \in \{180°, 135°, 90°, 45°, 5°\}$ and block sizes in $\{l, 5, 10\}$, an evaluation budget of $25000l$ and a value to reach of 10^{-10} are used. For block size l, the full FOS is use,d and the marginal product FOS that matches the block size is used for block sizes 5 and 10. Reported is the median number of evaluations and interdecile range of 31 runs, as well as the success rate. The initialization is performed in the feasible region to eliminate differences in the time to reach the feasible region. Thus, resulting differences in evaluations between CHTs are due to differences in behavior when approximating constraint boundaries.

4.2 CEC2006

The detailed results for each CHT on this benchmark suite are included in Appendix B.

The feasible rate for the CEC2006 problems, i.e. the percentage of the 25 runs that produced feasible solutions during optimization, is shown in Table 4.4. Note that for problem g22, no approach managed to find the feasible region. On the other problems, the tested CHTs are able to find the feasible region reliably, except for the Augmented Lagrangian method (AL) and the two ϵ-Constrained methods. These methods trade off optimizing towards better objective values and feasibility, whereas the other methods prefer feasibility until the population is at least partially feasible. Note that Stochastic Ranking (SR) is an exception as the method stochastically selects based on the objective before reaching the feasible region. However, due to accepting using CDP during variation, the population only moves towards the feasible region. For the Augmented Lagrangian method, ill-conditioned problems for the chosen parameter values can cause the optimization to get stuck in the infeasible region. Additionally, in [20] the implementation based on the constraint violation measure has been shown to clearly perform worse on problem g10 compared to handling constraints individually.For the ϵ-Constrained approaches, the reason for not reliably reaching the feasible region generally is that the ϵ threshold does not decrease quickly enough and the search converges in the infeasible region before the search is pressured towards feasibility. Notably, the ϵ-Constrained approaches struggle with different problems than the Augmented Lagrangian method.

	CDP	AL	DP-O	DP-T	EPS	IEPS	R-BS	R-R	SR
g01	**100.0%**	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$
g02	**100.0%**	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$
g03	**100.0%**	0.0%$^{-}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$
g04	**100.0%**	40.0%$^{-}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$
g05	**100.0%**	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$
g06	**100.0%**	96.0%$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	72.0%$^{-}$	88.0%$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$
g07	**100.0%**	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$
g08	**100.0%**	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	88.0%$^{\approx}$	92.0%$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$
g09	**100.0%**	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$
g10	**100.0%**	0.0%$^{-}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$
g11	**100.0%**	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$
g12	**100.0%**	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$
g13	**100.0%**	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$
g14	**100.0%**	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$
g15	**100.0%**	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	68.0%$^{-}$	60.0%$^{-}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	96.0%$^{\approx}$
g16	**100.0%**	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	92.0%$^{\approx}$	88.0%$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$
g17	**100.0%**	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	44.0%$^{-}$	40.0%$^{-}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$
g18	**100.0%**	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$
g19	**100.0%**	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$
g21	**100.0%**	44.0%$^{-}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	0.0%$^{-}$	0.0%$^{-}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$
g22	**0.0%**	**0.0%**$^{\approx}$	**0.0%**$^{\approx}$	**0.0%**$^{\approx}$	**0.0%**$^{\approx}$	**0.0%**$^{\approx}$	**0.0%**$^{\approx}$	**0.0%**$^{\approx}$	**0.0%**$^{\approx}$
g23	**100.0%**	0.0%$^{-}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$
g24	**100.0%**	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$
$+/\approx/-$	-	0/18/5	0/23/0	0/23/0	0/19/4	0/20/3	0/23/0	0/23/0	0/23/0

Table 4.4: The Feasible Rate metric for the CEC2006 problems. The best values achieved for each problem are highlighted in bold. The $+/\approx/-$ symbols indicate significantly better/not different/worse performance when compared to CDP using Fisher's exact test [23] with $p < 0.05$. The bottom row displays the total number of times the CHT was better/not different/worse compared to CDP.

The success rate, i.e. the percentage of the 25 runs that approximated the constrained optimum up to an error of 10^{-4} is shown in Table 4.5. In line with the feasible rate, the same reasons explain the decreased success rate for the AL, EPS, and IEPS approaches. For problem g06 the AL method reliably reaches feasible solutions but never reaches the approximation target. Similarly, the ϵ-Constrained method struggles with reaching the success criterion on problem g17. However, the increased bias towards better objective value leads to significantly better results on problems g12 and g13 compared to CDP. Other than that, the same problems tend to be hard to reliably solve, with no success on problem g22. With respect to the feasibility oriented approaches, the different methods perform similarly. Only SR and DP-O achieve significantly better results on some problems. Note that the recommended population size has been used for the CHTs. As the success rate for some problems and CHT combinations is high, but not 100%, this indicates that a larger population size could lead to an increased success rate.

	CDP	AL	DP-O	DP-T	EPS	IEPS	R-BS	R-R	SR
g01	76.0%	88.0%$^{\approx}$	**92.0%**$^{\approx}$	64.0%$^{\approx}$	72.0%$^{\approx}$	60.0%$^{\approx}$	68.0%$^{\approx}$	84.0%$^{\approx}$	80.0%$^{\approx}$
g02	0.0%	0.0%$^{\approx}$	**20.0%**$^{+}$	0.0%$^{\approx}$	0.0%$^{\approx}$	4.0%$^{\approx}$	8.0%$^{\approx}$	4.0%$^{\approx}$	4.0%$^{\approx}$
g03	**100.0%**	0.0%$^{-}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$
g04	**100.0%**	0.0%$^{-}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$
g05	**100.0%**	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$
g06	**100.0%**	0.0%$^{-}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	60.0%$^{-}$	88.0%$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$
g07	**100.0%**	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$
g08	**100.0%**	84.0%$^{\approx}$	92.0%$^{\approx}$	96.0%$^{\approx}$	48.0%$^{-}$	56.0%$^{-}$	92.0%$^{\approx}$	88.0%$^{\approx}$	96.0%$^{\approx}$
g09	**100.0%**	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$
g10	**100.0%**	0.0%$^{-}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$
g11	**100.0%**	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$
g12	44.0%	**92.0%**$^{+}$	72.0%$^{+}$	56.0%$^{\approx}$	80.0%$^{+}$	88.0%$^{+}$	48.0%$^{\approx}$	68.0%$^{\approx}$	84.0%$^{+}$
g13	32.0%	36.0%$^{\approx}$	32.0%$^{\approx}$	56.0%$^{\approx}$	**72.0%**$^{+}$	64.0%$^{+}$	36.0%$^{\approx}$	28.0%$^{\approx}$	52.0%$^{\approx}$
g14	**100.0%**	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$
g15	**100.0%**	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	68.0%$^{-}$	60.0%$^{-}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	96.0%$^{\approx}$
g16	**100.0%**	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	92.0%$^{\approx}$	88.0%$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$
g17	16.0%	24.0%$^{\approx}$	24.0%$^{\approx}$	36.0%$^{\approx}$	0.0%$^{\approx}$	0.0%$^{\approx}$	28.0%$^{\approx}$	24.0%$^{\approx}$	**48.0%**$^{+}$
g18	84.0%	92.0%$^{\approx}$	84.0%$^{\approx}$	68.0%$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	76.0%$^{\approx}$	92.0%$^{\approx}$	84.0%$^{\approx}$
g19	**100.0%**	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$
g21	**100.0%**	44.0%$^{-}$	76.0%$^{-}$	96.0%$^{\approx}$	0.0%$^{-}$	0.0%$^{-}$	88.0%$^{\approx}$	96.0%$^{\approx}$	96.0%$^{\approx}$
g22	**0.0%**	**0.0%**$^{\approx}$	**0.0%**$^{\approx}$	**0.0%**$^{\approx}$	**0.0%**$^{\approx}$	**0.0%**$^{\approx}$	**0.0%**$^{\approx}$	**0.0%**$^{\approx}$	**0.0%**$^{\approx}$
g23	**100.0%**	0.0%$^{-}$	96.0%$^{\approx}$	**100.0%**$^{\approx}$	76.0%$^{-}$	76.0%$^{-}$	**100.0%**$^{\approx}$	96.0%$^{\approx}$	**100.0%**$^{\approx}$
g24	**100.0%**	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	96.0%$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$
+/≈/−	-	1/16/6	2/20/1	0/23/0	2/16/5	2/17/4	0/23/0	0/23/0	2/21/0

Table 4.5: The Success Rate metric for the CEC2006 problems. The best values achieved for each problem are highlighted in bold. The +/≈/− symbols indicate significantly better/not different/worse performance when compared to CDP using Fisher's exact test [23] with $p < 0.05$. The bottom row displays the total number of times the CHT was better/not different/worse compared to CDP.

The mean number of evaluations needed divided by the success rate is shown in Table 4.6. Lower values are better and indicate reaching the target quicker. Unreliable success increases the value of the metric. Because both the evaluations and the success rate are contained, this table should be considered together with Table 4.5. For instance, the AL method achieves the best success performance on problems g17 and g21, but it is not the most reliable method on either problem. Notably, the AL method achieves the best success performance most often, on 12 different problems.Repair using binary search (R-BS) generally outperforms repair by resampling (R-R) and outperforms CDP on 12 problems, while the method using resampling does so only twice. Both Stochastic Ranking and repair using binary search tend to take fewer evaluations than CDP, while the dual-population methods need more evaluations. The ϵ-Constrained methods generally take more evaluations than the other methods, except for problems g12 and g13.

	CDP	AL	DP-O	DP-T	EPS	IEPS	R-BS	R-R	SR
g01	32534.5	**28845.6**	69020.7	48083.0	69261.8	81869.4	35442.7	33550.9	32305.1
g02	-	-	**557860.0**	-	-	1247600.0	565356.2	1155600.0	1122350.0
g03	48991.6	-	62014.8	76399.6	42305.9	42230.6	48844.8	57152.8	**20043.7**
g04	5423.7	-	11176.2	5886.8	15208.4	46091.2	5108.5	6059.8	**5039.8**
g05	35441.0	**4342.5**	50261.9	53809.5	47685.2	47402.7	31839.2	47221.4	10908.9
g06	2300.1	-	2938.3	3420.3	81187.8	55273.1	2615.2	2815.1	**1839.1**
g07	20423.8	18835.8	36153.3	34849.0	54946.4	55357.8	**17652.1**	27016.0	20200.4
g08	**232.6**	15996.4	701.9	299.5	76251.0	63779.1	260.5	290.4	274.5
g09	7422.0	**5582.9**	12865.4	9711.4	46183.8	45823.2	6976.6	9284.3	7160.1
g10	24388.0	-	41161.4	43585.7	67341.2	67014.8	22266.8	27568.0	**21621.0**
g11	4083.8	**836.3**	4654.7	5992.2	36948.8	35805.1	5017.7	5168.0	1148.2
g12	1633.1	**277.0**	1207.3	1758.0	417.5	387.0	1726.9	1294.4	839.7
g13	280714.8	157735.5	318559.8	202039.3	**58816.0**	65765.9	312176.2	441991.8	70190.5
g14	12976.7	**11155.2**	21133.4	13939.4	47591.5	48605.2	12225.8	14590.1	15784.0
g15	23149.6	**3667.3**	28744.8	26324.1	66322.1	76320.0	21947.8	32188.8	7200.4
g16	4490.0	**4194.8**	5725.7	5817.2	50373.3	53165.8	4349.1	5360.6	4449.8
g17	473603.1	**105107.6**	533517.4	393863.9	-	-	380642.9	603310.4	128697.6
g18	11794.3	**11133.4**	19425.9	21518.6	56121.9	67053.1	12809.0	13908.1	12342.7
g19	42077.3	**37341.5**	104095.6	63982.0	41715.4	55055.8	39519.7	51850.5	42052.8
g21	35539.8	**26563.4**	76180.1	47120.8	-	-	41402.5	49774.4	28786.4
g23	39846.2	-	75848.0	59739.2	74124.1	69606.2	43020.8	49666.3	**37625.1**
g24	836.2	857.2	1305.4	1033.6	881.0	44604.2	852.6	991.7	**787.0**
</>	-	14/7	2/20	2/19	4/17	4/18	12/10	2/20	19/3

Table 4.6: The Success Performance metric for the CEC2006 problems. The best values for each problem are highlighted in bold, and the total number of times a CHT achieved a better (<) or worse (>) Success Performance than CDP is listed in the last row.

Overall, the results show a clear distinction between the AL penalty function, the ϵ-Constrained techniques, and the remaining feasibility oriented methods. The AL penalty performs well in terms of evaluations, however, the method struggles with finding the feasible approximations of the constrained optimum. The ϵ-Constrained methods perform well on problems g12 and g13, but have a low success rate on the other problems and do not reliably find the feasible region. The remaining methods have similar success rates, with differences in the number of evaluations. Stochastic Ranking and DP-O achieve the highest success rates, followed by the current method, CDP. In terms of evaluations, Stochastic Ranking performs best compared to the other feasibility oriented methods. This is likely due to the initial focus on both objective and constraint violation. CDP, the dual-population and repair methods only begin to focus on the objective value after reaching the feasible region. Because of this, these approaches likely need to traverse a larger part of the feasible region compared to Stochastic Ranking.

4.3 bbob-constrained

The result overview from the experiments performed using the bbob-constrained benchmark functions are shown in figures 4.2,4.3,4.4 and 4.5. The results are obtained using the COCO[29] platform and follow the assessment method explained in [28]. The figures show the Empirical Cumulative Distribution Functions (ECDFs) of the number of targets solved over the runtime, expressed as the logarithm of the evaluations divided by the dimension. Note that the number of evaluations shown is the sum of the number of objective function and the constraint function evaluations. The number of targets corresponds to 41 different approximation levels, ranging from $f(\boldsymbol{x}_{opt}) + 100$ to $f(\boldsymbol{x}_{opt}) + 10^{-6}$. If there is a cross on a line, the method was able to solve all approximation targets up to this point in all runs. Additional approximation targets were not reached by at least one run. Furthermore, the fraction of function target pairs reached after exhausting the computational budget is marked with a small dot[2].

For the 2D instances, the results are shown in Figure 4.2. In this dimensionality, the results of most techniques are similar. After around $1000l$ evaluations, there is a big increase in the number of targets reached, and towards the final targets the number of evaluations needed increases drastically. However, there are some notable outliers. For a single constraint, the DP-O method shows a clear advantage in terms of the number of evaluations needed to reach the approximation targets. For the problems with more constraints, SR consistently needs fewer evaluations and reaches a higher fraction of the approximation targets before progress slows down. In line with the CEC2006 results, the ϵ-Constrained methods and the Augmented Lagrangian method reach fewer targets due to less pressure towards feasibility.

The results for the 5D instances, shown in Figure 4.3, display similar results. DP-O again has a clear advantage on problems with a single constraint and performs well on problems with more constraints. There is a clear gap between the performance of the other

[2]Note that due to the time limit, there can be differences in the total number of evaluations performed between CHTs.

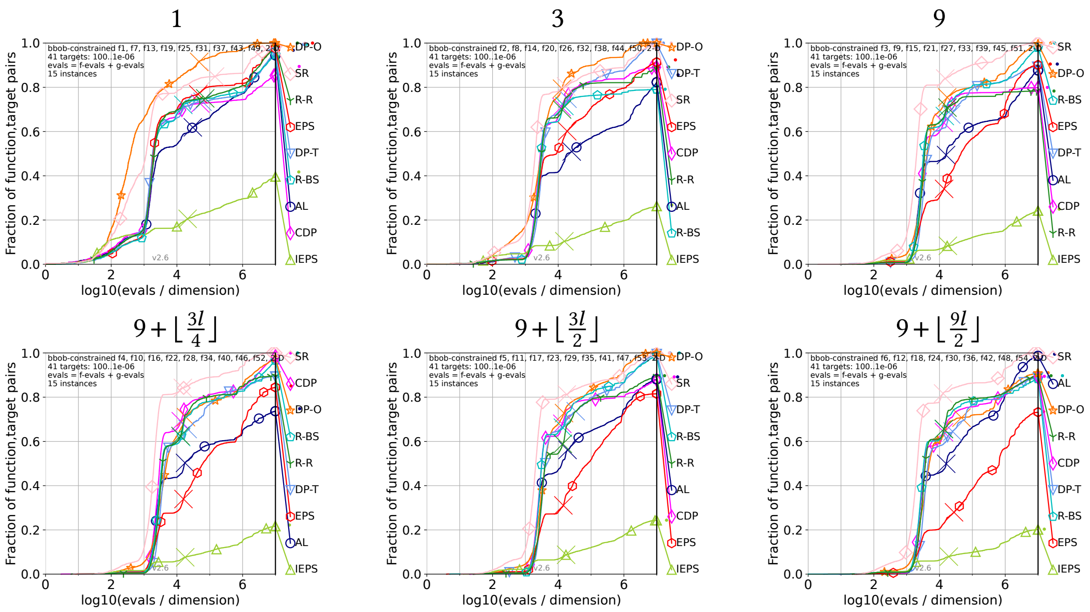


Figure 4.2: The results for the existing methods on the 2D instances of the bbob-constrained suite, grouped by the number of constraints, as described in Table 4.2. The number of total constraints is displayed above the figures.

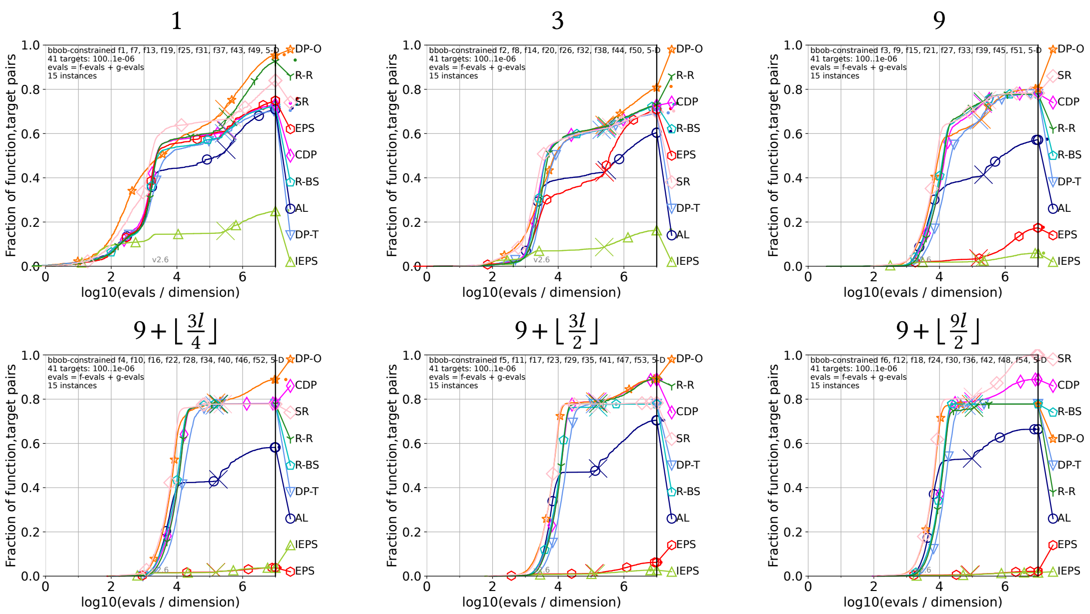


Figure 4.3: The results for the existing methods on the 5D instances of the bbob-constrained suite. The 54 functions are grouped by the number of constraints and active constraints, as described in Table 4.2. The number of total constraints is displayed above the figures.

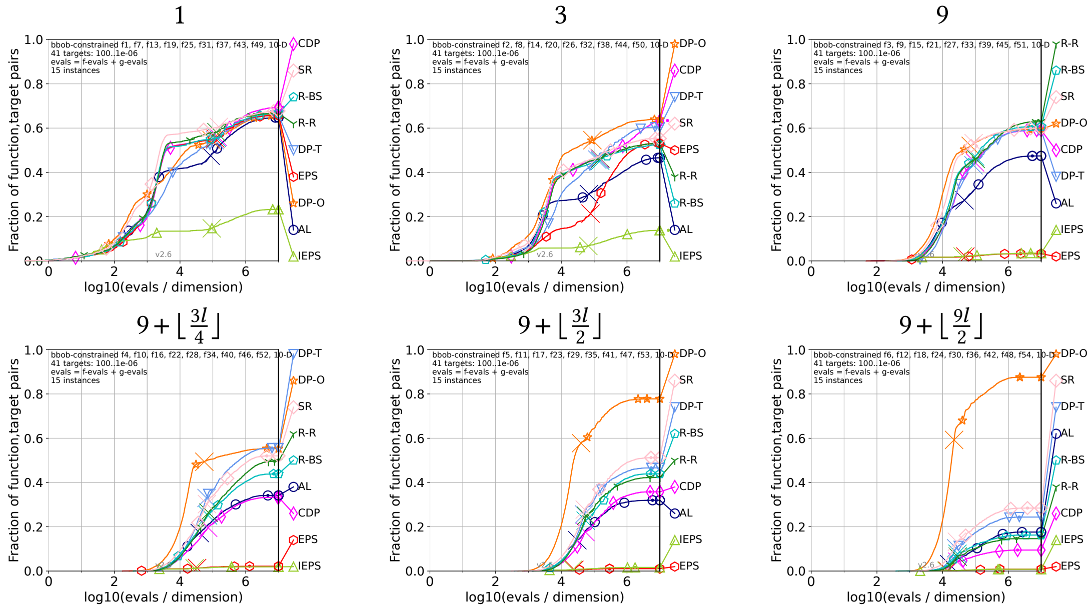


Figure 4.4: The results for the existing methods on the 10D instances of the bbob-constrained suite. The 54 functions are grouped by the number of constraints and active constraints, as described in Table 4.2. The number of total constraints is displayed above the figures.

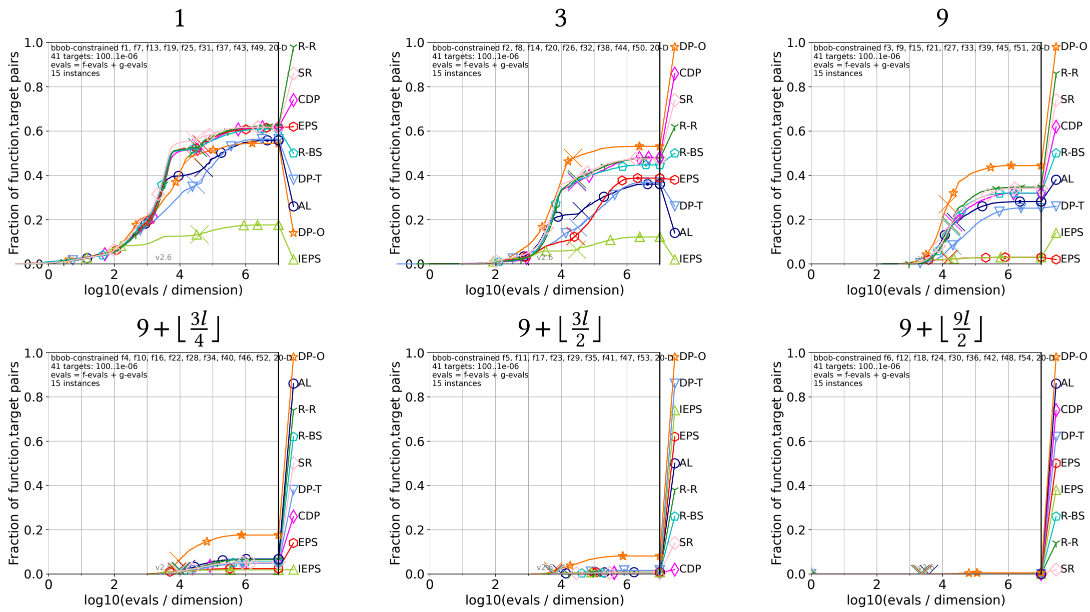


Figure 4.5: The results for the existing methods on the 20D instances of the bbob-constrained suite. The 54 functions are grouped by the number of constraints and active constraints, as described in Table 4.2. The number of total constraints is displayed above the figures.

methods and the AL method, which reaches fewer targets compared to the other methods. Notably, the performance of the AL method is similar up to a point where reaching further targets needs significantly more evaluations. The EPS method is able to perform similarly to the other methods on problems with up to 3 constraints, then there is a big drop in the number of targets reached. The IEPS method shows similar behavior, though it clearly reaches fewer targets on the problems with up to 3 constraints already.

A clear difference between the remaining methods begins to show in the 10D problems. In Figure 4.4, most approaches display a drop in the number of targets reached with an increased number of constraints. On the problems with more than 9 constraints, the DP-O method clearly reaches more approximation targets in fewer evaluations compared to the other methods. Compared to CDP, the SR and DP-T methods also reach more approximation targets on these problems. This holds true also for the repair methods, albeit with a smaller difference. While the AL method reaches fewer targets for up to 3 constraints, the method shows performance similar to CDP on problems with more constraints. For problems with a fixed number of constraints, success rates decrease as the number of constraints increases. Interestingly, this contrasts with the problems where the number of constraints is determined by the number of dimensions, where success rates seem to rise as more constraints are added. This anomaly is observed not only for the evaluated CHTs but also in other results presented in literature [19]. One plausible explanation is that this phenomenon is likely due to the problem construction.

In 20D, the overall success rate drastically decreases. Only for the problems with less than 9 constraints in Figure 4.5, more than half of the targets are reached by any method. However, this could be due to the exhaustion of the computational budget. For example, the slope of the line for the DP-O method with $9+\lfloor\frac{3l}{4}\rfloor$ constraints still indicates progress up to the point where the dot signifies reaching the time limit. Hence, the overall number of approximation targets hit would likely increase with a larger computational budget.

Nonetheless, assuming the observed trends continue, the DP-O method clearly outperforms the other methods both in terms of the proportion of targets reached as well as the number of evaluations needed for problems with more dimensions. Other methods show competitive performance only on the problems with a single constraint. In 2D, SR shows the best overall results. Considering the two dual-population methods, the DP-O method clearly outperforms DP-T. Nonetheless, given enough evaluations, the DP-T method performs well in terms of the proportion of targets reached up to 10D. The two repair methods perform similarly, and both tend to reach more targets compared to CDP.

4.4 Cone Problem

The results for the Cone problem are shown in figures 4.6 and 4.7. The different approaches are shown in two groups in order to increase the legibility of the plots. CDP, the currently used CHT is included as a reference in both figures. Note that the number of evaluations used is divided by dimension, making the horizontal grid lines correspond to linear growth in evaluations with increased dimensionality. The line style of the results indicates how reliably the approaches reach the approximation target of 10^{-10}. Solid lines correspond to

$\geq 90\%$ successful runs, dashed lines to $\geq 50\%$ and dotted lines to $\geq 10\%$. Less successful runs are not shown. A run can be unsuccessful by converging away from the constrained optimum or by exceeding the evaluation budget. The rows correspond to the block size and FOS used. The instances in the first row use a single problem of size l with the full FOS, while the other rows correspond to concatenated blocks of size 5 and 10 respectively. For the concatenated problems, the marginal product FOS with is used, such that each subset matches one block.

The results suggest a clear difference between a feasibility parameter of 180° and < 180°. This corresponds to a single active constraint and a constrained optimum located at the intersection of multiple constraints. An increase in evaluations needed with decreasing feasibility parameter is visible, however, small compared to the clear jump in resources needed between 180° and < 180°.

In Figure 4.6, AL performs clearly differently from the remaining CHTs. For the 180° problems, the target value is only reached reliably for the problems using the Full FOS and $l \geq 20$. For the Full FOS, the data spanning from 20 to 80 dimensions shows an increase in evaluations needed in line with the other methods. For the < 180° problems and Full FOS, the performance and success rate decreases with decreasing feasibility parameter of the problem. No reliable success is achieved for the 5° and 45° instances. However, the method shows the best performance with increased dimensions and reliable success on the instances with 90° and 135°. Similar to the 180° problem, the number of evaluations needed on low dimensional problems is comparatively high. The visibly linear increase in evaluations starting from $l = 10$ indicates quadratic growth in terms of evaluations needed in the observed dimensionality range. Considering the behavior of the AL method on the concatenated problems with the Marginal Product FOS, the success rate decreases, and the number of evaluations needed increases. This can be explained by the construction of the problem. The repeated blocks of the 5D instances lead to less success compared to the repeated blocks of size 10. This is in line with the performance using the Full FOS with problems, where the performance is more stable with increased dimensionality. Overall, the AL method shows competitive performance on the Full FOS instances where reliable success is observed. The reliability on the concatenated problems decreases compared to the instances with the same number of decision variables and a single block.This could be due to the block size being too small, as the performance on the 90° and 135° problems with block size 10 is better than the other methods for $l = 40$.

The remaining CHTs in Figure 4.6 all perform similarly to the CDP method for all FOS types. This is because of the initialization in the feasible region. As the methods differ in search behavior when the population is infeasible, these differences are not shown as the search in the feasible region effectively becomes CDP. IEPS deviates more, due to the additional rule that allows increasing the ϵ threshold when the population is feasible. The number of evaluations divided by dimensions needed for these methods grows linearly with increased dimensions for the dimensions tested. The transparent regression line in the color of CDP corresponds to quadratic scaling in evaluations w.r.t. dimensions for CDP. The fit of this regression line indicates quadratic growth. Considering the Marginal Product problems with repeated blocks of smaller instances, the increase in number of evaluations needed for twice the number of instances is less than double.

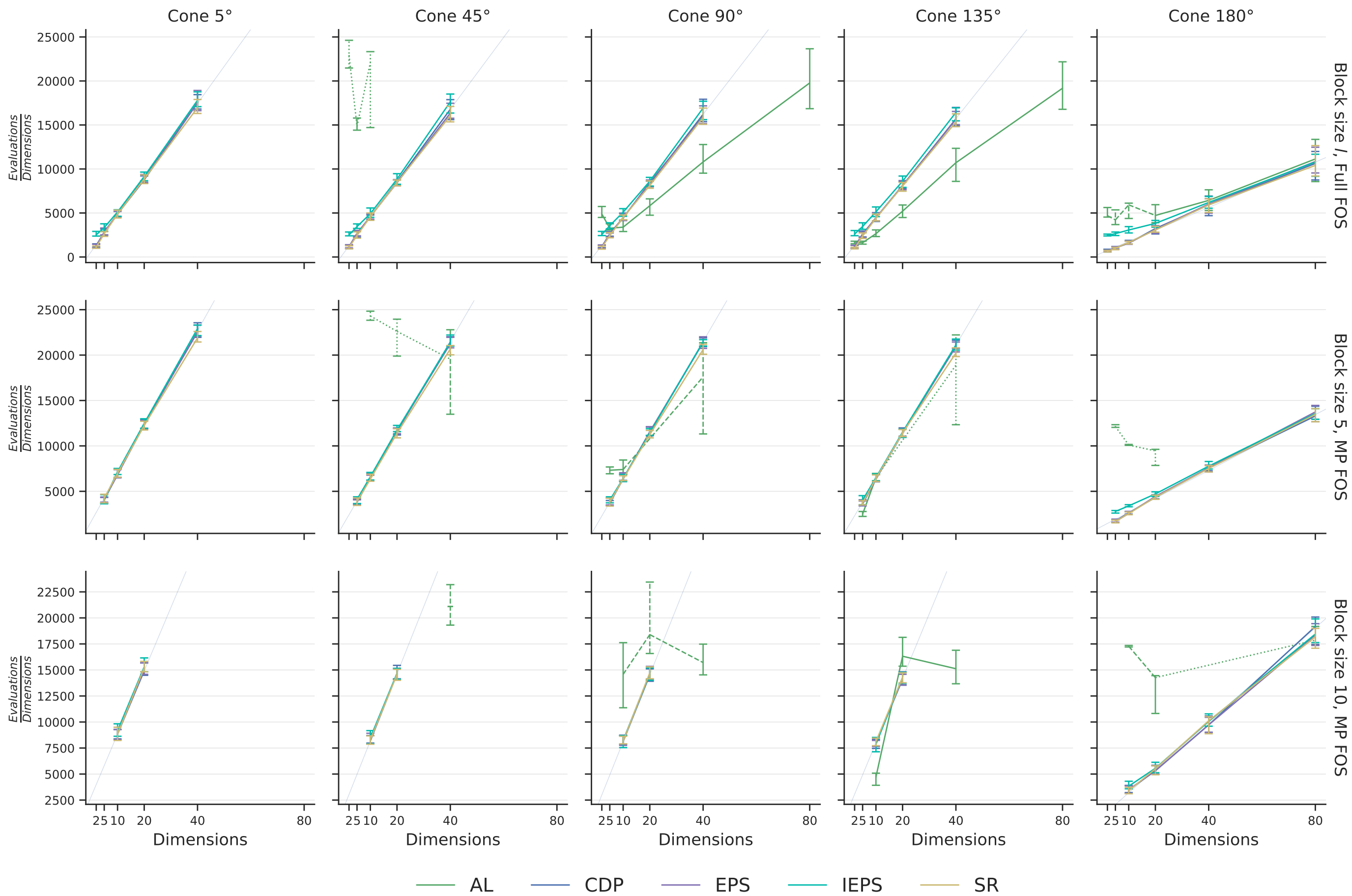


Figure 4.6: The median and interdecile range of the number of evaluations needed to reach the VTR of 10^{-10} on different Cone problem instances over 31 runs. The row corresponds to the block size used, and MP to the Marginal Product FOS. Solid lines indicate a success rate of ≥ 90%, dashed lines between 90% and 50%, dotted lines between 50% and 10% and less successful results are not shown.

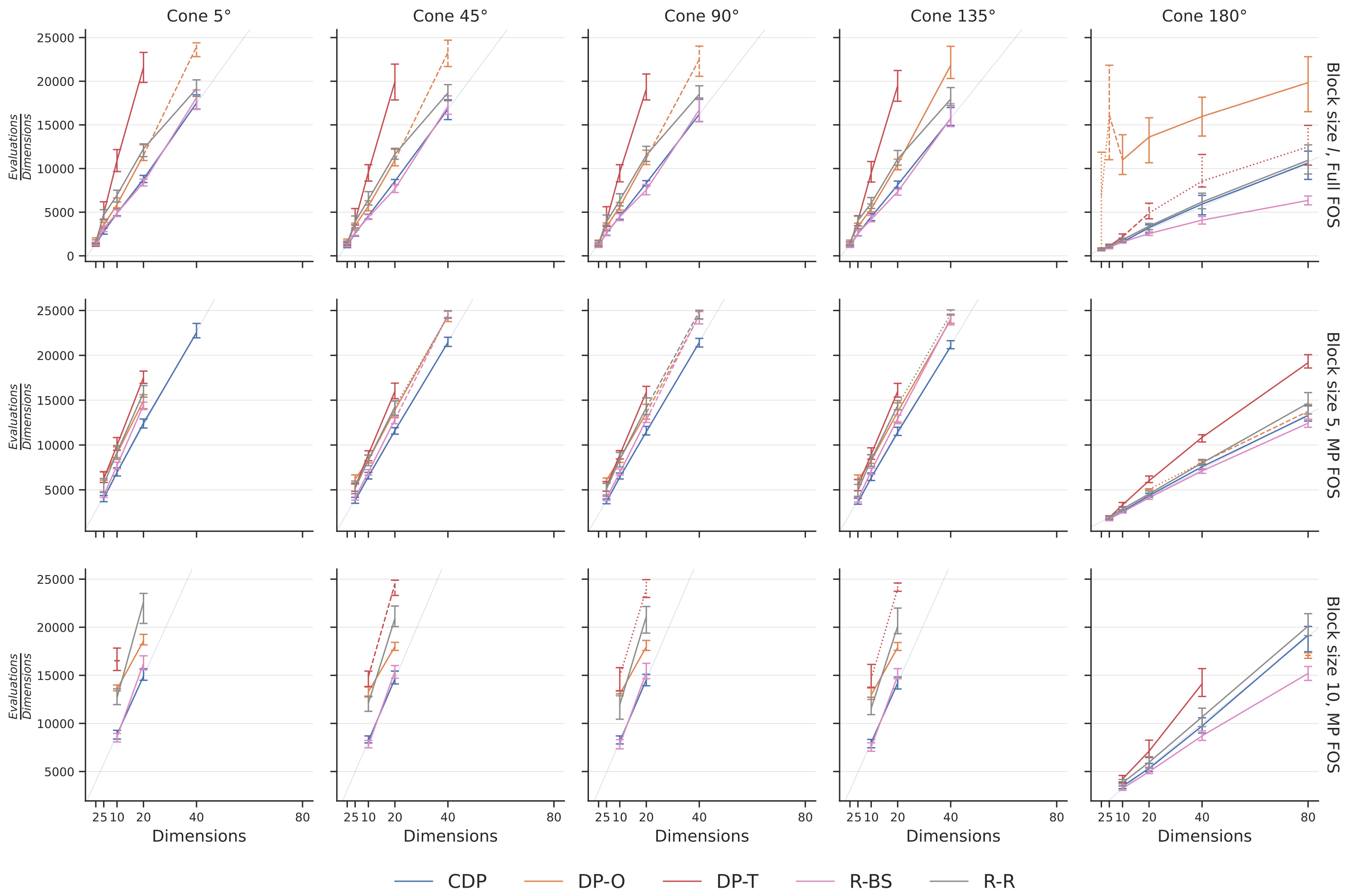


Figure 4.7: The median and interdecile range of the number of evaluations needed to reach the VTR of 10^{-10} on different Cone problem instances over 31 runs. The row corresponds to the block size used, and MP to the Marginal Product FOS. Solid lines indicate a success rate of ≥ 90%, dashed lines between 90% and 50%, dotted lines between 50% and 10% and less successful results are not shown.

In Figure 4.7, all methods show differences to CDP. Again, the differences between 180° and < 180° are large.

For 180°, repair using binary search shows the best performance and scales better than CDP. Repair using resampling performs similarly to CDP with the Full FOS, but scales worse when considering the concatenated problems using the Marginal Product FOS. The first dual population scheme, DP-O, fails to reliably reach the target value for the lower dimensions. With the Full FOS, the method clearly needs more evaluations compared to CDP. Nonetheless, the increases in evaluations from 10 to 80 dimensions suggest that the scaling behavior may be better than quadratic growth. In the observed dimension range, the method does not achieve better performance than CDP. For the concatenated problem instances, the minimum size needed for reliable results increases further. The second dual-population approach does not lead to reliable success with the Full FOS and no decomposition and scales worse than CDP on the problems with decomposition and the Marginal Product FOS. DP-T uses a second population, at times evolving more solutions compared to the other methods. Nevertheless, for most problems, the number of evaluations needed is less than double compared to CDP. While not more efficient w.r.t. evaluations needed, this indicates that the dual population scheme does benefit from the second, infeasible population.

On problems with < 180°, repair using binary search scales in line with CDP with no decomposition and worse with decomposition. The performance of the other Repair method using resampling also decreases when compared to CDP. Both dual population approaches scale worse than the other methods. DP-O fails to solve the problems with decreasing feasibility parameter reliably for higher dimensions. The median of the successful runs is close to the evaluation budget of $25000l$, indicating that the failed runs are possibly due to exhausting the computational budget rather than the method converging away from the constrained optimum. The decrease in reliable success for the DP-T method on the problems with decomposition also hints towards an exhausted evaluation budget. For the Full FOS, the number of evaluations needed for the DP-T method is close to double the evaluations needed when using CDP. This indicates that the second population is not effective for these problems.

Overall, there is a clearly visible difference in both the number of evaluations needed and the reliability of the different approaches between the problems with 180° and those with < 180°. For most approaches, further differences in the size of the feasible region have little effect. However, for the AL approach, there is a clear drop in performance for problems with ≤ 45°.

4.5 Effectiveness

In the previous section, experiments were performed to compare the performance of different CHTs and to find their strengths and weaknesses. However, in general, these are results that can only be considered in relation to other approaches. This is sufficient for the practical purpose of determining the best available optimization method for a given problem. Nonetheless, it is not possible to conclude that a particular CHT is performing effective search. This information is relevant for better understanding the strengths and weaknesses of an algorithm and potentially leads to algorithmic improvements by mitigating inefficiencies. Therefore, in this section, experiments are performed to gauge the effectiveness of the different CHTs when combined with RV-GOMEA. The main focus lies on CDP, the currently used technique, more detailed results for the other methods are in Appendix C.

Considering RV-GOMEA in particular, efficient search requires that the probability density of the estimated distributions matches the search space, i.e. solutions with better fitness should have a higher likelihood of being sampled. Ideally, over multiple generational steps this should increasingly be the case, i.e. the mean of the multivariate normal distributions should approximate the constrained optimum and the variance of the distribution should decrease.

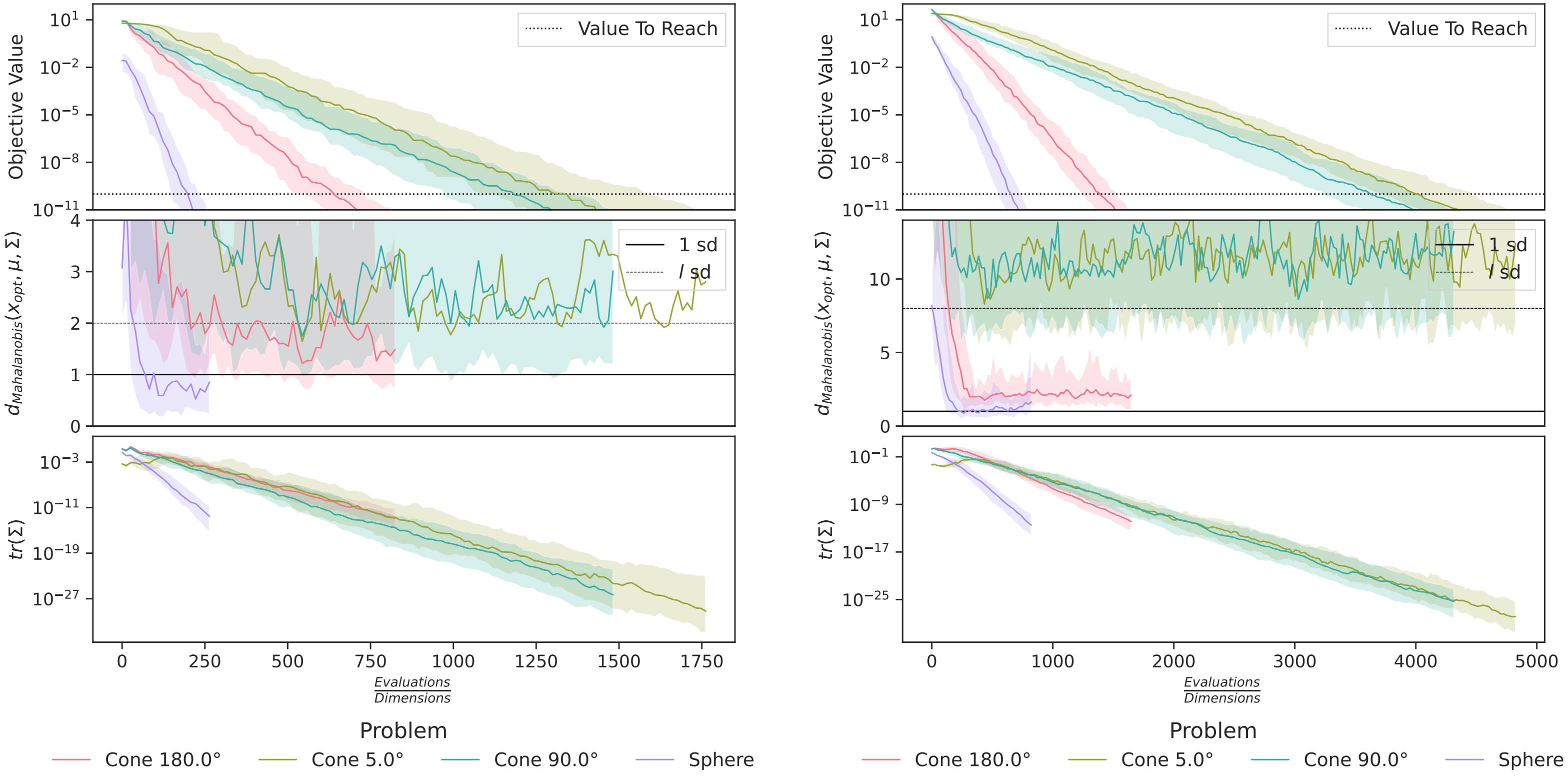
To test if this is the case, the trace of the covariance matrix and the Mahalanobis distance[40] between the distribution and the constrained optimum are analyzed for different problems. The trace of the covariance matrix gives an indication of how concentrated the probability density of the Gaussian distribution is and is defined as follows:

$$tr(\Sigma) := \sum_{i=1}^{l} (\Sigma)_{i,i} \tag{4.4}$$

Over time, this value should shrink as search is performed on an increasingly smaller scale. The Mahalanobis distance on the other hand is a distance metric that expresses the distance between a point and a distribution in standard deviations when taking the covariance into account. The definition is as follows:

$$d_M\left(\boldsymbol{x}, \hat{\mu}, \hat{\Sigma}\right) = \sqrt{(\boldsymbol{x} - \hat{\mu})^\top \hat{\Sigma}^{-1} (\boldsymbol{x} - \hat{\mu})} \tag{4.5}$$

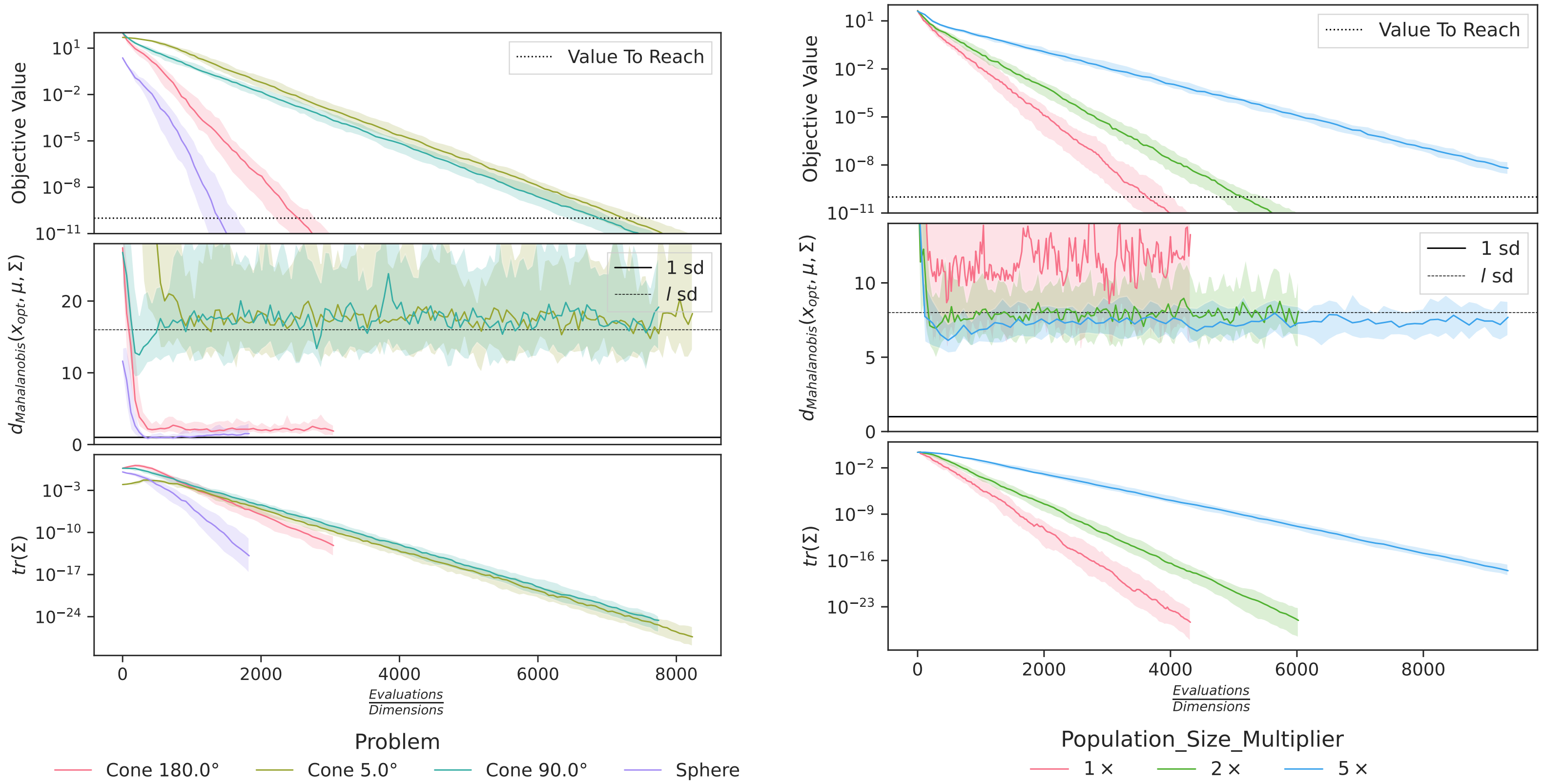
Given a distribution that models the fitness landscape well, this distance metric should be close to zero for the distance between the estimated distribution used by RV-GOMEA and the constrained optimum.



(a) All 2D problems using CDP

(b) All 8D problems using CDP

Figure 4.8: The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs.



(a) All 16D problems using CDP

(b) 8D Cone 90° problem using CDP

Figure 4.9: The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs.

Figures 4.8a,4.8b and 4.10a display these indicators for the fit of the estimated distribution and the corresponding objective value for multiple problems. In addition to different configurations of the previously introduced Cone problem, the unconstrained Sphere problem is shown as well. It can be seen that the achieved fitness is proportional to the Mahalanobis distance and directly proportional to the trace of the covariance matrix. Note that this follows from the unimodal construction of the used problems. If there are local optima the optimizer can get stuck in, the objective value and covariance can shrink while the Mahalanobis distance to the constrained optimum will increase. The difference in evaluations needed to reach the target value of 10^{-10} can be explained by the differences in the indicators of the distribution fit. A small Mahalanobis distance to the optimum together with a small covariance matrix trace indicates that the optimum is both in distribution and the probability density is concentrated, leading to efficient search. This can be observed for the unconstrained Sphere problem, where the steepest descent of the objective value w.r.t. evaluations can be observed. For the Cone problem with 180° feasibility, the constraint boundary is a single linear constraint. As CDP prioritizes feasible solutions, only solutions on the feasible side of the constraint boundary will be selected for distribution estimation. Because of this bias towards feasibility, the Mahalanobis distance stays larger and the performance decreases. For the Cone problems with 90° and 5° feasibility, the Mahalanobis distance increases with the dimensionality to the extent that sampling the constrained optimum is very unlikely. For these problems, the covariance shrinks, while the Mahalanobis distance stays relatively constant after the initial generations. This indicates that the distribution mean keeps getting closer to the constrained optimum. However, the Mahalanobis distance does not improve because the distribution shrinks, maintaining a similar likelihood of sampling the constrained optimum. Compared to the Sphere problem, the target is not reached because a distribution with a good fit shrinks over generations, but rather because each generation the distribution moves towards the optimum and shrinks until the target value is reached.

This can be interpreted as a trade-off between exploitation and exploration. The center bias of the multivariate normal distribution used in RV-GOMEA leads to a higher search resolution closer to the mean. Thus, for the Sphere problem, with a good fit of the distribution, more sampled GOM solutions will be improving in fitness, leading to increased selection pressure. On the other hand, for problems where the improvements are out-of-distribution, i.e. away from the distribution mean, fewer samples will be improvements in fitness, and selection pressure is lower. Consequently, this explains the difference in evaluations needed between the different problems, as the exploitation phase is never reached for the Cone problems with 90° and 5° feasibility.

Compared to the objective value and the trace of the covariance matrix, the variance of the Mahalanobis distance w.r.t. evaluations used is clearly larger. Figure 4.10b shows that this is due to the population size used. The recommended population size generally corresponds to a population size that is large enough to solve the problem, but small enough such that there is little redundancy and evaluations are thus used effectively. Using larger multiples of the recommended size, both variance and value of the Mahalanobis distance clearly shrink. This corresponds to better estimated distributions with an increased likelihood of sampling close to the constrained optimum. However, the number of evaluations needed increases as well. Put differently, considering the population size, there is a trade-off between increasing certainty and introducing redundancy. In this case, the increased certainty that comes with larger population sizes is not worthwhile when considering the number of evaluations needed.

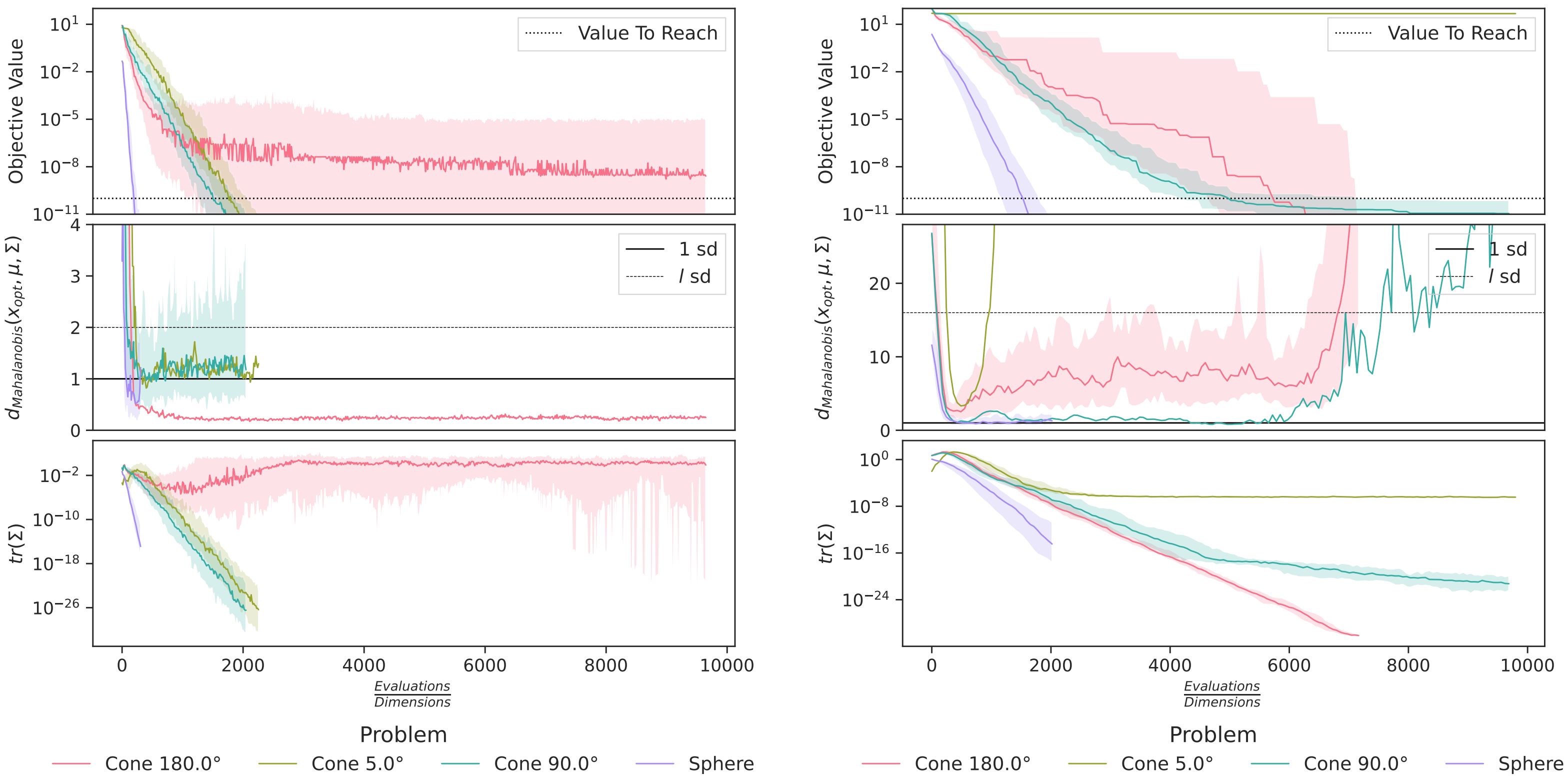
Another observation is that the fit of the distribution depends on the nature of the constraints. The Cone problem with a single linear constraint (180°) leads to better distributions than the same problem with smaller feasible angles. Furthermore, the angle does matter as well.

Considering this, it is clear that search close to the constraint boundaries is not as effective as RV-GOMEA unconstrained optimization using the same objective function. This can be explained by the fit of the estimated distribution, which indicates that for the constrained problems the exploitation scenario is never reached. To answer the second research question, the presence of active constraints leads to decreased search effectiveness. This drop in efficiency is significantly larger for Cone problems with a feasibility parameter $< 180°$, corresponding to problems with multiple active constraints. A possible explanation for this lies with the estimated distribution, as sampling closer to the constrained optimum becomes increasingly unlikely.

4.5.1 Other CHTs

The results for the other tested CHTs are largely in line with those of CDP and included in Appendix C. For the SR, EPS, and IEPS methods, this is explained by the initialization in the feasible region. By doing so, the ϵ comparison used in EPS and IEPS effectively behaves identically to CDP. Similarly, the feasible population together with using CDP to accept improvements leads to a selection in line with CDP for SR. The repair methods R-BS and R-R do influence the number of evaluations needed and the distribution indicators, but the changes are small.

Notable differences can be observed for the dual population approaches and the AL method. The DP-O method is seemingly limited in the scaling of the step size performed. For the Cone problem with 180°, optimization gets stuck. While the Mahalanobis distance is small, the covariance is large and does not shrink. This likely is explained by the acceptance criteria used for the subpopulations and the linear constraint boundary. In each generation, the feasible subpopulation crosses into the infeasible region and vice versa. This leads to the two subpopulations replacing each other, with no net gains towards closing in on the constraint boundary. The extent of this decreases with the dimensionality, and the problem gets solved reliably for the 8D and 16D instances. For the DP-T version, the second population naturally has an effect on the number of evaluations needed. The AL method highlights the difficulty of configuring penalty methods correctly. The Cone problem with 90° clearly is solved more efficiently than when using CDP, with both a small Mahalanobis distance and decreasing covariance. However, the AL method is not able to reliably solve the other Cone problems. After some initial progress, the covariance stops decreasing and the optimization stops progressing towards the constrained optimum.



(a) All 2D problems using DP-O

(b) All 16D problems using AL

Figure 4.10: The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs.

4.6 Summary

Overall, the performance of the various CHTs clearly shows that the approaches have different strengths and weaknesses. Looking at the previously introduced desired qualities of a CHT, the experiments performed suggest the following answer to the first research question:

- AL shows the best results in terms of efficiency on some problems, but clearly does not perform reliably on all problems. The limited applicability and parameter fragility demonstrate that it is hard to define a penalty function that performs well on all problems.
- Both the EPS and IEPS methods demonstrate that the tested ϵ-constrained techniques do not work well with RV-GOMEA. This likely is primarily due to the decrease of the threshold being tied to the computational budget and not the $<\epsilon$ comparison. The budget needed varies for different problems, which prevents a robust setting of the parameters across a wide range of problems.
- The repair methods pose a trade-off between spending more evaluations for increased selection pressure. On most problems, the performance is similar to CDP at the expense of additional parameters. An exception is the Cone problem with 180° feasibility, where repair using binary search shows a clear advantage.
- CDP is a robust baseline method that works on all problems tested without the need for parameter tuning. However, in terms of efficiency, the method is outperformed in both the feasible and infeasible region. In the infeasible region, the objective value is not considered, the technique is often outperformed by SR. In the feasible region, the selection of primarily feasible solutions leads to estimated distributions that prevent effective approximation of constraint boundaries.
- The dual-population approach DP-O shows robust performance on the CEC2006 and bbob-constrained problems, albeit more evaluations are needed compared to other methods. The second dual-population approach DP-T is competitive with CDP on the bbob-constrained problems, however, it generally takes more evaluations and is less reliable compared to the DP-O method.
- Especially on the CEC2006 problems where the optimization often starts in the infeasible region, the SR method shows robust results with impressive performance. On the other problems, the method is competitive with CDP. When combined with RV-GOMEA and using CDP for acceptance during variation, the P_f parameter is robust across the different problems tested.

The different methods tested either have limited general applicability or perform inefficient search in the feasible region when the optimum is located near a constraint boundary. This issue caused by the distributions estimated solely from feasible solutions. To mitigate this and to allow more effective search near constraint boundaries while also achieving reliable results, new approaches making use of the infeasible solutions encountered during optimization are needed.

Chapter 5

Proposed Improvements

There was no fully satisfactory CHT among the approaches evaluated in Chapter 3. The experiments show that either the effectiveness of the CHT is not as desired, the parameter settings are fragile or there are clear limitations. To this end, two adaptations to existing techniques are presented to mitigate some of the drawbacks and to facilitate more effective search close to constraint boundaries. The first presented method aims to mitigate the search bias induced by selecting only feasible solutions when using CDP. The second proposed method is designed to make the dual-population idea of approaching from both sides of the boundary work better with RV-GOMEA.

For both methods, parameter tuning is performed in Appendix A. Using the best performing configurations, the experiments from Chapter 4 are repeated, showcasing how the proposed changes compare to the other CHTs from Chapter 3.

5.1 Partially Infeasible Selection

CDP is parameter free and directly biases search towards the desired result, i.e. high quality feasible solutions. There is no need to perform parameter tuning, and there are no limits to the applicability of the technique. However, as the previous experiments have shown, the effectiveness of the search near constraint boundaries shows potential for improvement when used with RV-GOMEA. The changes proposed in the following paragraphs aim to mitigate this issue.

5.1.1 Motivation

Using CDP, the selection favors feasible solutions, and thus the subsequent search distributions will also favor exploring the feasible region instead of searching close to the constraint boundary. Due to the center bias of the Gaussian distribution, most samples drawn will be close to the distribution mean. For search close to a constraint boundary, this means that the search space between the constraint boundary and the feasible selection is likely out-of-distribution and not the focus of the search. If better solutions are located closer to the constraint boundary, this leads to ineffective search.

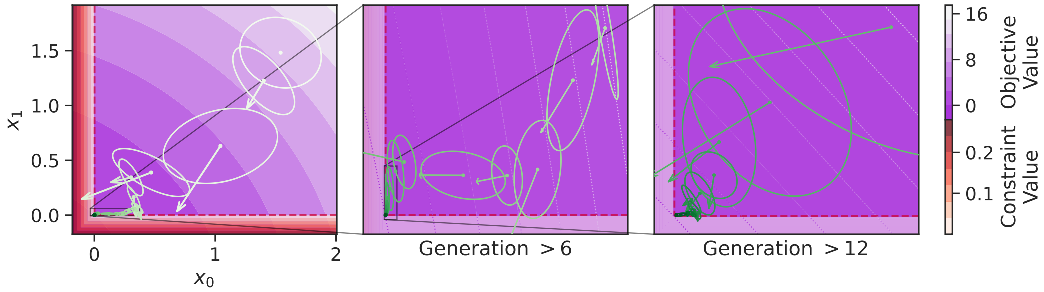


Figure 5.1: The estimated distributions and AMS vectors of the CDP method using the Full FOS on the Cone 90° problem. The ellipses correspond to a Mahalanobis distance of 1, and the color gradient from white to green indicates the increasing generation, the first 24 generations are shown.

Note that RV-GOMEA employs AVS and SDR [7, 9] to adapt to improvements being away from the distribution mean, however, as the experiments in Section 4.5 have shown these mechanisms are not sufficient in this case. For unconstrained problems, AMS solves a similar problem of the distribution naturally aligning with the contour lines rather than with the direction of steepest fitness increase[6]. However, applying AMS does not mitigate the inefficiency in this situation, as solutions shifted into the infeasible region are rejected by CDP.

Overall, these three mechanisms lead to the distribution shrinking at the boundary, such that the search is effectively performed only in the feasible region. Additionally, the distribution orients itself along contour lines of the objective function as AMS leads to infeasible and thus rejected solutions. Once the distribution has reoriented and shrunk enough, AMS does not reach beyond the constraint boundary anymore and reorients the search towards the constraint boundary as desired. This closes the gap between the search distribution and the constraint boundary. Once the distribution is close enough to the boundary again, this cycle repeats. This is illustrated in Figure 5.1. Nonetheless, as the distributions shrink, the achieved fitness continues to improve. The proposed method aims to avoid these generations spent on scaling down and reorienting the distribution by keeping the search at the constraint boundary.

5.1.2 Proposed Changes

To alleviate this bias towards feasibility and to facilitate search at constraint boundaries, it is necessary to also consider infeasible solutions during selection to place the distribution mean at the constraint boundary. With CDP as the main CHT, infeasible solutions will not be selected as long as there are enough feasible solutions in the population. Hence, the selection of infeasible solutions needs to be performed separately from the selection of feasible solutions. However, with CDP being used also for GOM acceptance, over time the number of infeasible solutions in the population will decline. Infeasible solutions will be replaced by feasible solutions, and new infeasible solutions will not be accepted. Nonetheless, during search, infeasible solutions are encountered. These can be used for this second

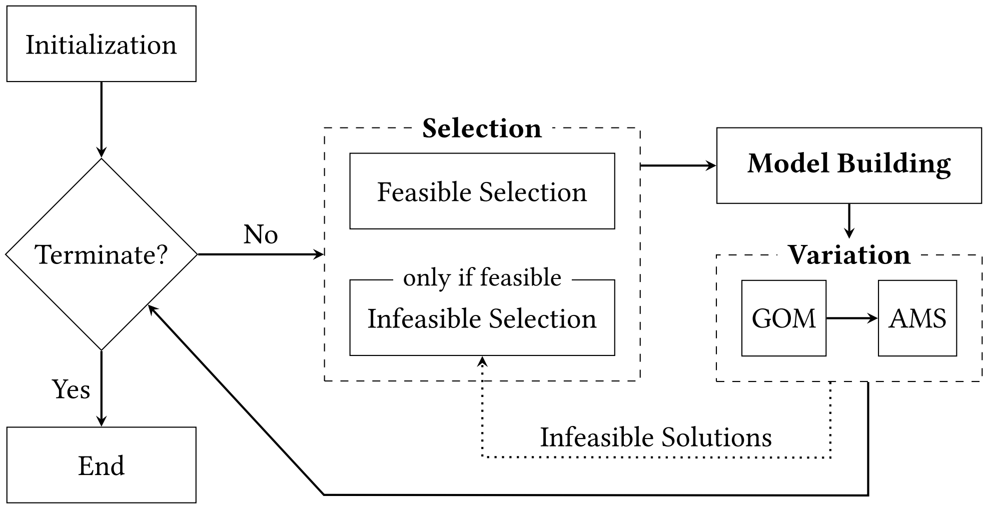


Figure 5.2: The outline of the proposed modification to RV-GOMEA. If the population is feasible, infeasible solutions encountered during the previous variation step are subject to a second selection in order to mitigate the bias towards feasibility induced by CDP.

selection of infeasible solutions in order to mitigate the bias towards feasibility induced by CDP. Contrary to other EAs, however, RV-GOMEA does not have an offspring population. Where other EAs first produce an offspring population in the variation step and then select the subsequent population from the available solutions, RV-GOMEA performs both during GOM for the individual solutions in the population. To have a candidate pool of infeasible solutions, all encountered but rejected infeasible solutions are collected, both during GOM and the full AMS steps, i.e. during variation.

Figure 5.2 depicts this modification to RV-GOMEA. Note that the infeasible selection is only performed if the population is in the feasible region. The criterion used for when the population is in the feasible region is when η% of the individuals in the population are feasible, where $\eta \in [0, 1]$ is a new parameter. In the infeasible region, using stochastic ranking instead of CDP is also considered. To implement this, several design choices have to be made:

1. *How many infeasible solutions should be used?* Using an equal split of feasible and infeasible solutions would concentrate the search at the boundary. However, as more GOM samples would be infeasible, the acceptance rate and thus the quality of feasible solutions would decrease, slowing down optimization progress. On the other hand, using no infeasible solutions does not mitigate the observed search bias that slows down exploration near constraint boundaries.
2. *Which infeasible solutions should be used?* This choice resembles the choice of a CHT in itself, as the aim is to select the infeasible solutions that best guide evolution toward the constrained optimum. Additionally, selecting infeasible solutions may not always be desired, i.e. when feasible solutions of better fitness are not closer to the constraint boundary than the current population.

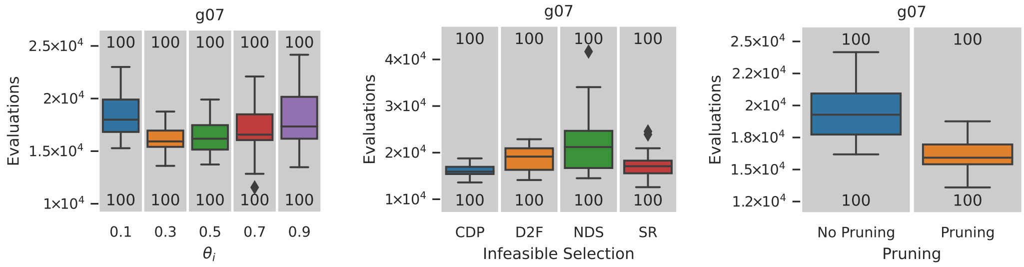


Figure 5.3: The effect of different parameters on problem g07 from the CEC2006 problems. The remaining parameters are set to the best performing version in Table A.9. The values at the top/bottom of the bar correspond to the feasible rate and success rate respectively.

3. *How should the infeasible solutions influence the next distribution?* One way of estimating the distributions is to replace a few of the feasible selected individuals. However, this changes the selection pressure for the feasible solutions as fewer solutions are selected. Another approach could be to simply add the infeasible solutions to the feasible selection, increasing the overall amount of information used to estimate the distribution. However, this possibly increases the variance and spreads the probability density.

How many infeasible solutions should be used? This is implemented as an additional parameter denoting the fraction of the population size θ_i which is the desired number of infeasible solutions used. A value of $\theta_i = 0$ corresponds to using no infeasible solutions, i.e. the result is CDP. At most half of the combined selection should be compromised of infeasible solutions to ensure that there is search on the feasible side of the constraint boundary. Furthermore, high values for θ_i lead to more probability density in the infeasible region and thus fewer feasible solutions are sampled in total. This decreases the selection pressure for the feasible selection. Hence, this introduces a trade-off where the lower number of feasible solutions sampled need to be of higher quality in order to make the decreased chance of searching in the feasible region worthwhile. This can be seen in Figure 5.3.

Which infeasible solutions should be used? To find a good choice, multiple selection strategies are tested. The following methods of selecting the infeasible solutions were considered:

1. CDP, leading to selecting solely based on the constraint value.
2. Non-dominated sorting (NDS), with the objective and the constraint value. This method is often used to find the best trade-offs available between multiple objectives in multi-objective optimization [18].
3. Selection based on distance to the feasible region (D2F), preferring infeasible solutions closer to the feasible solutions selected.
4. Stochastic Ranking

In addition to the selection, the infeasible solutions were further restricted such that only infeasible solutions with better objective value than the best feasible solution are considered. This is to make sure that finding solutions similar to the infeasible solutions is desirable. The reasoning behind this is that there is no need to get closer to a constraint boundary if the fitness near the boundary is not better. The performance with the different selection methods is shown in Figure 5.3, where each selection method is included combined with pruning. Furthermore, pruning has a big impact on the number of evaluations needed on problem g07. Overall, CDP with pruning performed best in Table A.9.

How should the infeasible solutions influence the next distribution? The way in which the infeasible solutions influence the estimated distribution needs to be determined. One approach is to maintain the number of solutions influencing the subsequent distribution, thus infeasible solutions replace otherwise feasible solutions. The second option considered is adding the infeasible selection to the feasible selection, increasing the total number of selected individuals. This option was included since the size of the selection pool increases because the infeasible solutions are not selected from the population but from previously encountered infeasible solutions. Note that when using learned linkage models, the model is learned from the correlation matrix of the selected solutions. The effect of including or omitting the infeasible solutions in this step has not been considered in this thesis. The parameter tuning results, included in Table A.9, suggest that replacing feasible solutions performs better.

5.2 Evolving Infeasible Solutions with a Partially Infeasible Population

The changes proposed in Section 5.1 aim to facilitate more efficient search close to constraint boundaries. However, the infeasible solutions used are not subject to further evolution compared to the feasible individuals in the population. Similar to the previously introduced approach, the method described in this section also handles feasible and infeasible solutions separately. However, infeasible solutions are subject to evolution as well and the overall scheme follows the dual population idea of actively approaching the constraint boundary from both sides. As the DP-O dual-population approach showed promising results on the bbob-constrained problems in Chapter 4, the proposed changes are based on this method in an attempt to adapt the method to work better with RV-GOMEA.

Merging the Distributions The first change considered is performing the distribution estimation and linkage learning jointly, not in separate per subpopulation. Where DP-O estimates two distributions with means on both side of the constraint boundary, this method should have the mean and thus most probability density close to the constraint boundary. The contribution to the selection of feasible/infeasible solutions is done proportionally to the number of feasible/infeasible solutions in the population. This avoids the need of either doing evolution on a smaller scale with a proportional approach or decreasing the selection pressure by keeping the selection size constant but decreasing the candidate pool, as done previously when adapting the DP-O version. In terms of previous work, this corresponds to the dual population scheme from [25] with handling of infeasible solutions similar to [51] and [27]. In other words, segregational selection is performed to avoid the bias induced by CDP and depending on the feasibility of the population, the constraint boundary is either approached from one side or both sides at once. A flowchart displaying the changes made to DP-O is shown in Figure 5.4.

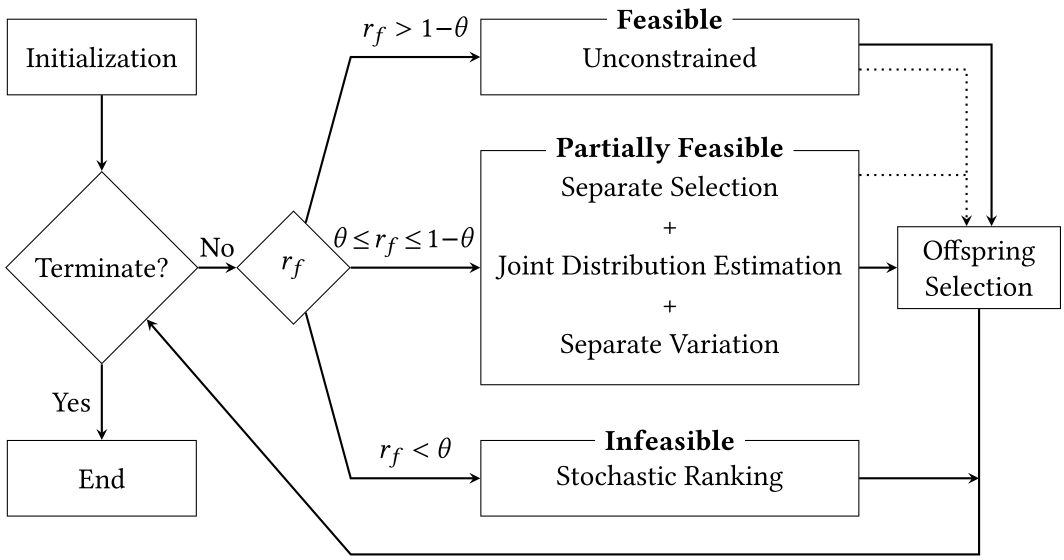


Figure 5.4: The outline of the proposed approach based on DP-O, where a partially feasible population is used instead of two subpopulations. In this mixed population, selection and variation is split by feasibility and model building is done jointly. Additionally, previously seen feasible solutions that were replaced by infeasible solutions, indicated by the dotted lines, are reconsidered during the added offspring selection step.

Search in Infeasible Subpopulations In addition to the previous change, the objective used for the infeasible solutions is modified. In DP-O, the infeasible solutions aim to become feasible. However, feasibility alone does not consider the objective value or the proximity to the feasible solutions. The experiments performed in Chapter 4 suggest that considering feasibility alone does not lead to the best performance. When considering optimization in the infeasible region, Stochastic Ranking showed promising results, leading to a trade-off between objective value and feasibility. Thus, the goal of the infeasible solutions is modified to use Stochastic Ranking for both the infeasible population and infeasible subpopulation. The implementation of this is in line with the Stochastic Ranking method adapted to RV-GOMEA in Chapter 3, i.e. the stochastic operator is only used for selection and $P_f = 0.45$ is used.

Preventing Information Loss In DP-O, the feasible solutions aim to improve in objective value and the infeasible ones aim to become feasible. This leads to loosing feasible solutions that are in the population in favor of infeasible solutions with better fitness. This is conflicting with the overarching goal of finding high quality feasible solutions, as feasibility should be prioritized over objective value. To prevent this loss of information, the solutions are collected similarly to the PIS approach. However, in this case, the feasible solutions are collected before they are replaced with infeasible solutions. At the end of the generation, the collected solutions are used in an additional offspring selection step together with the individuals in the population. The selection is performed using truncation separated by feasibility, similar to the selection for the distribution. The selection is based on the objective of each population, i.e. CDP for the infeasible solutions and the objective value for feasible solutions.

5.3 Experiments

In this section, the experiments previously performed using existing CHTs from literature are repeated with the proposed methods to allow a straightforward comparison. In order to avoid confusing plots containing the data of all CHTs, the methods are mainly compared to the currently employed CDP method.

The method names are abbreviated, such that PIS corresponds to the first proposed method using a partially infeasible selection. For the second proposed method, two versions are included. PI uses a single distribution, while DP-PI uses two populations like DP-O but reconsiders feasible solutions that would be lost when feasible solutions cross into the infeasible region. Both variants utilize stochastic ranking for infeasible subpopulations. Detailed results for determining the best parameter configuration can be found in Appendix A.

5.3.1 CEC2006

An overview of the results of the proposed methods and CDP for the CEC2006 problems is presented in Table 5.1, while detailed outcomes can be referenced in Appendix B. CDP is used as a reference, as it is the CHT currently used in RV-GOMEA. The feasible rate is not shown, since all methods were able to reliably reach a feasible region except for problem g22 where the feasible region was never reached. Overall, the success rate is consistent with the previous results for the feasibility oriented methods. Compared to CDP, all methods are never significantly worse and significantly better at reaching the approximation target on at least one problem. This difference can be observed for problems g12 and g17.

In terms of the success performance, PIS performs better than CDP on 19 problems and worse only on 3 problems. PI and DP-PI perform similarly to CDP when considering the number of times one method was better. Compared to DP-O, this is a notable improvement, as DP-O performed worse than CDP on 20 problems. This indicates that by utilizing infeasible solutions, constraint handling performance can be improved.

Considering the other existing CHTs, the proposed methods are competitive in terms of success rate and achieve significantly better results on some of the problems. In terms

	Success Rate				Success Performance			
	CDP	DP-PI	PI	PIS	CDP	DP-PI	PI	PIS
g01	76.0%	**84.0%**$^{\approx}$	80.0%$^{\approx}$	**84.0%**$^{\approx}$	32534.5	52165.9	70895.8	**31127.6**
g02	0.0%	**24.0%**$^{+}$	16.0%$^{\approx}$	4.0%$^{\approx}$	-	425000.0	**293550.0**	722775.0
g03	**100.0%**	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	48991.6	20217.0	20271.7	**19398.3**
g04	**100.0%**	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	5423.7	6571.7	6694.3	**4622.8**
g05	**100.0%**	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	35441.0	19371.8	13086.0	**8028.4**
g06	**100.0%**	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	2300.1	1793.6	**1743.2**	2151.1
g07	**100.0%**	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	20423.8	18706.2	22414.2	**16047.0**
g08	**100.0%**	96.0%$^{\approx}$	92.0%$^{\approx}$	88.0%$^{\approx}$	**232.6**	792.8	645.6	321.5
g09	**100.0%**	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	7422.0	6544.2	7080.9	**5677.4**
g10	**100.0%**	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**24388.0**	37638.4	128690.0	30181.4
g11	**100.0%**	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	4083.8	**1859.5**	2031.8	2389.4
g12	44.0%	**84.0%**$^{+}$	**84.0%**$^{+}$	76.0%$^{+}$	1633.1	731.8	**625.2**	774.8
g13	32.0%	40.0%$^{\approx}$	48.0%$^{\approx}$	**60.0%**$^{+}$	280714.8	146339.8	146451.9	**95605.2**
g14	**100.0%**	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**12976.7**	24463.9	22727.6	14287.4
g15	**100.0%**	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	23149.6	10954.9	**9256.2**	11969.8
g16	**100.0%**	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	4490.0	4968.8	5077.7	**3885.2**
g17	16.0%	**60.0%**$^{+}$	40.0%$^{\approx}$	56.0%$^{+}$	473603.1	167183.0	193848.2	**124345.3**
g18	84.0%	84.0%$^{\approx}$	**96.0%**$^{\approx}$	88.0%$^{\approx}$	11794.3	13886.2	10699.7	**10214.6**
g19	**100.0%**	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	42077.3	61539.2	66860.7	**39431.6**
g21	**100.0%**	96.0%$^{\approx}$	**100.0%**$^{\approx}$	92.0%$^{\approx}$	35539.8	37228.0	52343.8	**34987.1**
g22	**0.0%**	**0.0%**$^{\approx}$	**0.0%**$^{\approx}$	**0.0%**$^{\approx}$	-	-	-	-
g23	**100.0%**	88.0%$^{\approx}$	96.0%$^{\approx}$	96.0%$^{\approx}$	39846.2	68676.5	43860.1	**39718.3**
g24	**100.0%**	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	**100.0%**$^{\approx}$	836.2	851.9	932.4	**670.5**
+/≈/−	-	3/20/0	1/22/0	3/20/0	-	-	-	-
</>	-	-	-	-	-	11/11	11/11	19/3

Table 5.1: The Success Rate and Success Performance metrics for the CEC2006 problems. The best values for each problem are highlighted in bold. The +/≈/− symbols indicate significantly better/not different/worse performance when compared to CDP using Fisher's exact test [23] with $p < 0.05$. The bottom rows display the total number of times the CHT was better/not different/worse compared to CDP for the Success Rate, and the total number of times a CHT achieved a better (<) or worse (>) Success Performance than CDP.

of success performance, PIS shows results similar to SR while the PI and DP-PI methods improve compared to the other dual population methods.

5.3.2 bbob-constrained

The summary results of the proposed methods for the bbob-constrained suite are shown in figures 5.5-5.8. As a reference, CDP and DP-O are included as the current default and the overall best performing approach on bbob-constrained from Chapter 4. Similar to the results for the existing feasibility oriented CHTs, for the 2D and 5D problems, there is little distinction between the different methods with the exception of CDP. Compared to

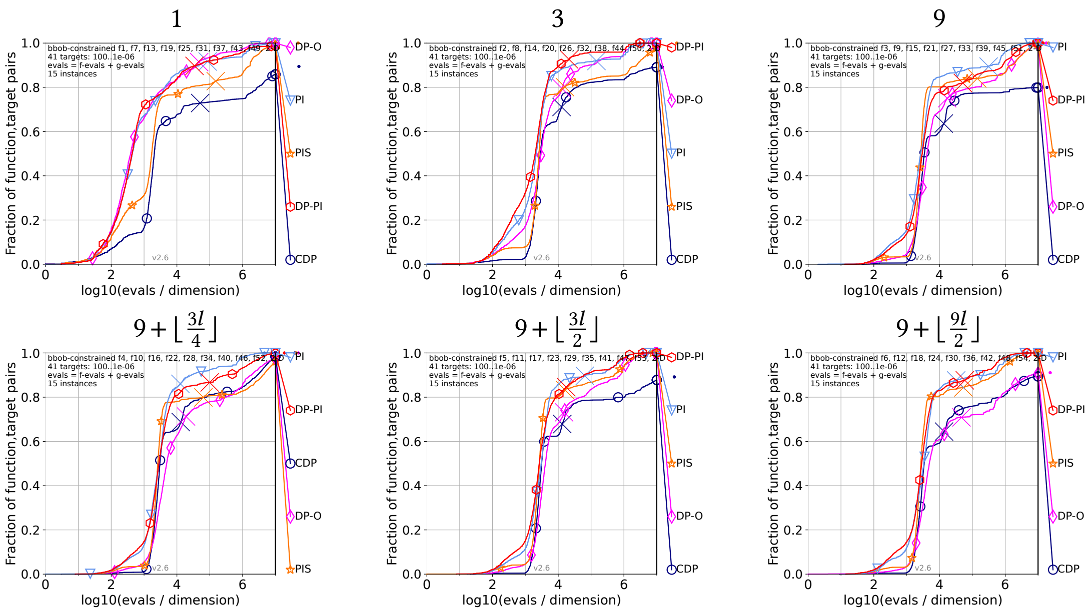


Figure 5.5: The results for the bbob-constrained suite, grouped by the number of constraints, as described in Table 4.2. The number of total constraints is displayed above the figures.

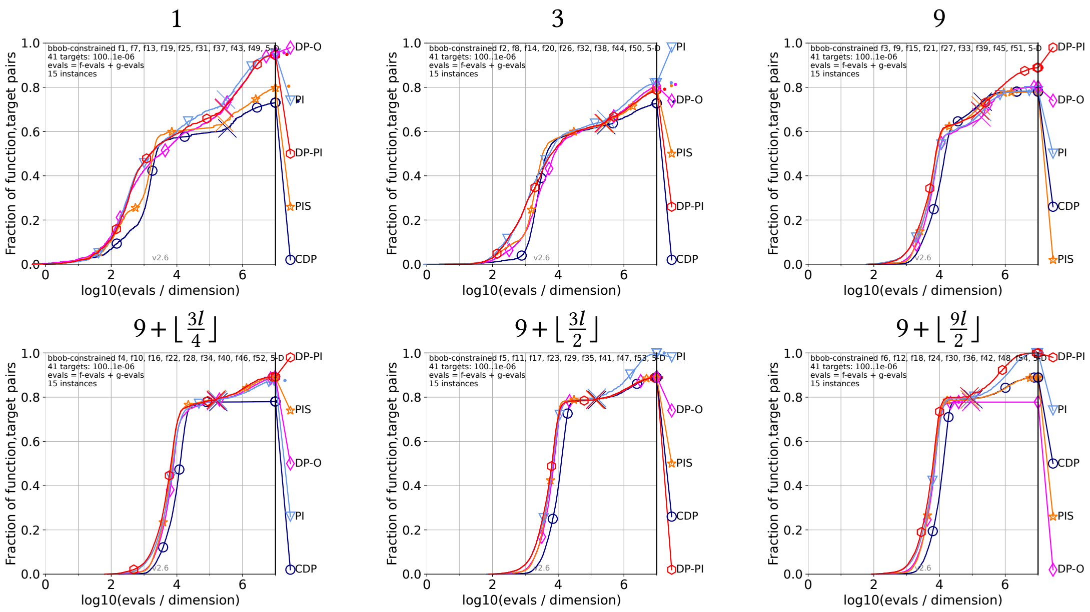


Figure 5.6: The results for the bbob-constrained suite. The 54 functions are grouped by the number of constraints and active constraints, as described in Table 4.2. The number of total constraints is displayed above the figures.

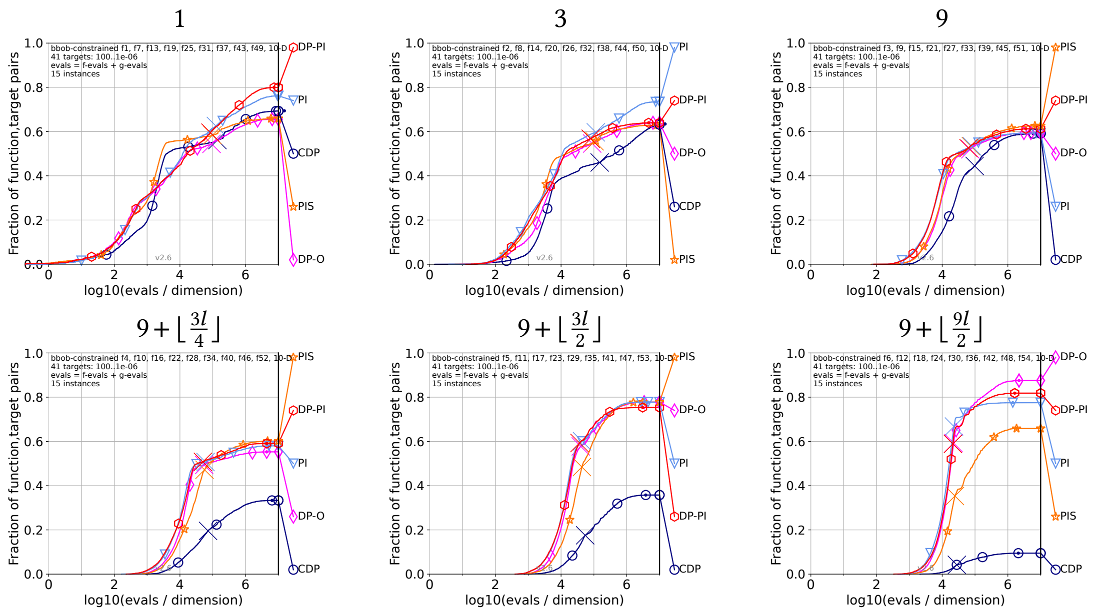


Figure 5.7: The results for the bbob-constrained suite. The 54 functions are grouped by the number of constraints and active constraints, as described in Table 4.2. The number of total constraints is displayed above the figures.

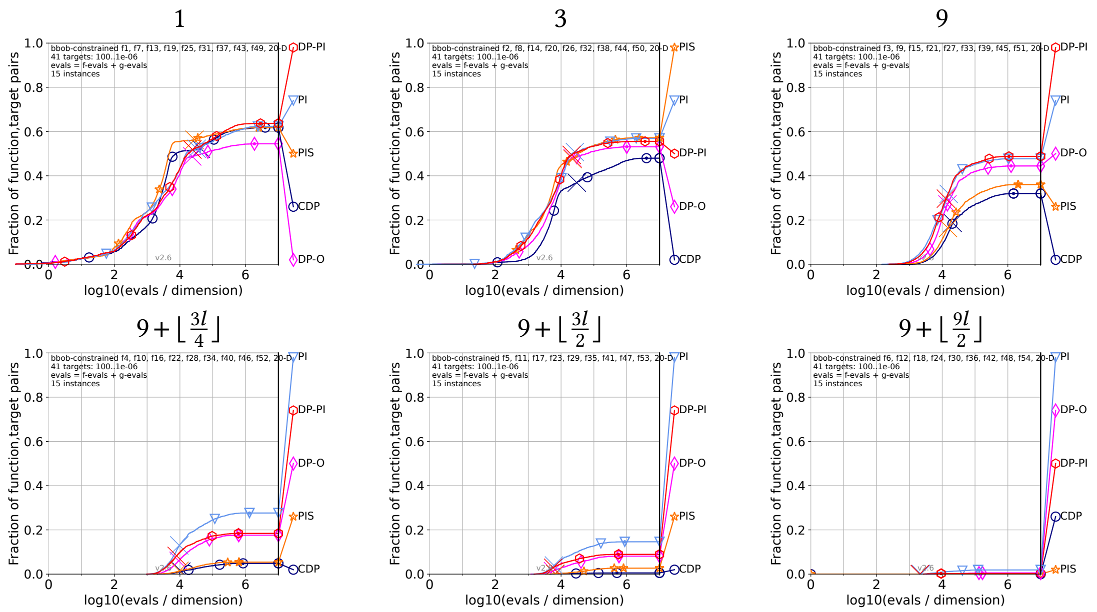


Figure 5.8: The results for the bbob-constrained suite. The 54 functions are grouped by the number of constraints and active constraints, as described in Table 4.2. The number of total constraints is displayed above the figures.

the other methods, CDP tends to take more evaluations and reaches fewer approximation targets. This difference is especially visible for the problems with a single constraint. The crosses in the plots denote that additional approximation targets were not reached by all runs, suggesting that the other approaches reliably get closer to the unconstrained optimum than CDP.

Interestingly, on the 5D problems, the advantage of methods utilizing infeasible solutions is less pronounced. CDP reaches a similar number of approximation targets, as indicated by the crosses being grouped together tightly. Nonetheless, the CDP baseline takes more evaluations to reach the same proportion of approximation targets for multiple problems. This is particularly evident for problems with a high number of constraints, where the line for CDP is visibly offset from the other CHTs until around 80% of the approximation targets are reached.

For the 10D problems, clear differences between the approaches start to surface. Especially for the problems with a dynamic number of constraints, i.e. the lower row in Figure 5.7, CDP needs more evaluations and achieves fewer function-target pairs overall. Compared to the dual-population based approaches, the PIS variant performs worse on problems with more than 9 constraints. This can be seen by both the crosses indicating that PIS reaches fewer approximation targets reliably, as well as the offset of the line that corresponds to an increase in the number of evaluations needed to reach the same approximation targets.

For the results on the 20D problems, the same trends seemingly continue, however, the computational budget is exhausted before optimization converges, leading to incomplete results. CDP reaches fewer approximation targets reliably and takes more evaluations. However, up to the point where the computational budget is exhausted, PIS reaches fewer approximation targets than the other methods on problems with 9 or more constraints. Furthermore, the incomplete results indicate possible performance differences between DP-O and the proposed modifications. PI in particular reaches more approximation targets on all problem groups.

Overall, the proposed methods exhibit similar or noticeably improved performance to the CDP method on the bbob-constrained problems. This shows that the use of infeasible solutions for adjusting the estimated distribution can be effective.

5.3.3 Cone problem

The results on the Cone problem instances are shown in Figure 5.9. Again, CDP and DP-O are included as a reference to the performance of the approaches tested in Chapter 4. In line with the previous results, there is a clear difference in performance between problems with $< 180°$ and $180°$. Furthermore, the proposed methods show clear performance improvement on the problems using concatenated blocks and the Marginal Product FOS.

On the problems where the Full FOS is used, the PIS method tends to perform similarly or better than CDP and shows a visible improvement in performance for some of the $< 180°$ instances. The DP-PI method performs similar to CDP on the $< 180°$ instances, showing a clear improvement over the DP-O method it is based on. However, on the $180°$ instance the method performs worse than DP-O for dimensions larger than 5, with the unsuccessful runs likely being due to exhausting the evaluation budget. Lastly, PI outperforms DP-O on all problems, both in terms of reliably reaching the approximation target and number of evaluations used.

Note that for the problems with decomposition, i.e. where the Marginal Product FOS and repeated blocks of the problem were used, the difference in evaluations compared to not using infeasible solutions indicates a possible change in the convergence behavior. Both PI and DP-PI outperform DP-O, and even CDP for the problems with block size 5 and more than 20 dimensions. PIS shows similar results, outperforming CDP on all problem instances with concatenated blocks. On many problems, PIS reliably solves larger instances than CDP before the computational budget becomes a limiting factor. For instance, on the instances with blocks of size 10, CDP fails to reach the approximation target for the 40D instances within the evaluation budget, while the PIS variants do. Notably, to reach the approximation target of a problem with double the number of decision variables, PIS needs less than double the evaluations per dimension. The transparent regression line in the color of CDP corresponds to quadratic scaling in evaluations w.r.t. dimensions for CDP. The available data for the PIS behavior indicates that the use of infeasible solutions improves this, however, the growth is still larger than linear in the observed range.

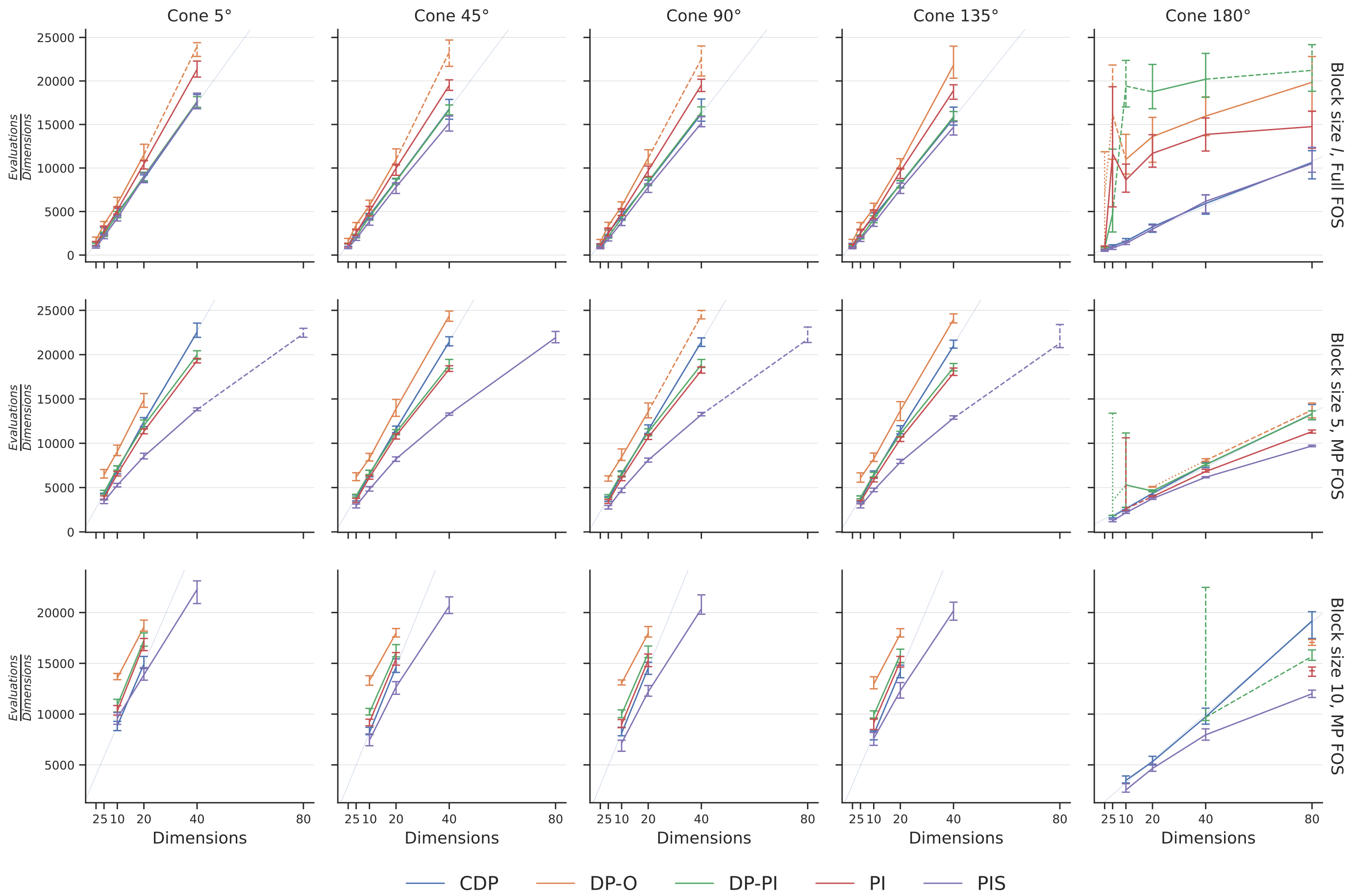


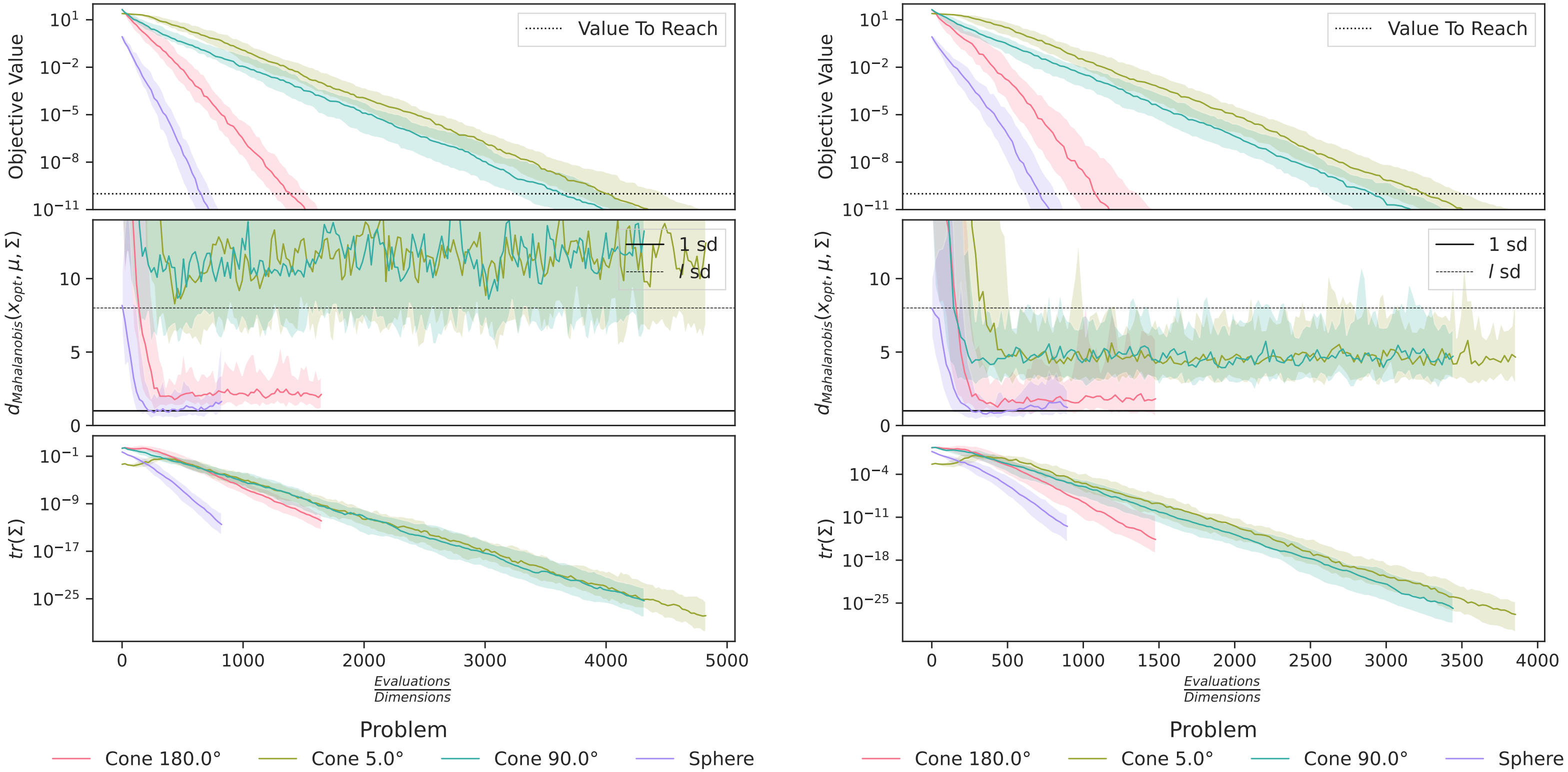
Figure 5.9: The median and interdecile range of the number of evaluations needed to reach the VTR of 10^{-10} on different Cone problem instances over 31 runs. The row corresponds to the block size used, and MP to the Marginal Product FOS. Solid lines indicate a success rate of ≥ 90%, dashed lines between 90% and 50%, dotted lines between 50% and 10% and less successful results are not shown.

5.3.4 Effectiveness

The aim of the proposed methods is to increase the efficiency of search near constraint boundaries by improving the fit of the estimated distributions. To do so, infeasible solutions are selected and thus influence the estimated distribution. In this section, the experiment to determine the search effectiveness is repeated for the proposed methods in order to determine if the performed modifications are effective in this regard.

The results for the PIS approach compared to CDP in 16D are shown in Figure 5.10. Compared to the CDP method, there is a clear difference for the Cone problems with 90° and 5°. The PIS method achieves a lower Mahalanobis distance between the distribution mean and the optimum, and reaches the target value with fewer evaluations. Compared to CDP, the Mahalanobis distance is nearly halved whereas, the effect on the number of evaluations needed is a $\approx 15\%$ decrease. This indicates that the proposed changes are effective. However, the Mahalanobis distance is still large compared to the value for the Sphere problem or Cone problem with 180° feasibility and grows with increased dimensionality. Thus, the problem of never reaching the exploitation phase still persists, albeit to a lesser extent.

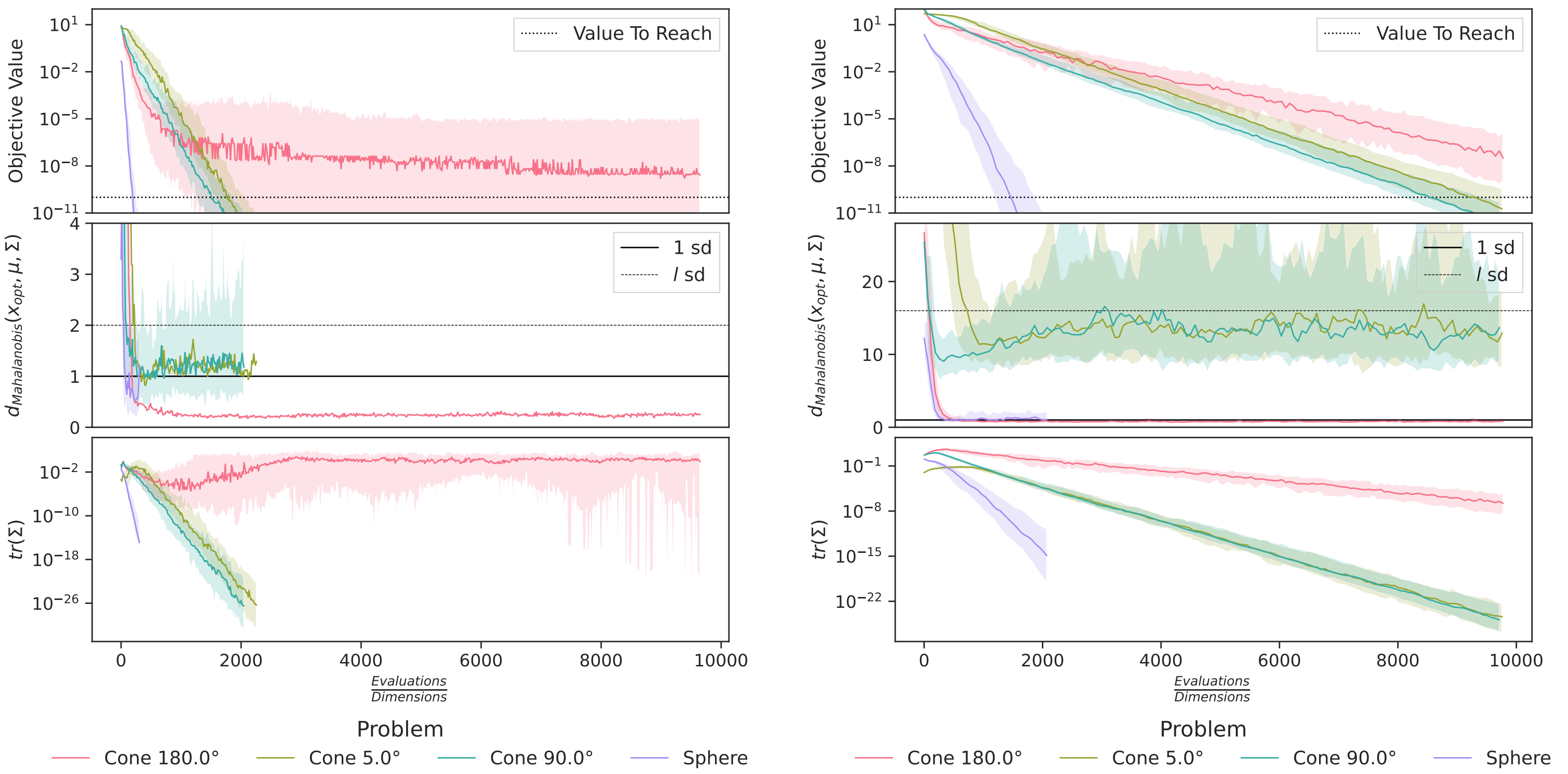
For the DP-O, PI and DP-PI methods, the results are shown in Figures 5.11 to 5.13. Where DP-O struggles with the 180° problem, the proposed modifications lead to reliable success in 2 dimensions. The Mahalanobis distance does not increase, but the trace of the covariance shrinks, explaining this increase in reliability. Both methods also show increased performance on the other constrained problems. However, the effectiveness on the Cone problem with 180° still decreases with increasing dimensionality. In 16D, PI shows a small improvement in the number of evaluations needed over DP-O for the constrained problems. Again, the decrease in the trace of the covariance matrix explains this improvement. On the other hand, the DP-PI method shows worse performance than DP-O on the 180° problem in 16D, as the trace of the covariance matrix decreases slower. It is clear that the spread of the distribution, measured by the trace of the covariance matrix, does not adapt fast enough for the DP-O method and the proposed changes. Nonetheless, both the performance and the Mahalanobis distance improve for the other constrained problems in 16D.



(a) All 8D problems using CDP

(b) All 8D problems using PIS

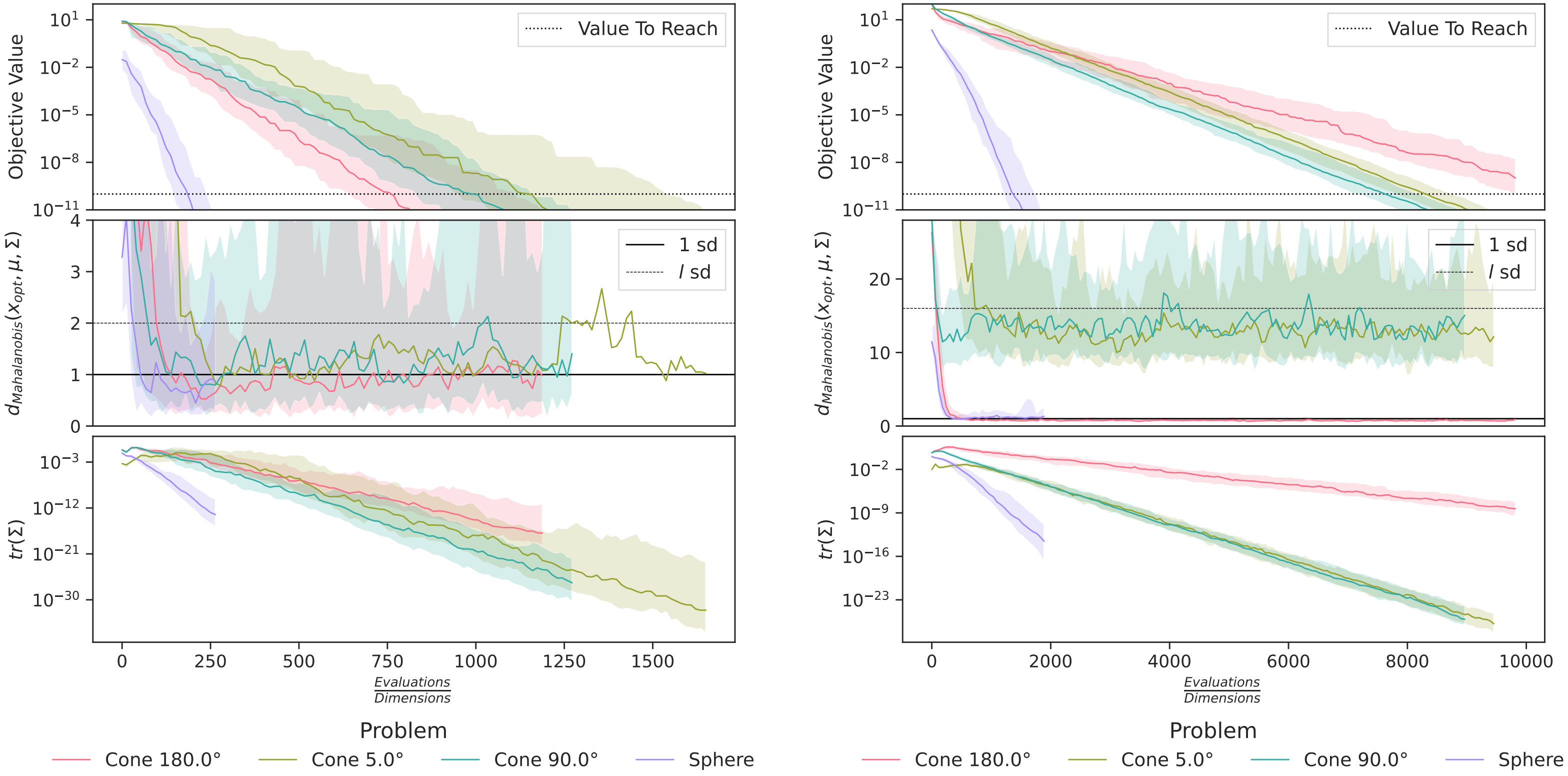
Figure 5.10: The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs.



(a) All 2D problems using DP-O

(b) All 8D problems using DP-O

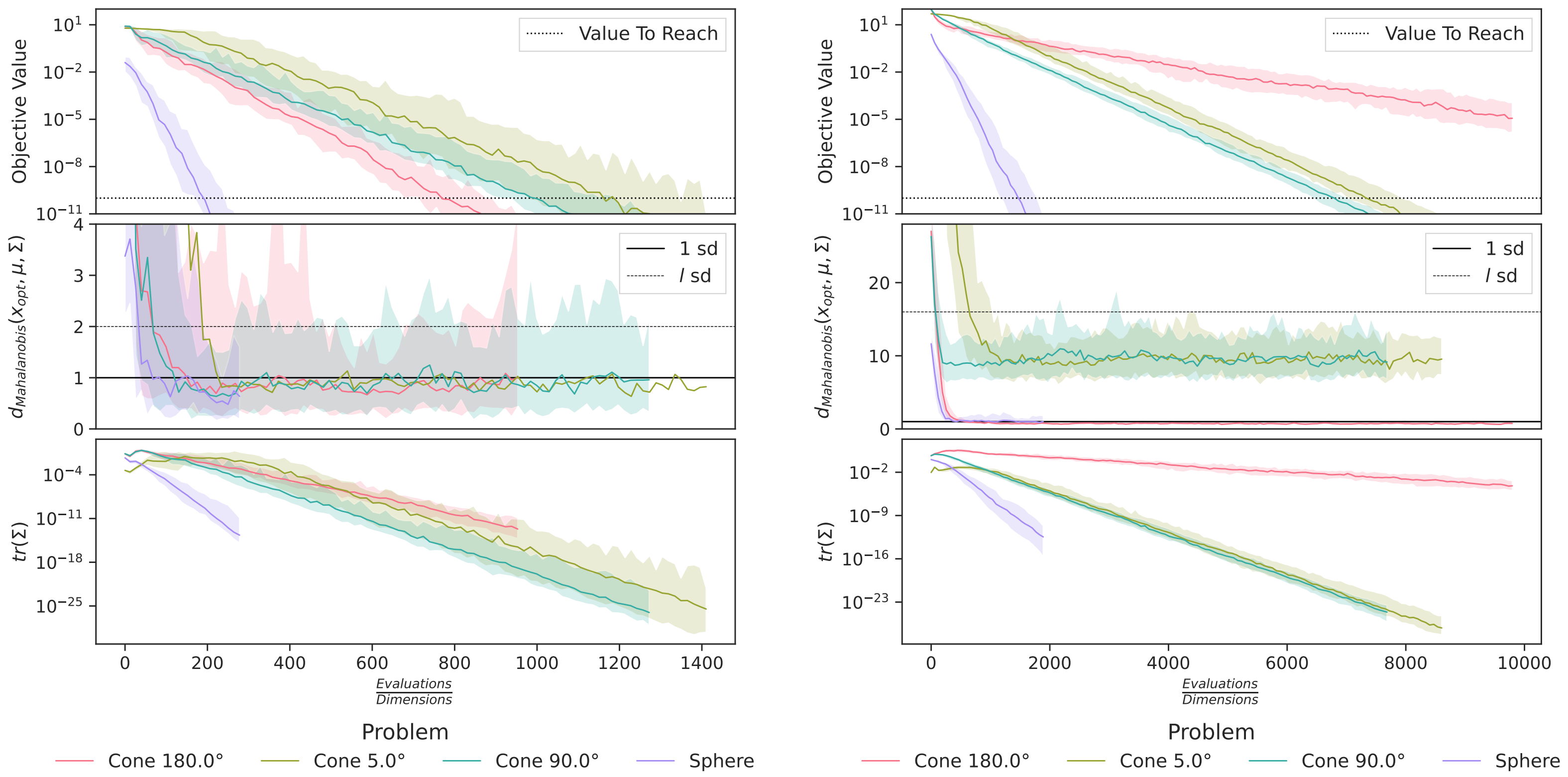
Figure 5.11: The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs.



(a) All 2D problems using PI

(b) All 8D problems using PI

Figure 5.12: The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs.



(a) All 2D problems using DP-PI

(b) All 8D problems using DP-PI

Figure 5.13: The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs.

5.3.5 Summary

Overall, the results show that the proposed methods can be effective for problems where the optimum is located at constraint boundaries, in particular the problems found in the bbob-constrained test suite. By using infeasible solutions, the search bias of feasibility oriented methods can be reduced, answering the third research question. This benchmark suite consists of problems where the optimization starts in the feasible region, removing the need to first find the feasible region. Furthermore, the proposed changes lead to clear gains in effectiveness on the Cone problems with decomposition.

Chapter 6

Discussion

In the previous chapters, various CHTs have been implemented and tested with RV-GOMEA. With respect to the different desired properties of a CHT, the results are mixed. The different approaches either do not generalize well and require parameter tuning or lack in performance compared to the best methods on a given problem. However, when considering the available trade-offs and the currently used technique CDP, the results are promising. In this chapter, the limitations of the work performed in this thesis are discussed.

Importance of Boundary Search The focus of the proposed methods was on improving the search close to constraint boundaries. This is reasonable, considering that this holds true for many real world problems [31]. However, taking the search for the feasible region and the desired approximation level into account, the importance of boundary search varies. With increased precision requirements, the importance of boundary search increases, since the proportion of evaluations spent approximating the constraint boundary grows. This can be observed for the benchmark suites used, where the main factor of differentiation between the various CHTs is the search in the infeasible region for the CEC2006 problems and search in the feasible region for bbob-constrained.

Lack of Representation The field of constraint handling for evolutionary algorithms is very active[50]. Therefore, the existing techniques used in this thesis are by necessity not an accurate and complete representation of the current state-of-the-art. In particular, there is recent work on novel ranking [15, 16] and dual-population schemes [76, 2, 34]. Additionally, no hybrid or ensemble methods were included.

Hyperparameter Settings Currently the parameters of the various approaches were chosen based on the CEC2006 benchmark suite, or the recommended parameter settings were used. This was done since the cost of performing hyperparameter tuning and determining the best population size on all problems used was intractable during this thesis. Because of this, the results obtained may not be the best possible results across all problems for some of the CHTs used.

Implementation and Adaption The existing techniques first proposed for different EAs needed some adaption to work with RV-GOMEA. By doing so, there is the risk of not representing the prior work wrongly and making choices that negatively affect the obtained performance. To mitigate this, multiple ways of adapting the techniques were considered and the best performing method was chosen. Another limitation is that all CHTs were implemented based on a single constraint violation value, while some approaches such as the Augmented Lagrangian or ϵ-Constrained methods can be extended to handle individual constraints.

Limited Testing of FOS Structure and GBO Scenarios RV-GOMEA is an EA designed for large-scale GBO optimization. Such a scenario has not been tested in this thesis due to a lack of benchmark problems that allow for a GBO approach with partial evaluations or decomposition. While the Marginal Product FOS was used for the Cone problem, all experiments were performed in a BBO setting.

Incomplete bbob-constrained Results The bbob-constrained benchmark suite includes the 54 problems in 2,5,10,20 and 40 dimensions. The computational budget used included a 60 second time limit per problem to make running the experiment for all CHTs and the different dimensions tractable. However, for the 20 and 40 dimensional problems, this turned out to be too little, as the optimization progress towards the approximation targets did not stop before the budget was exhausted. Therefore, the results obtained can not be considered conclusive, as a larger budget could change the results.

Lack of testing on real-world problems Recent literature indicates that the performance CHTs show on synthetic benchmarks does not necessarily translate to real-world problems [49]. Therefore, it is recommended to also use practical problems to compare the different CHTs. However, this is not common in the existing literature yet. To be able to compare with other constraint optimization algorithms, this thesis uses problems typically found in literature.

Chapter 7

Conclusion and Future Work

In this thesis, various existing CHTs have been adapted to RV-GOMEA and evaluated on several benchmark problems. Using the insights gained, modifications and a novel approach have been introduced and evaluated, showing promising performance in some aspects. In this chapter, conclusions are drawn from the work performed and potential directions for future work are identified.

7.1 Conclusion

The performance of an approach tends to differ greatly between problems where the constrained optimum is located at the intersection of multiple constraints and problems with a single active constraint. Furthermore, testing on the synthetic Cone problem has shown that some approaches such as AL perform clearly worse, when the feasible region connected to the constrained optimum is small, while other methods such as CDP are less affected. This loss in effectiveness is explained by the estimated distributions, where the solutions at the constraint boundary tend to be increasingly out-of-distribution with increasing problem dimension.

Furthermore, no CHT showed competitive performance across the different problems and scenarios used to evaluate the approaches, corroborating the assumption that a single CHT cannot possibly be suited for all possible problems. Nonetheless, some approaches showed good performance across a range of problems. While manual selection of a CHT is still required, the range of problems that can be solved with RV-GOMEA has increased through the addition of new techniques. CDP, the currently used technique, shows broad applicability, but for all problems there is an approach performing equal or better. On the bbob-constrained problems, DP-O and the proposed methods perform well and reach considerably more approximation targets than the CDP baseline. Other approaches such as AL perform well on individual problems but do not generalize. Overall, SR and PIS show increased performance and reliability across the different problems tested, performing better or equal to CDP on almost all problems.

The proposed modifications show promising initial results for improving the effectiveness of search in the feasible region near constraint boundaries. PIS shows competitive results with CDP, while performing better on the bbob-benchmark suite and the Cone problems, especially in combination with decomposition and the Marginal Product FOS. PI and DP-PI, the proposed modifications to DP-O show increased performance on some problems, however, they introduce new trade-offs. For example, using a single distribution is more effective for problems with a single active constraint, but less reliable overall. Overall, the results are promising and confirm that making use of infeasible solutions during search can improve the optimization efficiency.

Finally, as no constrained benchmark problems supporting a GBO approach exist in literature, a new problem was introduced supporting this setting. When considering the initial results obtained with the Marginal Product FOS, it is clear that it is possible for CHTs to exploit this setting. However, only the proposed methods show this behavior, with some CHTs such as the repair methods also performing worse. However, what makes an approach work better or worse in this setting is not analyzed in this thesis.

7.2 Future Work

In this section, several possible directions for future work are outlined.

Constraint Aggregation and Equality Constraints In this thesis, the constraint values were aggregated into a single constraint value, a common way of handling multiple constraints when using EAs. However, this is not ideal for problems where the constraint value differs in scale between multiple constraints or when equality constraints are present. Future work in this direction could incorporate techniques to better handle equality constraints and constraints values of different scales into RV-GOMEA.

Other Techniques While this thesis aims to provide an overview of how different CHTs perform when combined with RV-GOMEA, the research field is active, and exploring all approaches is not feasible. Future work could look at approaches not considered in this thesis. In particular, one focus of current research is a new form of dual-population approach, where two CHTs are combined. Instead of approaching the constraint boundary from both sides, these methods aim to combine the advantages of fast but unreliable CHTs with feasibility oriented techniques [2, 34, 75]. Another research area could be portfolio and ensemble methods, possibly making use of the CHTs described in this thesis.

Decomposition and GBO In this thesis, initial results suggest that scenarios with partial modifications can be effectively exploited with respect to constraint handling. As RV-GOMEA excels at large-scale GBO optimization, further work could be performed towards better understanding and improving performance in this setting. The methods examined in this thesis could serve as a starting point. Approaches tailored towards this scenario likely would need to examine how problems with constraints can be decomposed and perform constraint handling per FOS subset. In addition, more realistic problems supporting a GBO approach are needed.

Multi-Objective Optimization In this thesis, the focus was on single-objective optimization. However, many real-world problems have multiple conflicting objectives. Hence, further work could investigate constraint handling for the multi-objective variant of RV-GOMEA, possibly leading to performance improvements for real-world use cases.

Bibliography

[1] Georgios Andreadis, Peter A. N. Bosman, and Tanja Alderliesten. Multi-objective dual simplex-mesh based deformable image registration for 3d medical images - proof of concept. In Ivana Išgum and Olivier Colliot, editors, *Medical Imaging 2022: Image Processing*. SPIE, April 2022. doi: 10.1117/12.2605498. URL https://doi.org/10.1117/12.2605498.

[2] Qian Bao, Maocai Wang, Guangming Dai, Xiaoyu Chen, Zhiming Song, and Shuijia Li. A dual-population based bidirectional coevolution algorithm for constrained multi-objective optimization problems. *Expert Systems with Applications*, 215:119258, April 2023. doi: 10.1016/j.eswa.2022.119258. URL https://doi.org/10.1016/j.eswa.2022.119258.

[3] Helio J. C. Barbosa, Afonso C. C. Lemonge, and Heder S. Bernardino. A critical review of adaptive penalty techniques in evolutionary computation. In *Infosys Science Foundation Series*, pages 1–27. Springer India, December 2014. doi: 10.1007/978-81-322-2184-5_1. URL https://doi.org/10.1007/978-81-322-2184-51.

[4] Danique L.J. Barten, Bradley R. Pieters, Anton Bouter, Marjolein C. van der Meer, Stef C. Maree, Karel A. Hinnen, Henrike Westerveld, Peter A.N. Bosman, Tanja Alderliesten, Niek van Wieringen, and Arjan Bel. Towards artificial intelligence-based automated treatment planning in clinical practice: A prospective study of the first clinical experiences in high-dose-rate prostate brachytherapy. *Brachytherapy*, 22(2):279–289, March 2023. doi: 10.1016/j.brachy.2022.11.013. URL https://doi.org/10.1016/j.brachy.2022.11.013.

[5] Peter A. N. Bosman, Jörn Grahl, and Franz Rothlauf. SDR. In *Proceedings of the 9th annual conference on Genetic and evolutionary computation*. ACM, July 2007. doi: 10.1145/1276958.1277066. URL https://doi.org/10.1145/1276958.1277066.

[6] Peter A. N. Bosman, Jörn Grahl, and Dirk Thierens. Enhancing the performance of maximum–likelihood gaussian EDAs using anticipated mean shift. In *Parallel*

Problem Solving from Nature – PPSN X, pages 133–143. Springer Berlin Heidelberg, 2008. doi: 10.1007/978-3-540-87700-4_14. URL https://doi.org/10.1007/978-3-540-87700-414.

[7] Peter A. N. Bosman, Jörn Grahl, and Dirk Thierens. Benchmarking parameter-free AMaLGaM on functions with and without noise. *Evolutionary Computation*, 21(3): 445–469, September 2013. doi: 10.1162/evco_a_00094. URL https://doi.org/10.1162/evcoa00094.

[8] Anton Bouter. *Optimal Mixing Evolutionary Algorithms for Large-Scale Real-Valued Optimization: Including Real-World Medical Applications*. PhD thesis, Delft University of Technology, 2023.

[9] Anton Bouter, Tanja Alderliesten, Cees Witteveen, and Peter A. N. Bosman. Exploiting linkage information in real-valued optimization with the real-valued gene-pool optimal mixing evolutionary algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, July 2017. doi: 10.1145/3071178.3071272. URL https://doi.org/10.1145/3071178.3071272.

[10] Janez Brest. Constrained real-parameter optimization with ϵ-self-adaptive differential evolution. In *Constraint-Handling in Evolutionary Optimization*, pages 73–93. Springer Berlin Heidelberg, 2009. doi: 10.1007/978-3-642-00619-7_4. URL https://doi.org/10.1007/978-3-642-00619-74.

[11] Sébastien Bubeck. Convex optimization: Algorithms and complexity. 2014. doi: 10.48550/ARXIV.1405.4980. URL https://arxiv.org/abs/1405.4980.

[12] Bodhi Buurman, Konstantinos Gkiotsalitis, and Eric C. van Berkum. Railway maintenance reservation scheduling considering detouring delays and maintenance demand. *Journal of Rail Transport Planning & Management*, 25:100359, March 2023. doi: 10.1016/j.jrtpm.2022.100359. URL https://doi.org/10.1016/j.jrtpm.2022.100359.

[13] Carlos A Coello Coello. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering*, 191(11-12):1245–1287, January 2002. doi: 10.1016/s0045-7825(01)00323-1. URL https://doi.org/10.1016/s0045-7825(01)00323-1.

[14] Carlos A. Coello Coello. Constraint-handling techniques used with evolutionary algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. ACM, July 2022. doi: 10.1145/3520304.3533640. URL https://doi.org/10.1145/3520304.3533640.

[15] Max de Castro Rodrigues, Beatriz Souza Leite Pires de Lima, and Solange Guimarães. Balanced ranking method for constrained optimization problems using evolutionary algorithms. *Information Sciences*, 327:71–90, January 2016. doi: 10.1016/j.ins.2015.08.012. URL https://doi.org/10.1016/j.ins.2015.08.012.

[16] Rafael de Paula Garcia, Beatriz Souza Leite Pires de Lima, Afonso Celso de Castro Lemonge, and Breno Pinheiro Jacob. A rank-based constraint handling technique for engineering design optimization problems solved by genetic algorithms. *Computers &amp Structures*, 187:77–87, July 2017. doi: 10.1016/j.compstruc.2017.03.023. URL https://doi.org/10.1016/j.compstruc.2017.03.023.

[17] Kalyanmoy Deb. An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*, 186(2-4):311–338, June 2000. doi: 10.1016/s0045-7825(99)00389-8. URL https://doi.org/10.1016/s0045-7825(99)00389-8.

[18] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002. doi: 10.1109/4235.996017.

[19] Paul Dufossé and Asma Atamna. Benchmarking several strategies to update the penalty parameters in al-cma-es on the bbob-constrained testbed. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '22, page 1691–1699, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392686. doi: 10.1145/3520304.3534014. URL https://doi.org/10.1145/3520304.3534014.

[20] Paul Dufossé and Nikolaus Hansen. Augmented lagrangian, penalty techniques and surrogate modeling for constrained optimization with CMA-ES. In *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, June 2021. doi: 10.1145/3449639.3459340. URL https://doi.org/10.1145/3449639.3459340.

[21] Arkadiy Dushatskiy, Marco Virgolin, Anton Bouter, Dirk Thierens, and Peter A. N. Bosman. Parameterless gene-pool optimal mixing evolutionary algorithms, 2021. URL https://arxiv.org/abs/2109.05259.

[22] Zhun Fan, Hui Li, Caimin Wei, Wenji Li, Han Huang, Xinye Cai, and Zhaoquan Cai. An improved epsilon constraint handling method embedded in moea/d for constrained multi-objective optimization problems. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8, 2016. doi: 10.1109/SSCI.2016.7850224.

[23] Jenny Freeman and Michael Campbell. The analysis of categorical data: Fisher's exact test. *Scope*, 16, 01 2007.

[24] Cheng gang Cui, Yan jun Li, and Tie jun Wu. A relative feasibility degree based approach for constrained optimization problems. *Journal of Zhejiang University SCIENCE C*, 11(4):249–260, April 2010. doi: 10.1631/jzus.c0910072. URL https://doi.org/10.1631/jzus.c0910072.

[25] Wei-Feng Gao, Gary G. Yen, and San-Yang Liu. A dual-population differential evolution with coevolution for constrained optimization. *IEEE Transactions on Cybernetics*, 45(5):1108–1121, 2015. doi: 10.1109/TCYB.2014.2345478.

[26] Roland Glowinski and Patrick Le Tallec. *Augmented Lagrangian and Operator Splitting Methods in Nonlinear Mechanics.* Society for Industrial and Applied Mathematics (SIAM), January 1989. ISBN 978-0-898712-30-8. doi: 10.1137/1.9781611970838. URL `https://doi.org/10.1137/1.9781611970838`.

[27] Sana Ben Hamida and Marc Schoenauer. ASCHEA: new results using adaptive segregational constraint handling. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*, volume 1, pages 884–889 vol.1. IEEE, 2002. doi: 10.1109/CEC.2002.1007042. URL `https://doi.org/10.1109/cec.2002.1007042`.

[28] Nikolaus Hansen, Anne Auger, Dimo Brockhoff, Dejan Tušar, and Tea Tušar. Coco: Performance assessment, 2016. URL `https://arxiv.org/abs/1605.03560`.

[29] Nikolaus Hansen, Anne Auger, Raymond Ros, Olaf Mersmann, Tea Tušar, and Dimo Brockhoff. COCO: a platform for comparing continuous optimizers in a black-box setting. *Optimization Methods and Software*, 36(1):114–144, August 2020. doi: 10.1080/10556788.2020.1808977. URL `https://doi.org/10.1080/10556788.2020.1808977`.

[30] Gregory Hornby, Al Globus, Derek Linden, and Jason Lohn. Automated antenna design with evolutionary algorithms. In *Space 2006*. American Institute of Aeronautics and Astronautics, September 2006. doi: 10.2514/6.2006-7242. URL `https://doi.org/10.2514/6.2006-7242`.

[31] Hisao Ishibuchi, Takefumi Fukase, Naoki Masuyama, and Yusuke Nojima. Dual-grid model of moea/d for evolutionary constrained multiobjective optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '18, page 665–672, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450356183. doi: 10.1145/3205455.3205644. URL `https://doi-org.tudelft.idm.oclc.org/10.1145/3205455.3205644`.

[32] Guanbo Jia, Yong Wang, Zixing Cai, and Yaochu Jin. An improved $(\mu+\lambda)$-constrained differential evolution for constrained optimization. *Information Sciences*, 222:302–322, February 2013. doi: 10.1016/j.ins.2012.01.017. URL `https://doi.org/10.1016/j.ins.2012.01.017`.

[33] Alexander H.G. Rinnooy Kan and Jan Telgen. The complexity of linear programming. *Statistica Neerlandica*, 35(2):91–107, June 1981. doi: 10.1111/j.1467-9574.1981.tb00717.x. URL `https://doi.org/10.1111/j.1467-9574.1981.tb00717.x`.

[34] Xiangsong Kong, Yongkuan Yang, Zhisheng Lv, Jing Zhao, and Rong Fu. A dynamic dual-population co-evolution multi-objective evolutionary algorithm for constrained multi-objective optimization problems. *Applied Soft Computing*, 141:110311, July

2023. doi: 10.1016/j.asoc.2023.110311. URL https://doi.org/10.1016/j.asoc.2023.110311.

[35] Oliver Kramer. A review of constraint-handling techniques for evolution strategies. *Applied Computational Intelligence and Soft Computing*, 2010:1–11, 2010. doi: 10.1155/2010/185063. URL https://doi.org/10.1155/2010/185063.

[36] Guillermo Leguizamon and Carlos A. Coello Coello. Boundary search for constrained numerical optimization problems with an algorithm inspired by the ant colony metaphor. *IEEE Transactions on Evolutionary Computation*, 13(2):350–368, April 2009. doi: 10.1109/tevc.2008.926731. URL https://doi.org/10.1109/tevc.2008.926731.

[37] Shuang Li, Ke Li, and Wei Li. Do we really need to use constraint violation in constrained evolutionary multi-objective optimization?, 2022. URL https://arxiv.org/abs/2205.14349.

[38] Jing Liang, Thomas Runarsson, Efrén Mezura-Montes, Maurice Clerc, Ponnuthurai Suganthan, Carlos Coello, and Kalyan Deb. Problem definitions and evaluation criteria for the cec 2006 special session on constrained real-parameter optimization. *Nangyang Technological University, Singapore, Tech. Rep*, 41, 01 2006.

[39] Yuqiao Liu, Yanan Sun, Bing Xue, Mengjie Zhang, Gary G. Yen, and Kay Chen Tan. A survey on evolutionary neural architecture search. 2020. doi: 10.48550/ARXIV.2008.10937. URL https://arxiv.org/abs/2008.10937.

[40] Roy J.E. De Maesschalck, Delphine Jouan-Rimbaud, and Désiré Luc Massart. The mahalanobis distance. *Chemometrics and Intelligent Laboratory Systems*, 50(1):1–18, January 2000. doi: 10.1016/s0169-7439(99)00047-7. URL https://doi.org/10.1016/s0169-7439(99)00047-7.

[41] Rammohan Mallipeddi and Ponnuthurai Nagaratnam Suganthan. Differential evolution with ensemble of constraint handling techniques for solving CEC 2010 benchmark problems. In *IEEE Congress on Evolutionary Computation*. IEEE, July 2010. doi: 10.1109/cec.2010.5586330. URL https://doi.org/10.1109/cec.2010.5586330.

[42] Henry B. Mann and Donald R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics*, 18(1):50–60, 1947. ISSN 00034851. URL http://www.jstor.org/stable/2236101.

[43] Efrén Mezura-Montes and Carlos A. Coello Coello. Constraint-handling in nature-inspired numerical optimization: Past, present and future. *Swarm and Evolutionary Computation*, 1(4):173–194, December 2011. doi: 10.1016/j.swevo.2011.10.001. URL https://doi.org/10.1016/j.swevo.2011.10.001.

[44] Efrén Mezura-Montes and Carlos A. Coello Coello. A simple multimembered evolution strategy to solve constrained optimization problems. *IEEE Transactions on Evolutionary Computation*, 9(1):1–17, 2005. doi: 10.1109/TEVC.2004.836819.

[45] Zbigniew Michalewicz. A Survey of Constraint Handling Techniques in Evolutionary Computation Methods. In *Evolutionary Programming IV: Proceedings of the Fourth Annual Conference on Evolutionary Programming*. The MIT Press, 08 1995. ISBN 9780262290920. doi: 10.7551/mitpress/2887.003.0018. URL https://doi.org/10.7551/mitpress/2887.003.0018.

[46] Mengjun Ming, Anupam Trivedi, Rui Wang, Dipti Srinivasan, and Tao Zhang. A dual-population-based evolutionary algorithm for constrained multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 25(4):739–753, August 2021. doi: 10.1109/tevc.2021.3066301. URL https://doi.org/10.1109/tevc.2021.3066301.

[47] Mengjun Ming, Rui Wang, Hisao Ishibuchi, and Tao Zhang. A novel dual-stage dual-population evolutionary algorithm for constrained multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 26(5):1129–1143, 2022. doi: 10.1109/TEVC.2021.3131124.

[48] Jan Paredis. Co-evolutionary constraint satisfaction. In *Parallel Problem Solving from Nature — PPSN III*, pages 46–55. Springer Berlin Heidelberg, 1994. doi: 10.1007/3-540-58484-6_249. URL https://doi.org/10.1007/3-540-58484-6249.

[49] Cyril Picard and Jürg Schiffmann. Realistic constrained multiobjective optimization benchmark problems from design. *IEEE Transactions on Evolutionary Computation*, 25(2):234–246, 2021. doi: 10.1109/TEVC.2020.3020046.

[50] Iman Rahimi, Amir H. Gandomi, Fang Chen, and Efrén Mezura-Montes. A review on constraint handling techniques for population-based algorithms: from single-objective to multi-objective optimization. *Archives of Computational Methods in Engineering*, 30(3):2181–2209, December 2022. doi: 10.1007/s11831-022-09859-9. URL https://doi.org/10.1007/s11831-022-09859-9.

[51] Tapabrata Ray, Hemant Kumar Singh, Amitay Isaacs, and Warren Smith. Infeasibility driven evolutionary algorithm for constrained optimization. In *Constraint-Handling in Evolutionary Optimization*, pages 145–165. Springer Berlin Heidelberg, 2009. doi: 10.1007/978-3-642-00619-7_7. URL https://doi.org/10.1007/978-3-642-00619-77.

[52] Rodolphe Le Riche, Catherine Knopf-Lenoir, and Raphael T. Haftka. A segregated genetic algorithm for constrained structural optimization. In *Proceedings of the 6th International Conference on Genetic Algorithms*, page 558–565, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. ISBN 1558603700.

[53] Marco M. Rosso, Raffaele Cucuzza, Fabio Di Trapani, and Giuseppe C. Marano. Non-penalty machine learning constraint handling using PSO-SVM for structural optimization. *Advances in Civil Engineering*, 2021:1–17, February 2021. doi: 10.1155/2021/6617750. URL https://doi.org/10.1155/2021/6617750.

[54] Thomas P. Runarsson and Xin Yao. Stochastic ranking for constrained evolutionary optimization. *IEEE Transactions on Evolutionary Computation*, 4(3):284–294, 2000. doi: 10.1109/4235.873238. URL https://doi.org/10.1109/4235.873238.

[55] Sancho Salcedo-Sanz. A survey of repair methods used as constraint handling techniques in evolutionary algorithms. *Computer Science Review*, 3(3):175–192, August 2009. doi: 10.1016/j.cosrev.2009.07.001. URL https://doi.org/10.1016/j.cosrev.2009.07.001.

[56] Carlos Segura, Carlos A. Coello Coello, Gara Miranda, and Coromoto León. Using multi-objective evolutionary algorithms for single-objective constrained and unconstrained optimization. *Annals of Operations Research*, 240(1):217–250, September 2015. doi: 10.1007/s10479-015-2017-z. URL https://doi.org/10.1007/s10479-015-2017-z.

[57] Abolfazl Shirazi, Josu Ceberio, and Jose A. Lozano. EDA++: Estimation of distribution algorithms with feasibility conserving mechanisms for constrained continuous optimization. *IEEE Transactions on Evolutionary Computation*, 26(5):1144–1156, October 2022. doi: 10.1109/tevc.2022.3153933. URL https://doi.org/10.1109/tevc.2022.3153933.

[58] Petru A. Simionescu, David G. Beale, and Gerry V. Dozier. Constrained optimization problem solving using estimation of distribution algorithms. In *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753)*, volume 1, pages 296–302 Vol.1. IEEE, 2004. doi: 10.1109/CEC.2004.1330870. URL https://doi.org/10.1109/cec.2004.1330870.

[59] Adam Slowik and Halina Kwasnicka. Evolutionary algorithms and their applications to engineering problems. *Neural Computing and Applications*, 32(16):12363–12379, March 2020. doi: 10.1007/s00521-020-04832-8. URL https://doi.org/10.1007/s00521-020-04832-8.

[60] Vladimir Stanovov, Shakhnaz Akhmedova, and Eugene Semenkin. Combined fitness–violation epsilon constraint handling for differential evolution. *Soft Computing*, 24(10):7063–7079, March 2020. doi: 10.1007/s00500-020-04835-6. URL https://doi.org/10.1007/s00500-020-04835-6.

[61] Tetsuyuki Takahama and Setsuko Sakai. Constrained optimization by the ϵ constrained differential evolution with gradient-based mutation and feasible elites. In *2006 IEEE International Conference on Evolutionary Computation*. IEEE, 2006. doi: 10.1109/cec.2006.1688283. URL https://doi.org/10.1109/cec.2006.1688283.

[62] Tetsuyuki Takahama and Setsuko Sakai. Constrained optimization by the ϵ constrained differential evolution with an archive and gradient-based mutation. In *IEEE Congress on Evolutionary Computation*. IEEE, July 2010. doi: 10.1109/cec.2010.5586484. URL https://doi.org/10.1109/cec.2010.5586484.

[63] Tetsuyuki Takahama and Setsuko Sakai. Efficient constrained optimization by the ϵ constrained differential evolution with rough approximation. In *Infosys Science Foundation Series*, pages 157–180. Springer India, December 2014. doi: 10.1007/978-81-322-2184-5_6. URL https://doi.org/10.1007/978-81-322-2184-56.

[64] Tetsuyuki Takahama, Setsuko Sakai, and Noriyuki Iwane. Solving Nonlinear Constrained Optimization Problems by the ϵ Constrained Differential Evolution. In *2006 IEEE International Conference on Systems, Man and Cybernetics*. IEEE, October 2006. doi: 10.1109/icsmc.2006.385209. URL https://doi.org/10.1109/icsmc.2006.385209.

[65] Dirk Thierens. Scalability problems of simple genetic algorithms. *Evolutionary Computation*, 7(4):331–352, December 1999. doi: 10.1162/evco.1999.7.4.331. URL https://doi.org/10.1162/evco.1999.7.4.331.

[66] Marco Virgolin. *Design and Application of Gene-pool Optimal Mixing Evolutionary Algorithms for Genetic Programming*. PhD thesis, 2020. URL http://resolver.tudelft.nl/uuid:03641b5f-f8f6-4ff9-be7f-11948f6d3cc7.

[67] Marco Virgolin and Solon P. Pissis. Symbolic regression is np-hard, 2022. URL https://arxiv.org/abs/2207.01018.

[68] Marco Virgolin, Tanja Alderliesten, Arjan Bel, Cees Witteveen, and Peter A. N. Bosman. Symbolic regression and feature construction with GP-GOMEA applied to radiotherapy dose reconstruction of childhood cancer survivors. In *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, July 2018. doi: 10.1145/3205455.3205604. URL https://doi.org/10.1145/3205455.3205604.

[69] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

[70] Yanping Wang, Yuan Liu, Juan Zou, Jinhua Zheng, and Shengxiang Yang. A novel two-phase evolutionary algorithm for solving constrained multi-objective optimization problems. *Swarm and Evolutionary Computation*, 75:101166, December 2022. doi: 10.1016/j.swevo.2022.101166. URL https://doi.org/10.1016/j.swevo.2022.101166.

[71] Yong Wang, Zixing Cai, Yuren Zhou, and Wei Zeng. An adaptive tradeoff model for constrained evolutionary optimization. *IEEE Transactions on Evolutionary Computation*, 12(1):80–92, February 2008. doi: 10.1109/tevc.2007.902851. URL https://doi.org/10.1109/tevc.2007.902851.

[72] Yong Wang, Jia-Peng Li, Xihui Xue, and Bing chuan Wang. Utilizing the correlation between constraints and objective function for constrained evolutionary optimization. *IEEE Transactions on Evolutionary Computation*, 24(1):29–43, February 2020. doi: 10.1109/tevc.2019.2904900. URL https://doi.org/10.1109/tevc.2019.2904900.

[73] David H. Wolpert and William G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, April 1997. doi: 10.1109/4235.585893. URL https://doi.org/10.1109/4235.585893.

[74] Tao Xu, Jun He, and Changjing Shang. Helper and equivalent objectives: An efficient approach to constrained optimisation. *IEEE Transactions on Cybernetics*, 2019. doi: 10.48550/ARXIV.1903.04886. URL https://arxiv.org/abs/1903.04886.

[75] Kaixi Yang, Jinhua Zheng, Juan Zou, Fan Yu, and Shengxiang Yang. A dual-population evolutionary algorithm based on adaptive constraint strength for constrained multi-objective optimization. *Swarm and Evolutionary Computation*, 77:101247, March 2023. doi: 10.1016/j.swevo.2023.101247. URL https://doi.org/10.1016/j.swevo.2023.101247.

[76] Kaixi Yang, Jinhua Zheng, Juan Zou, Fan Yu, and Shengxiang Yang. A dual-population evolutionary algorithm based on adaptive constraint strength for constrained multi-objective optimization. *Swarm and Evolutionary Computation*, 77:101247, March 2023. doi: 10.1016/j.swevo.2023.101247. URL https://doi.org/10.1016/j.swevo.2023.101247.

[77] Amir Yazdanbakhsh, Christof Angermueller, Berkin Akin, Yanqi Zhou, Albin Jones, Milad Hashemi, Kevin Swersky, Satrajit Chatterjee, Ravi Narayanaswami, and James Laudon. Apollo: Transferable architecture exploration, 2021. URL https://arxiv.org/abs/2102.01723.

[78] Yang Yu and Zhi-Hua Zhou. On the usefulness of infeasible solutions in evolutionary search: A theoretical study. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*. IEEE, June 2008. doi: 10.1109/cec.2008.4630893. URL https://doi.org/10.1109/cec.2008.4630893.

[79] Chunjiang Zhang, A. K. Qin, Weiming Shen, Liang Gao, Kay Chen Tan, and Xinyu Li. ϵ-constrained differential evolution using an adaptive ϵ-level control method. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52(2):769–785, February 2022. doi: 10.1109/tsmc.2020.3010120. URL https://doi.org/10.1109/tsmc.2020.3010120.

[80] Xinggang Zhao, Kai Zhang, Guodong Chen, Xiaoming Xue, Chuanjin Yao, Jian Wang, Yongfei Yang, Hui Zhao, and Jun Yao. Surrogate-assisted differential evolution for production optimization with nonlinear state constraints. *Journal of Petroleum Science and Engineering*, 194:107441, November 2020. doi: 10.1016/j.petrol.2020.107441. URL https://doi.org/10.1016/j.petrol.2020.107441.

[81] Karin Zielinski, Shyam Praveen Vudathu, and Rainer Laur. Influence of different deviations allowed for equality constraints on particle swarm optimization and differential evolution. In *Nature Inspired Cooperative Strategies for Optimization (NICSO 2007)*, pages 249–259. Springer Berlin Heidelberg, 2008. doi: 10.1007/978-3-540-78987-1_23. URL https://doi.org/10.1007/978-3-540-78987-123.

Appendix A

Parameter Configuration

The existing CHTs used in this thesis come with additional parameters, which need to be set correctly in order to achieve the best performance possible. While the need for parameter tuning is not desirable, a fair comparison should ensure that the different approaches use recommended parameter settings or tune the parameters.

In the following sections, the method used to determine the best parameter configuration is described, then the parameters that need to be adjusted are described and experiments are performed to determine the best parameter settings.

A.1 Method

If available and applicable, the recommended parameter settings are used. For the methods with parameters to test, the performance of 25 repetitions on the problems in the CEC2006 benchmark suite[38] is compared. Note that g20 and g22 are excluded as either no feasible solution is known or none of the methods were able to reach the feasible region. The computational budget used is $5e5$ evaluations. A single run is considered feasible if at least one feasible solution is found and successful if the approximation target ($f(x_{opt}) + 10^{-4}$) is reached as per [38].

To compare two parameter configurations, statistical significance tests are performed in the following order[1]:

1. Fisher's exact test[23] for categorical data is used to determine if one method reaches the feasible region significantly more reliably.

2. If there is no significant difference in reaching the feasible region, Fisher's exact test is again used to determine if one of the configurations leads to significantly more successful runs.

3. If there are no significant differences in feasibility or success rate, then the number of evaluations of the feasible runs is compared. As the number of evaluations is continuous, the non-parametric Mann-Whitney-U test[42] is used for this comparison.

[1] The tests were performed using the implementation provided in [69].

This order of comparisons determines if one configuration performs significantly better on a single problem. As significance level, $p = 0.01$ is used. This is done for all problems, accumulating the total number of times a configuration of the CHT is performing significantly better/not different/worse than the other configurations for the same CHT across all problems. Finally, the configuration that is worse least often is used. If there are ties, the configuration that is significantly better more often is used. If there still are ties, the method used is picked randomly from the best-performing methods. The p-values are not compared as different statistical tests are used to determine performance differences between configurations.

A.2 Parameters

Constraint Domination Principle CDP has no parameters and hence no need for parameter configuration.

ϵ-Constrained Method The original method has three parameters θ, $b_{control}$, and cp for determining the start, end, and rate of decline of the ϵ threshold respectively [61]. The values tested are $cp \in \{1, 3, 5, 7\}$ and $b_{control} \in \{0.1, 0.2, 0.5, 0.8\}$. For θ, the recommended value of $\theta = 0.2$[61] is used.

For the improved ϵ-constrained method proposed in [22], two additional parameters $\alpha = 0.8 \in [0, 1]$ and $\beta = 0.1 \in [0, 1]$ are introduced to control the feasibility of the population and the increase of the ϵ threshold respectively. In addition to the other parameters, the values $\alpha \in \{0.7, 0.8, 0.9\}$ and $\beta \in \{0.01, 0.1, 0.25\}$ are considered.

Stochastic Ranking This method originally has a single parameter P_f determining the chance of ignoring the constraint violation. In [54], values in the range $P_f \in [0.4, 0.5]$ are recommended, hence the values $P_f \in \{0.4, 0.45, 0.5\}$ are considered. In addition, the adaption to RV-GOMEA introduces a second parameter determining if the stochastic operator is used for acceptance during variation in addition to selection. Both options are tested.

Augmented Lagrangian This penalty function adds four parameters, determining the update of the Lagrangian multiplier and the penalty weight. In [19], extensive parameter tuning has been performed for this method and thus the first proposed setting is used.

Repair Operators Both introduced repair operators build on CDP and have an additional parameter $k \in \mathbb{N}$ determining the number of repair steps performed. The values considered are $k \in \{1, 2, 3, 4\}$. The resampling method has a second parameter $s_f \in [0, 1]$, determining the minimum feasibility of the selection. For this parameter, the values $s_f \in \{0.1, 0.5, 0.9\}$ are considered.

Dual-Population Methods The first, "oscillating" dual population approach has a parameter $\theta \in [\tau, \frac{1}{2}]$ determining when to split the population into two subpopulations. Another parameter $\theta_s \in [0, \frac{1}{2}]$ controls how much information is shared between the two subpopulations. The values considered are $\theta \in \{\tau = 0.35, 0.45\}$ and $\theta_s \in \{0, 0.2, 0.4\}$. In addition, both the proportional and full subpopulation schemes introduced in Chapter 3 are tested.

The second dual-population approach, DP-T, has parameters determining the traction strategy used to tether the infeasible population to the feasible population. Tested is no traction strategy (CDP), optimizing the distance to the feasible solutions (D2F), or using feasible solutions to influence the infeasible distribution estimation (FSI). For adding feasible solutions, $\theta_s \in [0, \frac{1}{2}]$ determines how many infeasible solutions are used. The values tested for this traction strategy are $\theta_s \in \{0.1, 0.3, 0.5\}$. The last parameter tested is $\tau_{min} \in [0, 1]$, controlling how big the infeasible population needs to be for it to be used. Here the values $\tau_{min} \in \{0.5, 0.7, 0.9\}$ are tested.

Furthermore, as the dual-population approaches change the structure of RV-GOMEA, the optimal population size is found as the recommendations in **??** do not necessarily apply to these approaches. A population size is considered optimal if the problem is solved reliably, i.e. 95% of the time, and with the least number of function evaluations. According to [5], finding the smallest population size where the problem is solved reliably is insufficient. This is due to a tradeoff between information and cost. Large population sizes provide a lot of information and reduce the uncertainty at the cost of an increased number of function evaluations. On the other hand, small sizes may have too little information, and thus the variance and number of function evaluations needed increases due to the stochastic nature of the algorithm. To correctly determine the optimal size, first, the smallest reliable size is found by exponentially increasing the population size and then performing binary search on the success rate. Then an upper bound is found similarly, before grid search is used in-between. The grid search step is then repeated using increased resolutions between updated bounds until finally the best population size for the problem is found.

Partially Infeasible Selection For this approach, several new parameters were added. The parameter $\eta \in [0, 1]$ controls when a population is considered to be feasible. In this thesis, η was set to 0.7, indicating that at least 70% of the individuals in the population need to be feasible before infeasible solutions are used. For search in the infeasible region, CDP or stochastic ranking (SR) is used during selection. Furthermore, $\theta_i \in [0, 1]$ controls how many infeasible solutions are used during selection, where the values $\{0.1, 0.3, 0.5, 0.7, 0.9\}$ are considered. Before the infeasible solutions are selected, however, the option to prune infeasible solutions with insufficient objective values is tested. The selection of infeasible solutions can be made using CDP, stochastic ranking (SR), based on the distance to the feasible selection (D2F), or using non-dominated sorting (NDS). Finally, the last parameter determines whether the selected infeasible solutions are added to the feasible selection or used to replace feasible solutions during distribution estimation.

Partially Infeasible This method consists of several proposed modifications to the DP-O approach and hence the parameter enabling proportional subpopulations is reevaluated. In addition, the distribution estimation can either be done jointly, or separately for each subpopulation. Selection for both infeasible populations and the infeasible subpopulation is done using either CDP or stochastic ranking (SR). The last parameter determines how many feasible solutions can be carried over to the next generation, given that they are still competitive with the current population. For this parameter, the values $\{0, \frac{\tau}{2}, \tau\}$ are considered.

A.3 Results

First, the results for determining appropriate population sizes are shown. In addition to the results of the statistical significance tests considering all combinations of the parameters, additional plots are included showcasing the effect of each individual parameter across all configurations tested. The configurations used in the main chapters are highlighted in bold in the comparison tables.

A.3.1 Population Size

The results for determining the best sizes are shown in Figure A.1. Only results are shown where reliable success was observed, other problems are not shown. The recommended population size for the Full FOS as per [7] is shown as a vertical line. The results for CDP indicate that the recommended size does indeed apply to the majority of problems. Only problems g01, g12, and g18 are exceptions. The same holds for the dual-population approaches, hence all methods are used with the recommended population size.

A.3.2 Best Configurations

ϵ-Constrained Method

Table A.1: The significance testing results ($p < 0.01$) over all used CEC2006 problems for the EPS method. Values indicate the number of times the configuration was significantly better/not significantly different/significantly worse compared to the other configurations. The colors correspond to the ranks of the configurations after non-dominated sorting, darker colors correspond to worse performance.

	$b_{control}$			
cp	0.1	0.2	0.5	0.8
1	155 / 128 / 47	100 / 138 / 92	55 / 135 / 140	10 / 135 / 185
3	178 / 126 / 26	105 / 137 / 88	56 / 137 / 137	11 / 144 / 175
5	203 / 109 / 18	134 / 122 / 74	69 / 139 / 122	22 / 142 / 166
7	**232 / 95 / 3**	147 / 126 / 57	77 / 142 / 111	36 / 145 / 149

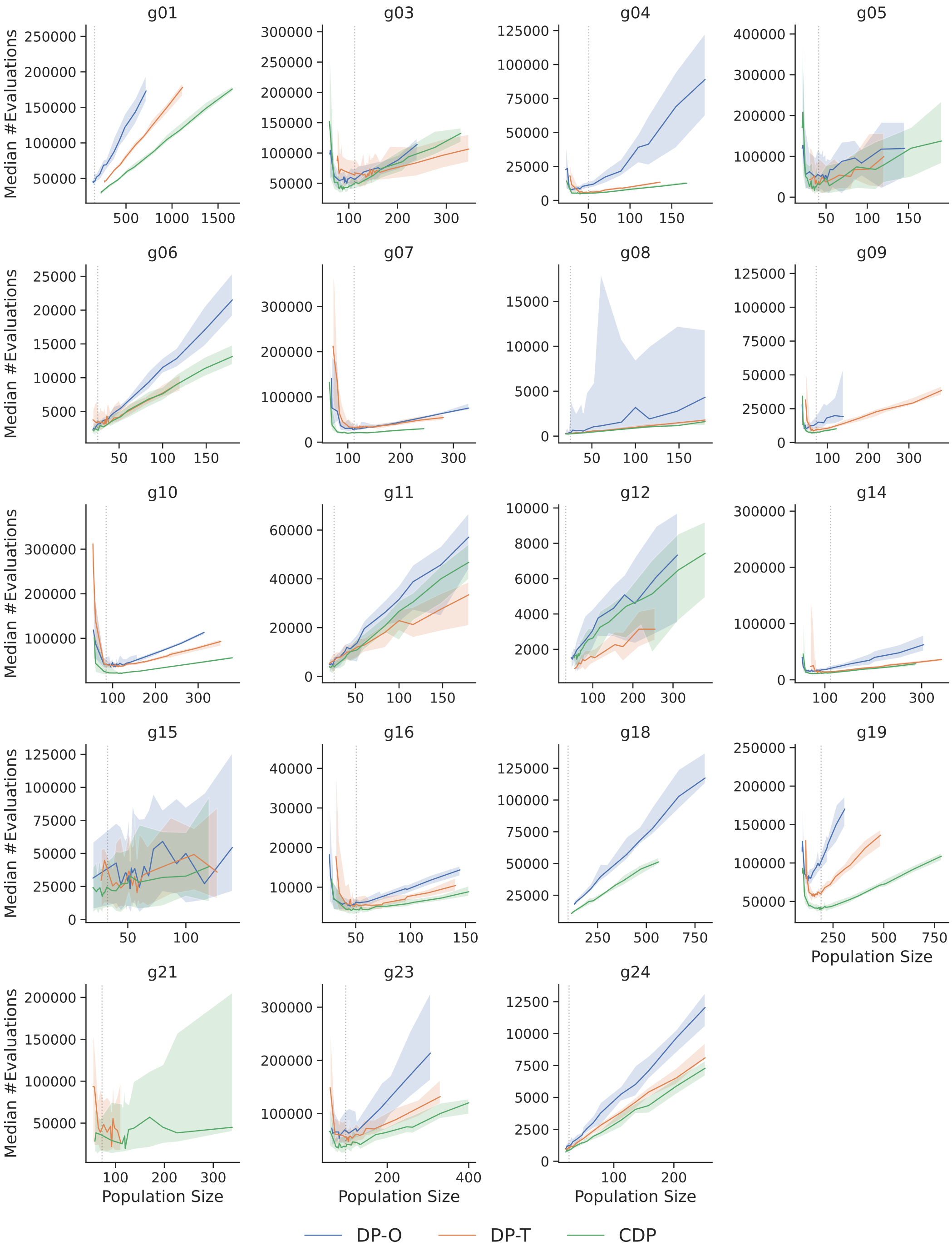


Figure A.1: The effect of the population size on the number of evaluations needed, for population sizes reaching the approximation target in at least 95% of 25 runs.

Improved ϵ-Constrained Method

Table A.2: The significance testing results ($p < 0.01$) over all used CEC2006 problems for the IEPS method. Values indicate the number of times the configuration was significantly better/not significantly different/significantly worse compared to the other configurations. The colors correspond to the ranks of the configurations after non-dominated sorting, darker colors correspond to worse performance.

cp	α	β	$b_{control}$ 0.1	0.2	0.5	0.8
1	0.7	0.01	1732 / 929 / 485	1114 / 996 / 1036	613 / 980 / 1553	90 / 986 / 2070
		0.1	1730 / 924 / 492	1114 / 999 / 1033	603 / 986 / 1557	88 / 982 / 2076
		0.25	1739 / 926 / 481	1129 / 978 / 1039	605 / 998 / 1543	93 / 964 / 2089
	0.8	0.01	1735 / 926 / 485	1115 / 1002 / 1029	608 / 971 / 1567	104 / 981 / 2061
		0.1	1741 / 920 / 485	1097 / 1005 / 1044	613 / 985 / 1548	91 / 988 / 2067
		0.25	1718 / 911 / 517	1117 / 1003 / 1026	608 / 990 / 1548	104 / 972 / 2070
	0.9	0.01	1749 / 938 / 459	1114 / 1004 / 1028	602 / 990 / 1554	98 / 991 / 2057
		0.1	1725 / 932 / 489	1115 / 1004 / 1027	604 / 986 / 1556	100 / 988 / 2058
		0.25	1718 / 937 / 491	1147 / 956 / 1043	603 / 989 / 1554	88 / 993 / 2065
3	0.7	0.01	1983 / 841 / 322	1198 / 996 / 952	652 / 1027 / 1467	141 / 1029 / 1976
		0.1	2002 / 775 / 369	1208 / 976 / 962	640 / 1027 / 1479	148 / 1053 / 1945
		0.25	2049 / 781 / 316	1180 / 1009 / 957	688 / 951 / 1507	124 / 1066 / 1956
	0.8	0.01	2094 / 733 / 319	1265 / 931 / 950	651 / 1021 / 1474	149 / 1061 / 1936
		0.1	1985 / 843 / 318	1273 / 922 / 951	644 / 1033 / 1469	145 / 1032 / 1969
		0.25	1989 / 844 / 313	1238 / 963 / 945	643 / 1041 / 1462	146 / 1000 / 2000
	0.9	0.01	1984 / 844 / 318	1225 / 967 / 954	628 / 1069 / 1449	178 / 976 / 1992
		0.1	1982 / 851 / 313	1186 / 1008 / 952	649 / 1026 / 1471	140 / 1017 / 1989
		0.25	1985 / 855 / 306	1208 / 981 / 957	653 / 1021 / 1472	140 / 1025 / 1981
5	0.7	0.01	2203 / 755 / 188	1511 / 844 / 791	766 / 1044 / 1336	259 / 1049 / 1838
		0.1	2240 / 762 / 144	1431 / 930 / 785	799 / 1012 / 1335	260 / 1059 / 1827
		0.25	2230 / 750 / 166	1386 / 971 / 789	763 / 1039 / 1344	255 / 1094 / 1797
	0.8	0.01	2248 / 764 / 134	1516 / 839 / 791	803 / 1009 / 1334	259 / 1043 / 1844
		0.1	2202 / 794 / 150	1464 / 896 / 786	795 / 1007 / 1344	262 / 1048 / 1836
		0.25	2233 / 772 / 141	1493 / 874 / 779	781 / 1023 / 1342	258 / 1044 / 1844
	0.9	0.01	2233 / 776 / 137	1508 / 845 / 793	784 / 1039 / 1323	242 / 1072 / 1832
		0.1	2181 / 813 / 152	1453 / 905 / 788	811 / 1001 / 1334	286 / 1026 / 1834
		0.25	2238 / 779 / 129	1469 / 898 / 779	778 / 1031 / 1337	262 / 1052 / 1832
7	0.7	0.01	2477 / 635 / 34	1650 / 862 / 634	843 / 1031 / 1272	332 / 1012 / 1802
		0.1	2513 / 630 / 3	1653 / 850 / 643	884 / 1012 / 1250	370 / 982 / 1794
		0.25	2508 / 614 / 24	1628 / 871 / 647	848 / 1029 / 1269	337 / 1044 / 1765
	0.8	0.01	2505 / 618 / 23	1620 / 867 / 659	835 / 1035 / 1276	321 / 1070 / 1755
		0.1	2500 / 643 / 3	1649 / 855 / 642	854 / 1028 / 1264	320 / 1067 / 1759
		0.25	2518 / 626 / 2	1643 / 847 / 656	894 / 995 / 1257	330 / 1075 / 1741
	0.9	0.01	2509 / 605 / 32	1622 / 868 / 656	872 / 996 / 1278	334 / 1032 / 1780
		0.1	**2538 / 607 / 1**	1618 / 873 / 655	872 / 1021 / 1253	324 / 1097 / 1725
		0.25	2542 / 602 / 2	1657 / 850 / 639	869 / 1010 / 1267	334 / 1017 / 1795

Stochastic Ranking

Table A.3: The significance testing results ($p < 0.01$) over all used CEC2006 problems for the SR method. Values indicate the number of times the configuration was significantly better/not significantly different/significantly worse compared to the other configurations. The colors correspond to the ranks of the configurations after non-dominated sorting, darker colors correspond to worse performance.

Acceptance	P_f 0.4	P_f 0.45	P_f 0.5
CDP	57 / 40 / 13	**60 / 43 / 7**	63 / 39 / 8
SR	30 / 24 / 56	23 / 27 / 60	4 / 13 / 93

Repair using Binary Search

Table A.4: The significance testing results ($p < 0.01$) over all used CEC2006 problems for the R-BS method. Values indicate the number of times the configuration was significantly better/not significantly different/significantly worse compared to the other configurations. The colors correspond to the ranks of the configurations after non-dominated sorting, darker colors correspond to worse performance.

k = 1	2	3	4
18 / 48 / 0	15 / 49 / 2	7 / 47 / 12	0 / 40 / 26

Repair using Resampling

Table A.5: The significance testing results ($p < 0.01$) over all used CEC2006 problems for the R-R method. Values indicate the number of times the configuration was significantly better/not significantly different/significantly worse compared to the other configurations. The colors correspond to the ranks of the configurations after non-dominated sorting, darker colors correspond to worse performance.

θ_s	k = 1	2	3	4
0.1	84 / 154 / 4	28 / 189 / 25	7 / 192 / 43	0 / 178 / 64
0.5	**89 / 153 / 0**	31 / 193 / 18	9 / 194 / 39	2 / 169 / 71
0.9	75 / 166 / 1	39 / 186 / 17	8 / 197 / 37	2 / 185 / 55

Oscillating Dual-Population Method

Table A.6: The significance testing results ($p < 0.01$) over all used CEC2006 problems for the DP-O method. Values indicate the number of times the configuration was significantly better/not significantly different/significantly worse compared to the other configurations. The colors correspond to the ranks of the configurations after non-dominated sorting, darker colors correspond to worse performance.

θ	Proportional	θ_s 0	0.2	0.4
0.35	No	67 / 165 / 10	**72 / 161 / 9**	52 / 173 / 17
	Yes	29 / 146 / 67	32 / 160 / 50	41 / 152 / 49
0.45	No	33 / 173 / 36	33 / 178 / 31	23 / 172 / 47
	Yes	19 / 174 / 49	21 / 179 / 42	19 / 189 / 34

Dual-Population Method with Traction Strategy

Table A.7: The significance testing results ($p < 0.01$) over all used CEC2006 problems for the DP-T method. Values indicate the number of times the configuration was significantly better/not significantly different/significantly worse compared to the other configurations. The colors correspond to the ranks of the configurations after non-dominated sorting, darker colors correspond to worse performance.

Traction Strategy	τ_{min}	-	FSI Percentile 0.1	0.3	0.5
CDP	0.5	0 / 175 / 133	-	-	-
	0.7	1 / 173 / 134	-	-	-
	0.9	0 / 180 / 128	-	-	-
D2F	0.5	0 / 172 / 136	-	-	-
	0.7	0 / 177 / 131	-	-	-
	0.9	0 / 177 / 131	-	-	-
FSI	0.5	-	80 / 189 / 39	101 / 206 / 1	**121 / 187 / 0**
	0.7	-	87 / 196 / 25	108 / 196 / 4	104 / 203 / 1
	0.9	-	87 / 191 / 30	99 / 209 / 0	105 / 203 / 0

Partially Infeasible

For the partially infeasible approach, an exception was made, as there are two configurations that perform well but differ substantially. One configuration uses a joint distribution, while the other one does not and makes use of feasible solutions that would otherwise be discarded when accepting infeasible solutions. Hence, both configurations are evaluated on the other problems.

Table A.8: The significance testing results ($p < 0.01$) over all used CEC2006 problems for the PI method. Values indicate the number of times the configuration was significantly better/not significantly different/significantly worse compared to the other configurations. The colors correspond to the ranks of the configurations after non-dominated sorting, darker colors correspond to worse performance.

Proportional	Joint Distribution	Infeasible Subpopulation	Feasible Solutions Reconsidered: -	$\frac{\tau}{2}$	τ
No	No	CDP	-	32 / 195 / 125	41 / 186 / 125
		SR	130 / 179 / 43	**139 / 202 / 11**	149 / 186 / 17
	Yes	CDP	31 / 171 / 150	9 / 179 / 164	13 / 176 / 163
		SR	126 / 216 / 10	122 / 202 / 28	111 / 227 / 14
Yes	Yes	CDP	38 / 170 / 144	7 / 193 / 152	14 / 188 / 150
		SR	**134 / 209 / 9**	118 / 208 / 26	131 / 207 / 14

Partially Infeasible Selection

Table A.9: The significance testing results ($p < 0.01$) over all used CEC2006 problems for the PIS method. Values indicate the number of times the configuration was significantly better/not significantly different/significantly worse compared to the other configurations. The colors correspond to the ranks of the configurations after non-dominated sorting, darker colors correspond to worse performance.

Infeasible Region	Infeasible Selection	Pruning	Distribution	P_f	θ_i 0.1	0.3	0.5	0.7	0.9
CDP	CDP	No	A	-	1309/3277/1112	1178/3288/1232	845/3093/1760	683/2645/2370	510/2088/3100
CDP	CDP	No	R	-	1274/3217/1207	1175/3194/1329	930/3257/1511	554/2539/2605	461/2566/2671
CDP	CDP	Yes	A	-	1826/2942/930	1960/3030/708	1863/3112/723	1435/3190/1073	992/2771/1935
CDP	CDP	Yes	R	-	1379/3109/1210	1986/3102/610	1987/3003/708	2011/3051/636	1817/3081/800
CDP	D2F	No	A	-	1198/3155/1345	1048/3115/1535	815/2945/1938	648/2495/2555	493/1940/3265
CDP	D2F	No	R	-	1206/3335/1157	996/3008/1694	801/2996/1901	684/2691/2323	578/2321/2799
CDP	D2F	Yes	A	-	1530/3266/902	1582/3177/939	1001/3042/1655	832/2829/2037	560/2572/2566
CDP	D2F	Yes	R	-	1465/3098/1135	1331/3273/1094	963/3137/1598	969/2946/1783	909/2895/1894
CDP	NDS	No	A	-	1617/3173/908	1543/2524/1631	997/2312/2389	716/2019/2963	509/1427/3762
CDP	NDS	No	R	-	1393/3316/989	1388/2637/1673	579/2333/2786	321/1374/4003	215/1290/4193
CDP	NDS	Yes	A	-	1666/3159/873	1067/2535/2096	513/1768/3417	314/1470/3914	171/1441/4086
CDP	NDS	Yes	R	-	1409/3396/893	1052/2570/2076	392/1742/3564	210/1494/3994	113/1263/4322
CDP	SR	No	A	0.4	1270/3260/1168	1509/3470/719	1447/2965/1286	1239/3053/1406	842/2889/1967
CDP	SR	No	A	0.45	1504/3170/1024	1758/3283/657	1697/3050/951	1539/2941/1218	1250/2878/1570
CDP	SR	No	R	0.4	1217/3142/1339	1410/3428/860	1407/3334/957	1323/3225/1150	1278/3223/1197
CDP	SR	No	R	0.45	1187/3221/1290	1677/3239/782	1568/3376/754	1516/3166/1016	1465/3306/927
CDP	SR	Yes	A	0.4	1697/3020/981	1843/2998/857	1560/3106/1032	1116/2971/1611	796/2560/2342
CDP	SR	Yes	A	0.45	1864/2994/840	1874/2873/951	1295/3020/1383	912/2582/2204	619/2400/2679
CDP	SR	Yes	R	0.4	1478/3139/1081	1874/3115/709	1716/3084/898	1625/3050/1023	1427/3149/1122
CDP	SR	Yes	R	0.45	1435/3268/995	1708/2998/992	1504/3168/1026	1035/2804/1859	1272/2872/1554
	CDP	No	A	0.4	1607/3414/677	1451/3424/823	1405/3061/1232	1065/2643/1990	806/2406/2486
	CDP	No	A	0.45	1975/3203/520	1903/3035/760	1545/2974/1179	1356/2504/1838	1069/2394/2235
	CDP	No	R	0.4	1530/3493/675	1543/3401/754	1265/3186/1247	940/2880/1878	814/2395/2489
	CDP	No	R	0.45	1848/3228/622	1835/3069/794	1574/3026/1098	1337/2635/1726	1103/2288/2307
	CDP	Yes	A	0.4	2046/3182/470	2624/2939/135	2222/3134/342	1878/3123/697	1314/2983/1401
	CDP	Yes	A	0.45	2434/3054/210	2673/2817/208	2583/2777/338	1910/3050/738	1664/2760/1274
	CDP	Yes	R	0.4	1639/3511/548	2224/3238/236	2531/3022/145	2277/3242/179	2273/3143/282
	CDP	Yes	R	0.45	2033/3240/425	**2789/2819/90**	2709/2874/115	2656/2758/284	2625/2788/285
	D2F	No	A	0.4	1537/3597/564	1381/3303/1014	1150/3166/1382	916/2657/2125	808/2255/2635
	D2F	No	A	0.45	1818/3200/680	1688/2997/1013	1317/2817/1564	1143/2504/2051	1032/2301/2365
	D2F	No	R	0.4	1508/3553/637	1363/3208/1127	1179/2948/1571	1020/2787/1891	780/2744/2174
	D2F	No	R	0.45	1680/3244/774	1607/3023/1068	1433/2813/1452	1306/2668/1724	1145/2560/1993
	D2F	Yes	A	0.4	1724/3620/354	1795/3490/413	1342/3403/953	1105/2993/1600	816/2584/2298

Continued on next page

Table A.9: The significance testing results ($p < 0.01$) over all used CEC2006 problems for the PIS method. Values indicate the number of times the configuration was significantly better/not significantly different/significantly worse compared to the other configurations. The colors correspond to the ranks of the configurations after non-dominated sorting, darker colors correspond to worse performance.

Infeasible Region	Infeasible Selection	Pruning	Distribution	P_f	θ_i 0.1	0.3	0.5	0.7	0.9
				0.45	2177/3198/323	2050/3183/465	1538/3111/1049	1276/2902/1520	1228/2498/1972
			R	0.4	1614/3476/608	1699/3415/584	1357/3372/969	1234/3107/1357	1194/3097/1407
				0.45	1851/3355/492	2104/3159/435	1520/3202/976	1461/3005/1232	1364/2910/1424
	NDS	No	A	0.4	1950/3420/328	1623/3103/972	1244/2511/1943	781/2620/2297	695/2136/2867
				0.45	2061/3316/321	2049/3159/490	1203/3194/1301	904/2924/1870	700/2147/2851
			R	0.4	1757/3405/536	1516/3231/951	597/2713/2388	283/1755/3660	184/1610/3904
				0.45	1945/3369/384	1867/3321/510	864/2850/1984	408/1930/3360	370/1652/3676
		Yes	A	0.4	1813/3479/406	1167/2931/1600	466/2271/2961	270/2051/3377	191/1909/3598
				0.45	2063/3369/266	1403/3211/1084	706/2547/2445	495/2165/3038	421/1934/3343
			R	0.4	1760/3465/473	1261/2878/1559	321/2071/3306	148/1599/3951	86/1678/3934
				0.45	2084/3237/377	1372/3065/1261	590/2345/2763	227/1732/3739	328/1499/3871
	SR	No	A	0.4	1550/3621/527	1962/3273/463	1918/3090/690	1728/3103/867	1266/2897/1535
				0.45	1903/3291/504	2415/3025/258	2260/2972/466	2025/3082/591	1800/2872/1026
			R	0.4	1442/3493/763	1611/3593/494	1814/3193/691	1885/3163/650	1634/3347/717
				0.45	1838/3319/541	2398/3106/194	2460/2809/429	2253/2957/488	2262/2890/546
		Yes	A	0.4	1915/3401/382	2122/3324/252	1907/3262/529	1360/3284/1054	1016/2819/1863
				0.45	2227/3217/254	2432/3042/224	1798/3220/680	1277/3087/1334	1087/2595/2016
			R	0.4	1706/3495/497	2159/3360/179	2011/3271/416	1775/3505/418	1769/3323/606
				0.45	2021/3287/390	2347/3115/236	2118/3124/456	1716/3252/730	1590/3223/885

A.3.3 Effect of Individual Parameters

Here the results for the CEC2006 problems are presented where the effect of individual parameters is shown across all configurations used to determine the best parameter settings. Some parameters show clear differences in performance, for instance the cp parameter for the ϵ-Constrained methods or the number of repair steps k for the repair methods. Other parameters have a different effect when combined with other parameter values. One example is the P_f parameter of stochastic ranking, where the effect on the performance and reliability is overshadowed by the choice of whether the stochastic comparison is used for accepting offspring solutions during variation. Lastly, some parameters, such as the α and β values for the IEPS method, have little to no effect on the performance when considering all possible configurations for the remaining parameters.

In the figures, the light gray bars and the percentage at the top of each column correspond to the feasible rate, and the darker bar and percentage at the bottom to the success rate.

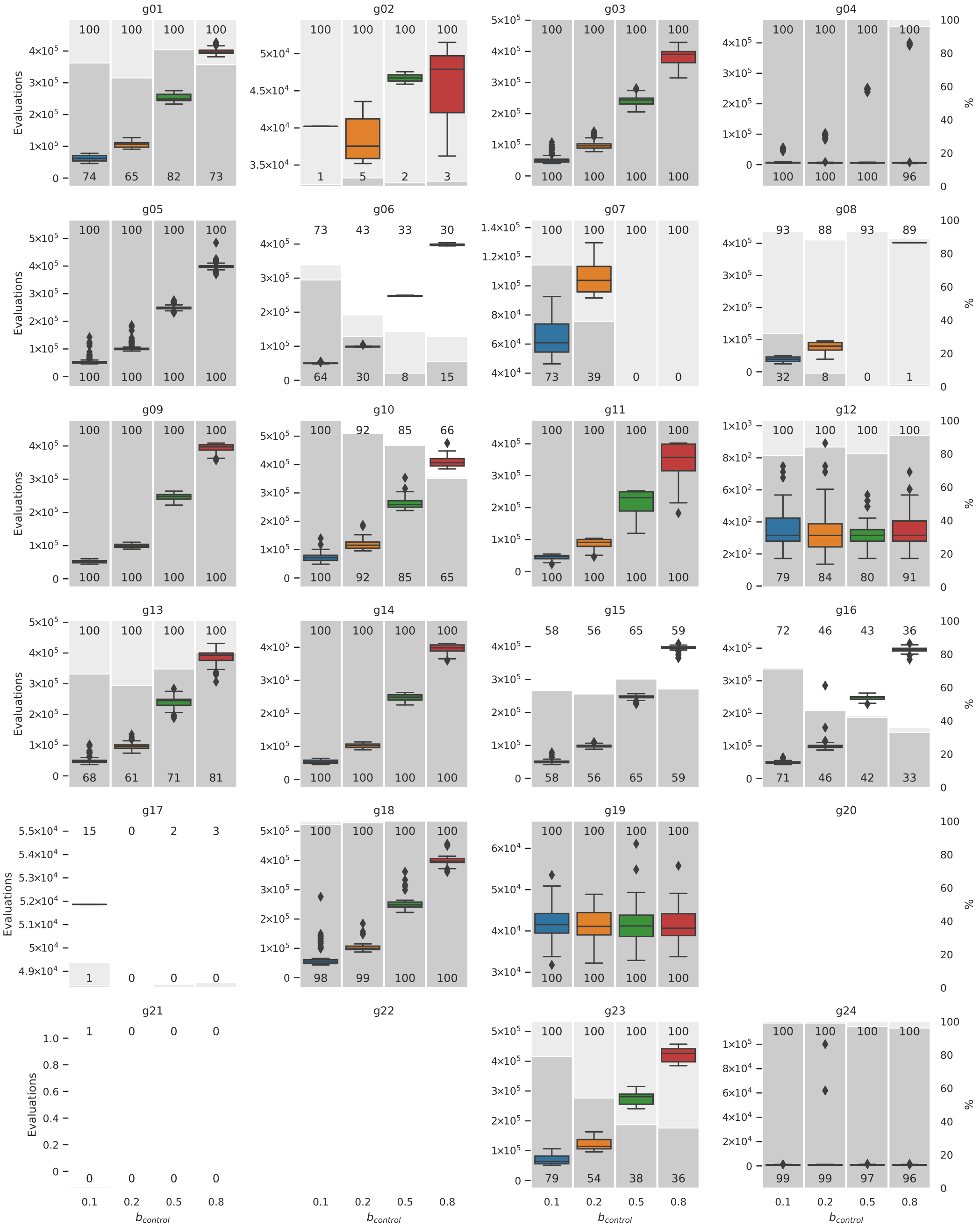


Figure A.2: The effect of the $b_{control}$ parameter of the EPS method over all runs performed.

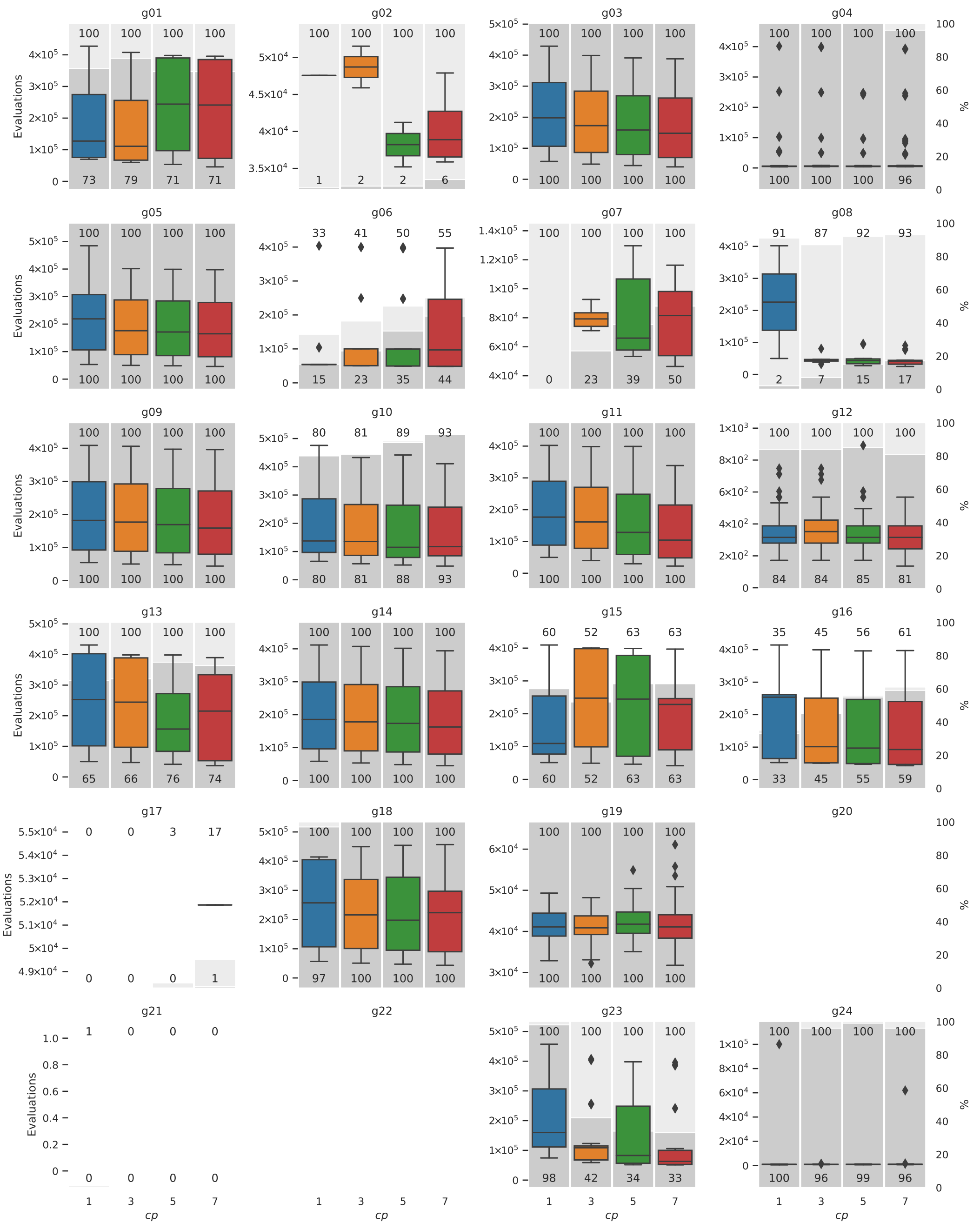


Figure A.3: The effect of the *cp* parameter of the EPS method over all runs performed.

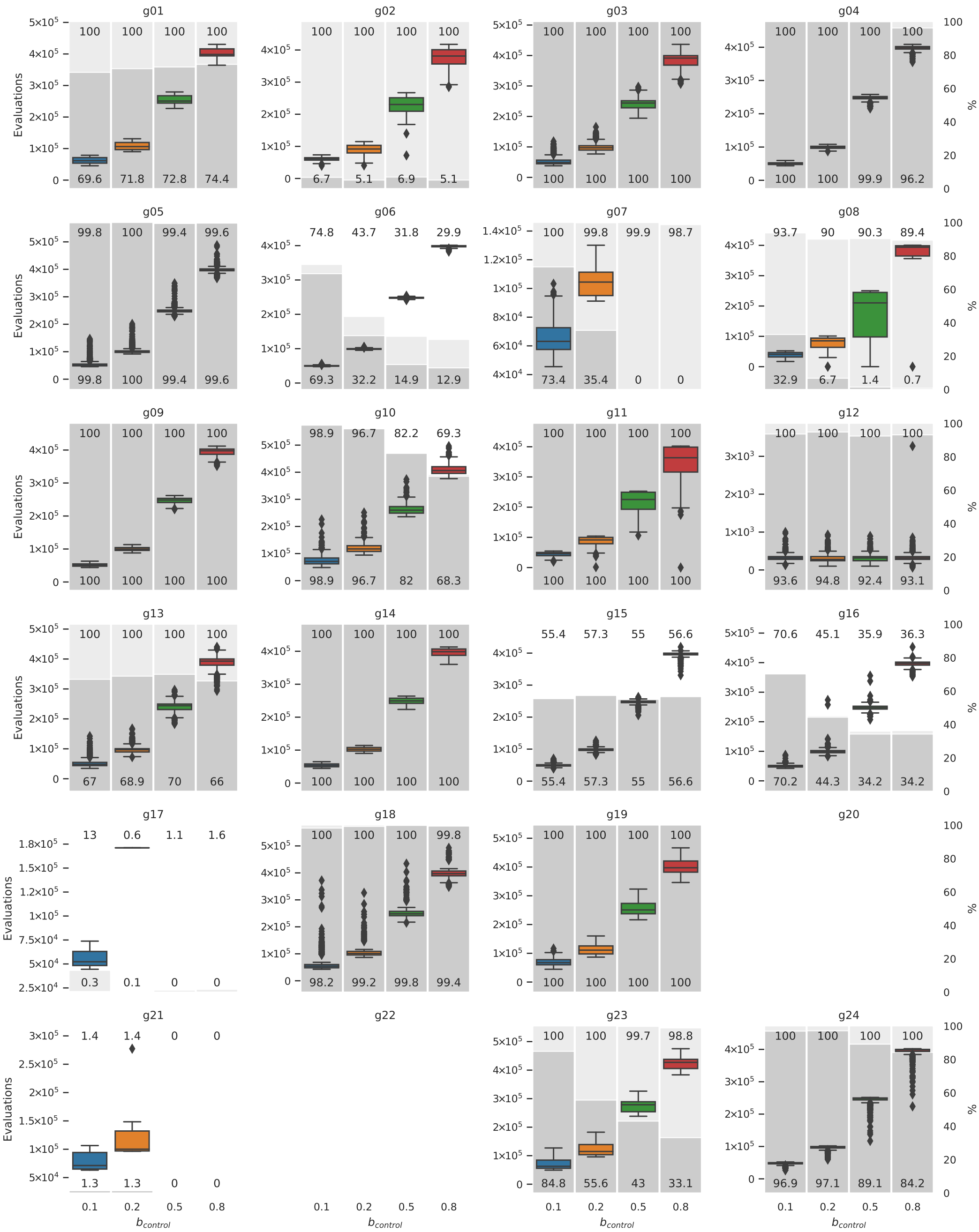


Figure A.4: The effect of the $b_{control}$ parameter of the IEPS method over all runs performed.

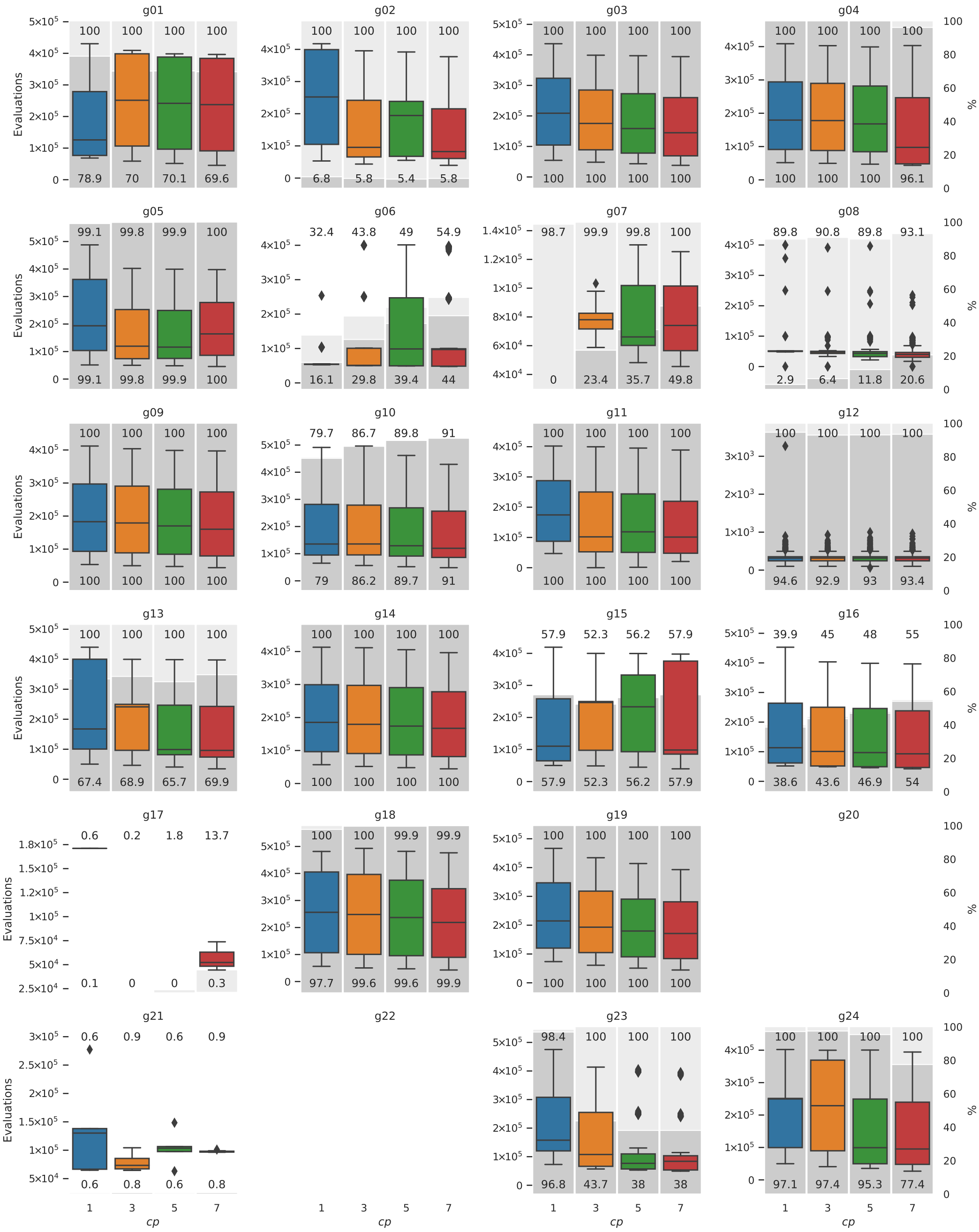


Figure A.5: The effect of the *cp* parameter of the IEPS method over all runs performed.

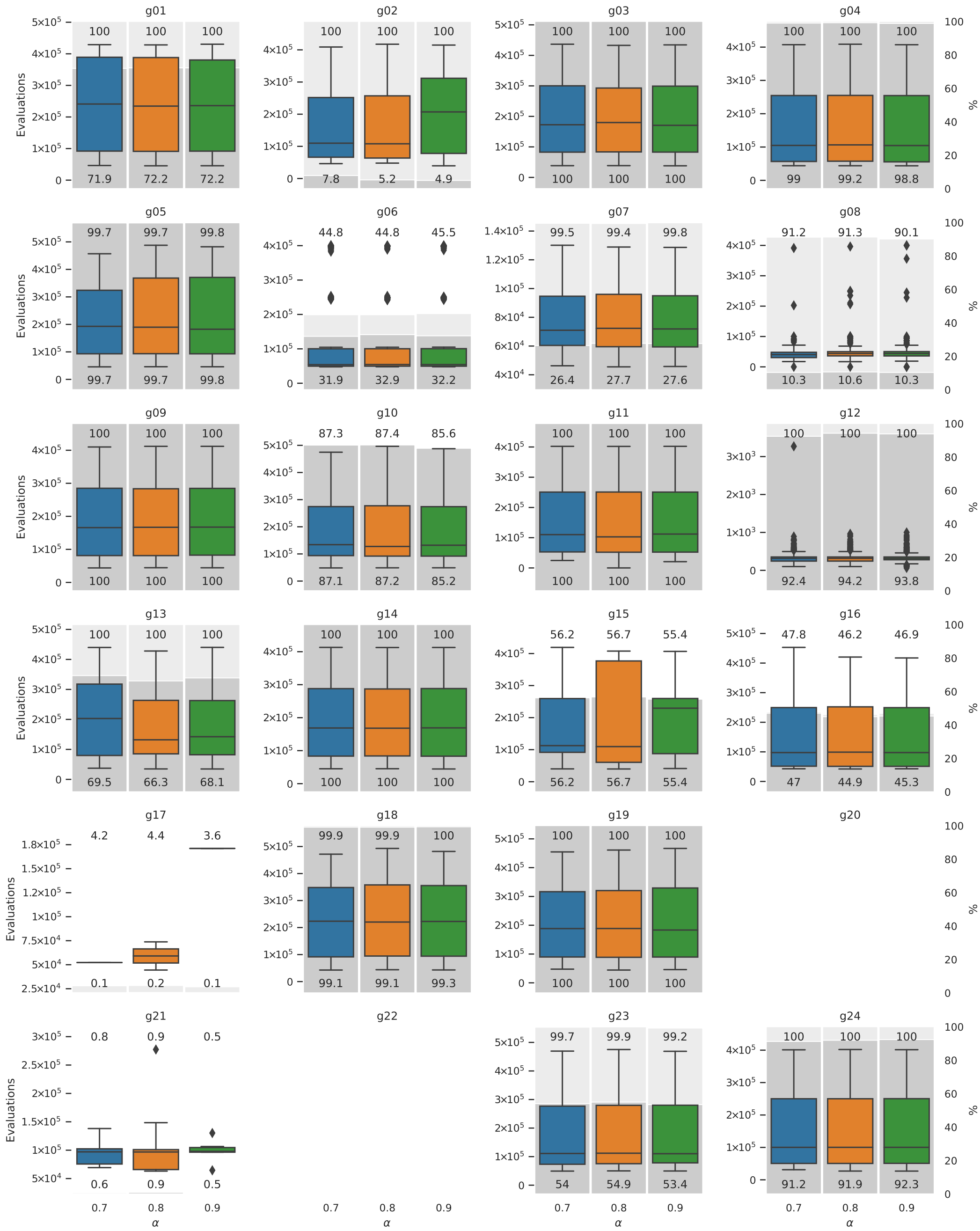


Figure A.6: The effect of the α parameter of the IEPS method over all runs performed.

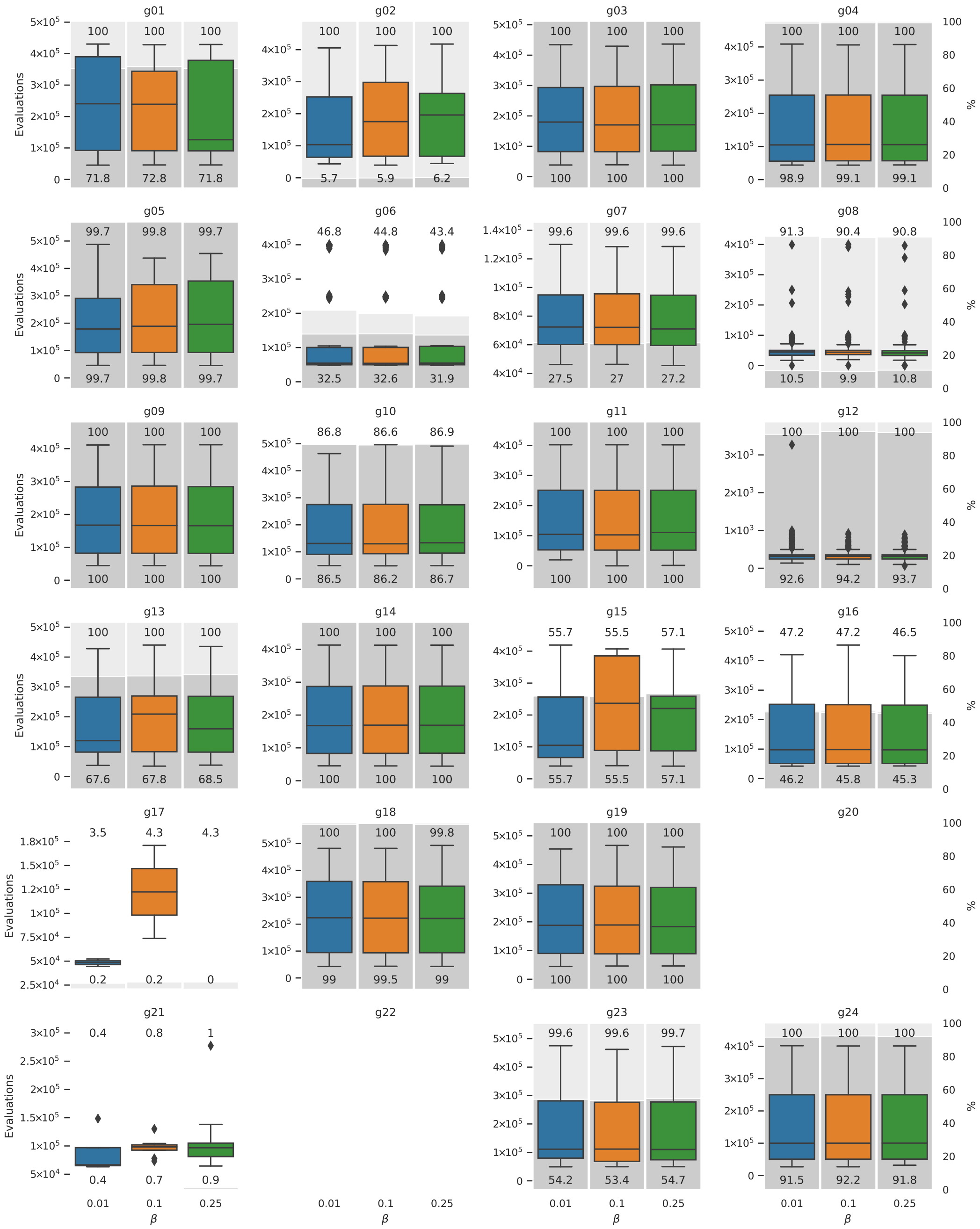


Figure A.7: The effect of the β parameter of the IEPS method over all runs performed.

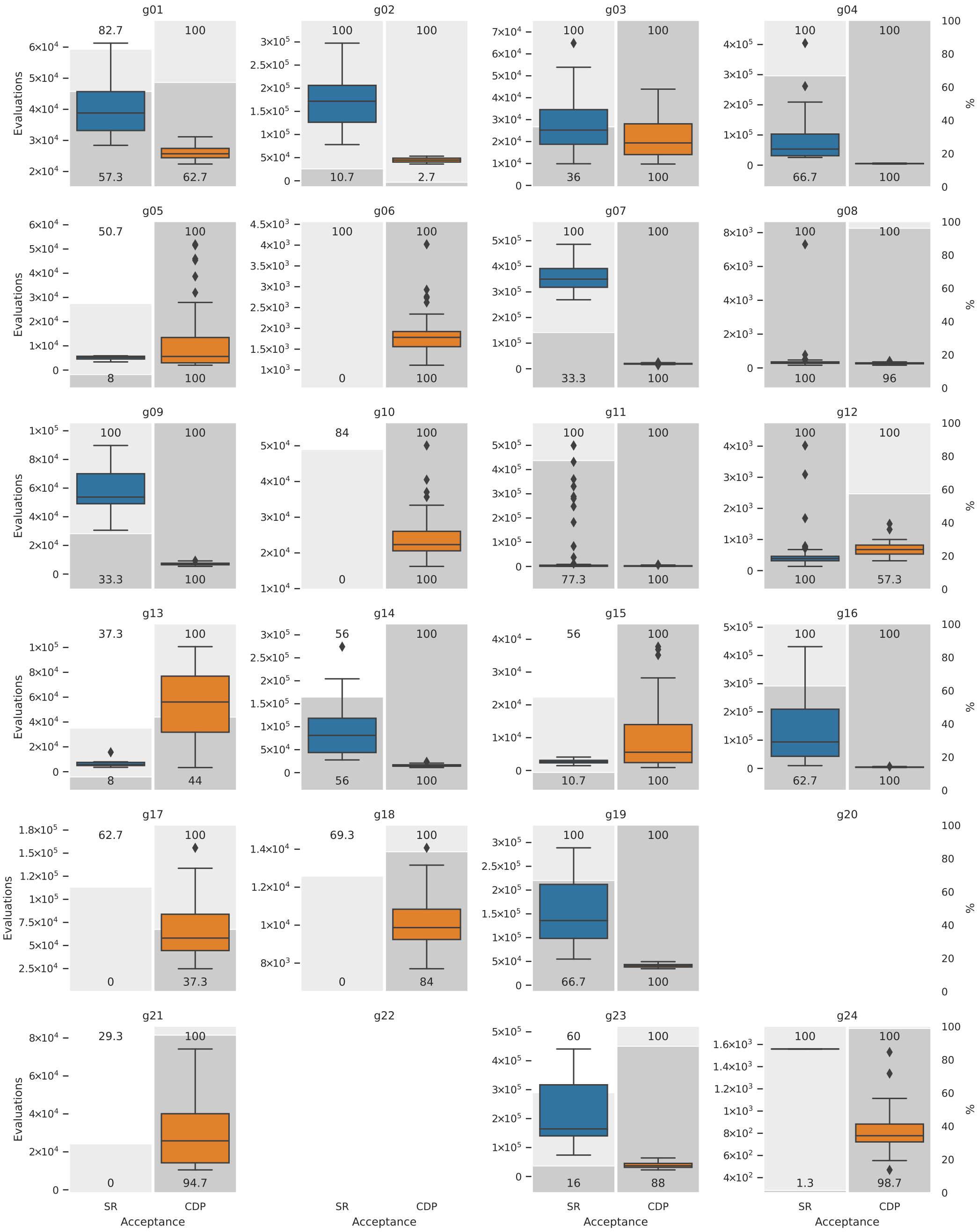


Figure A.8: The effect of the acceptance strategy of the SR method over all runs performed.

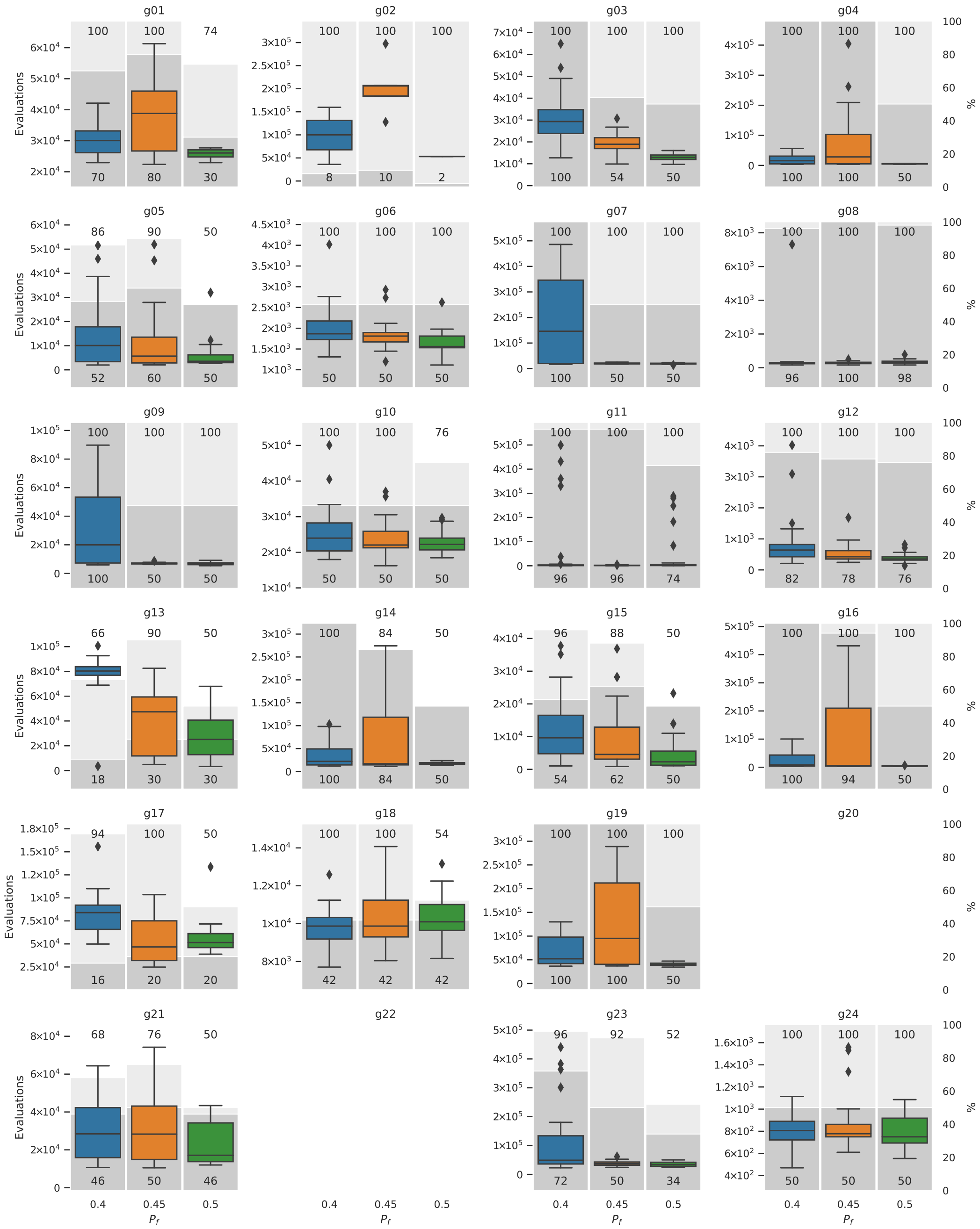


Figure A.9: The effect of the P_f parameter of the SR method over all runs performed.

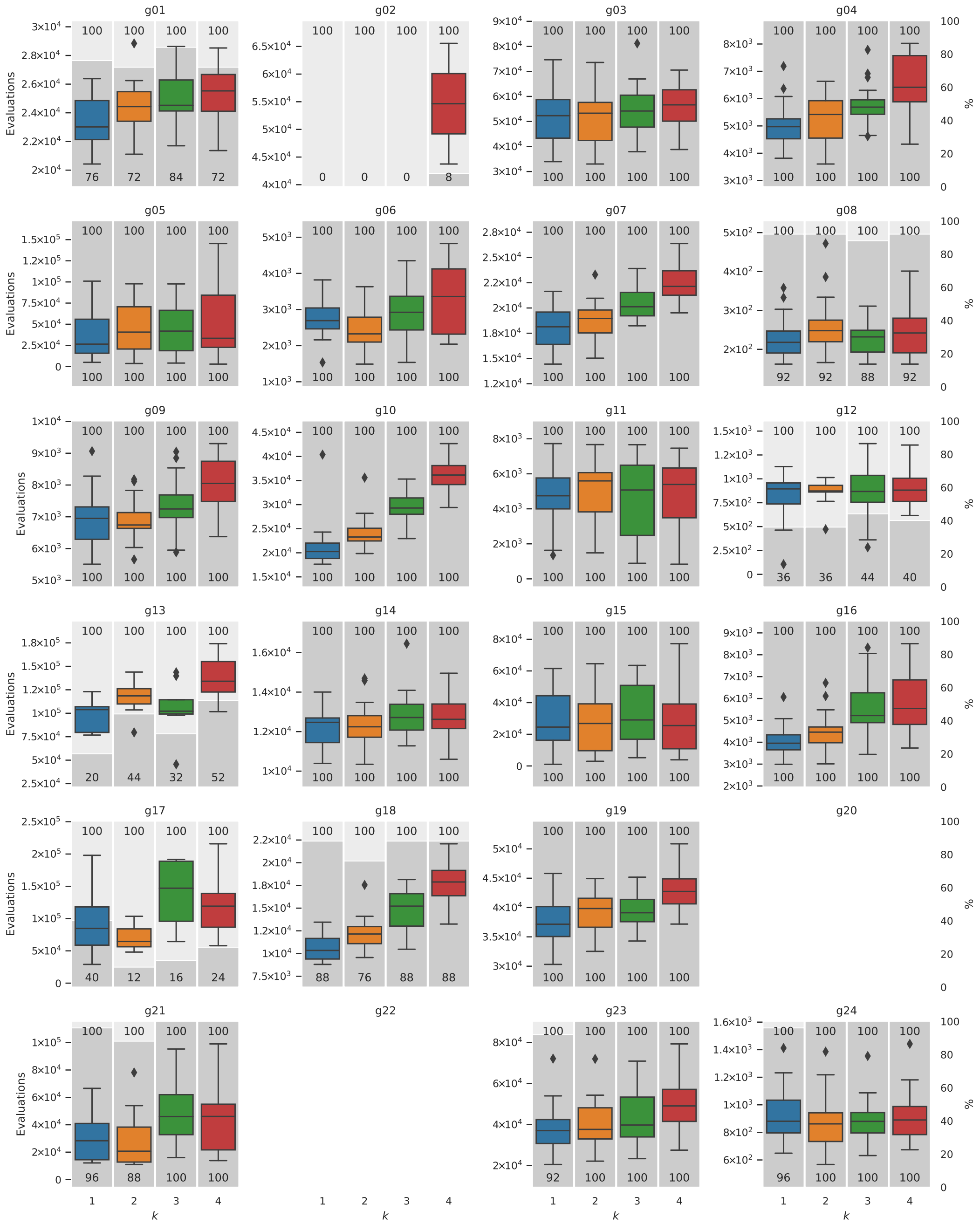


Figure A.10: The effect of the k parameter of the R-BS method over all runs performed.

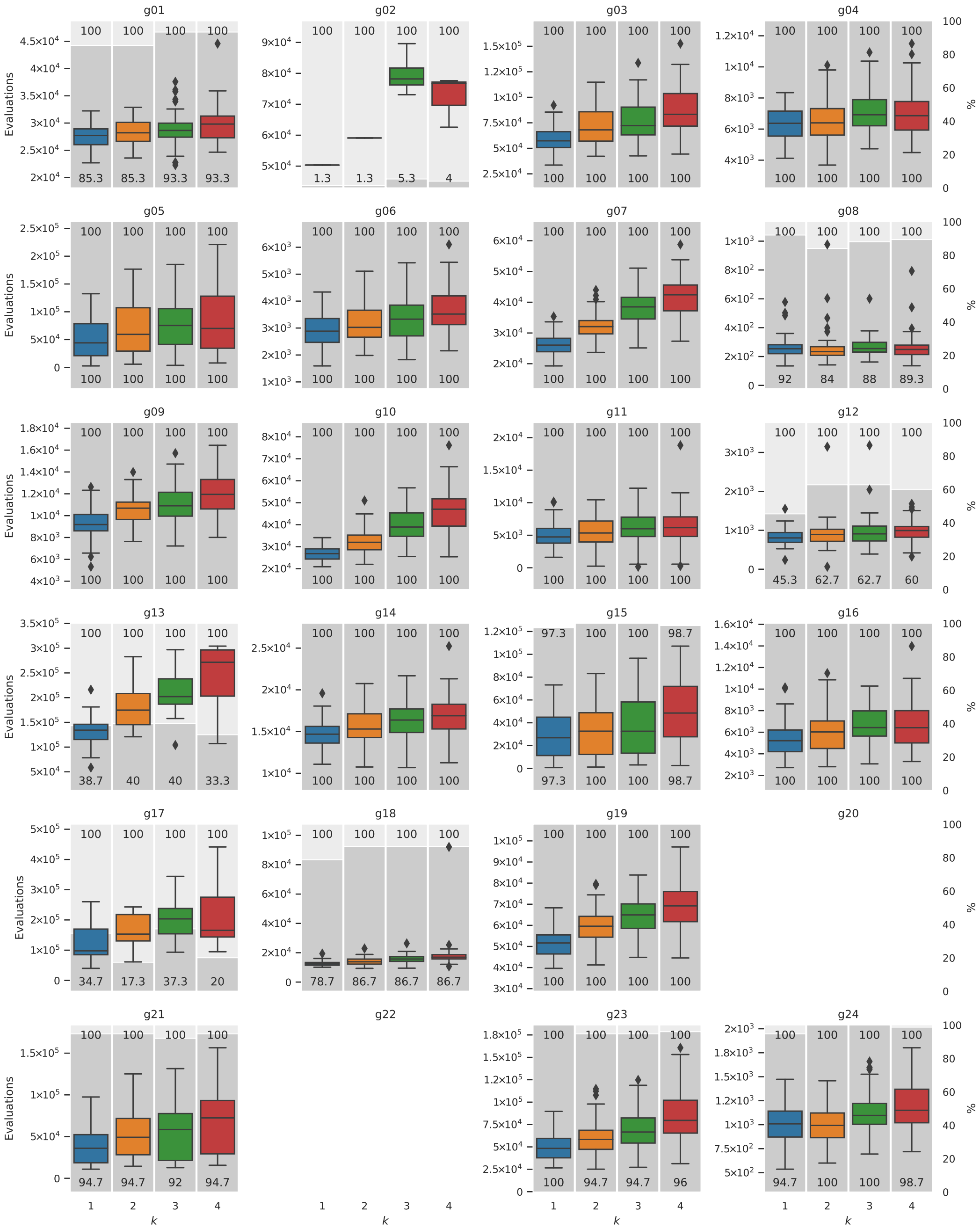


Figure A.11: The effect of the k parameter of the R-R method over all runs performed.

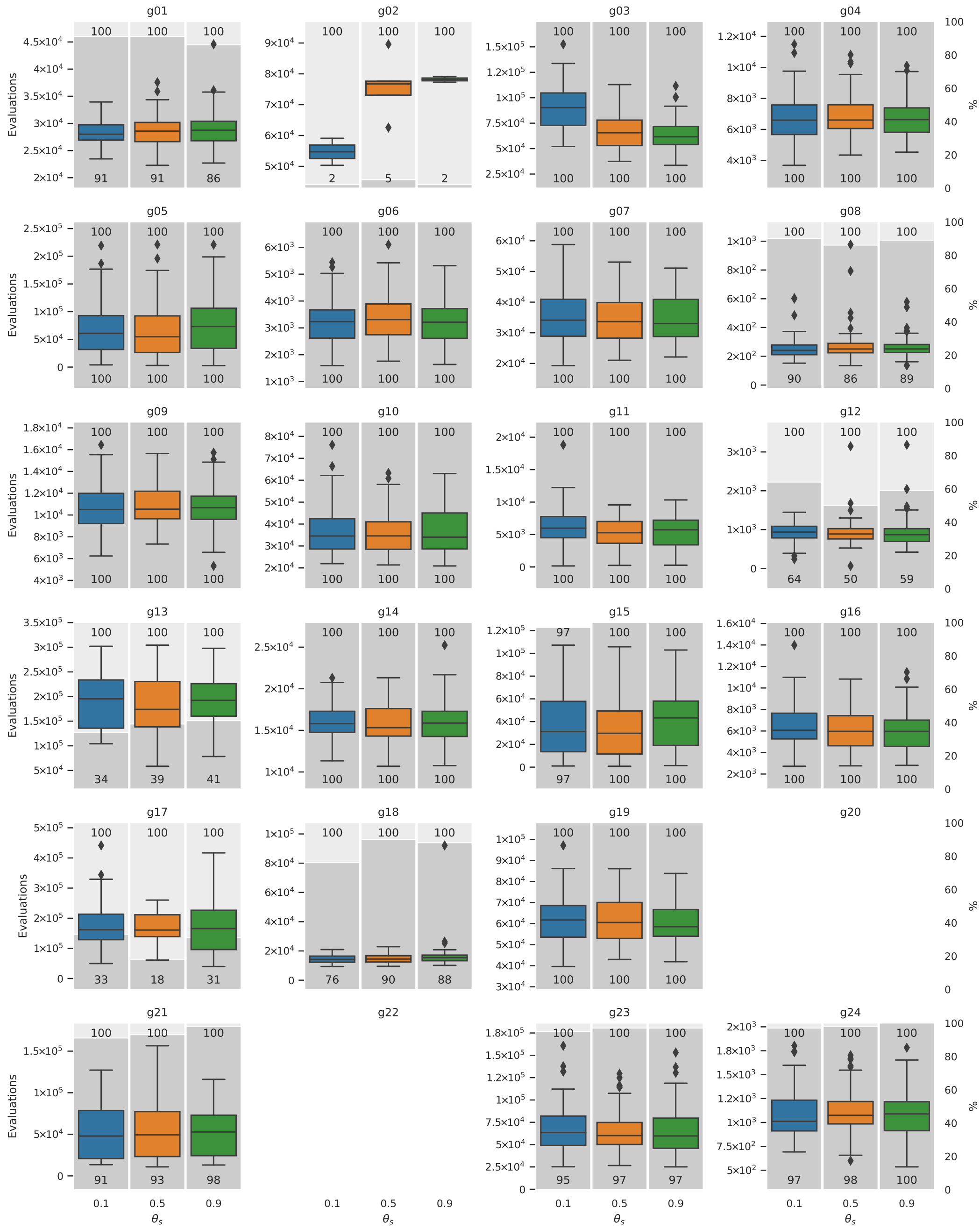


Figure A.12: The effect of the θ_s parameter of the R-R method over all runs performed.

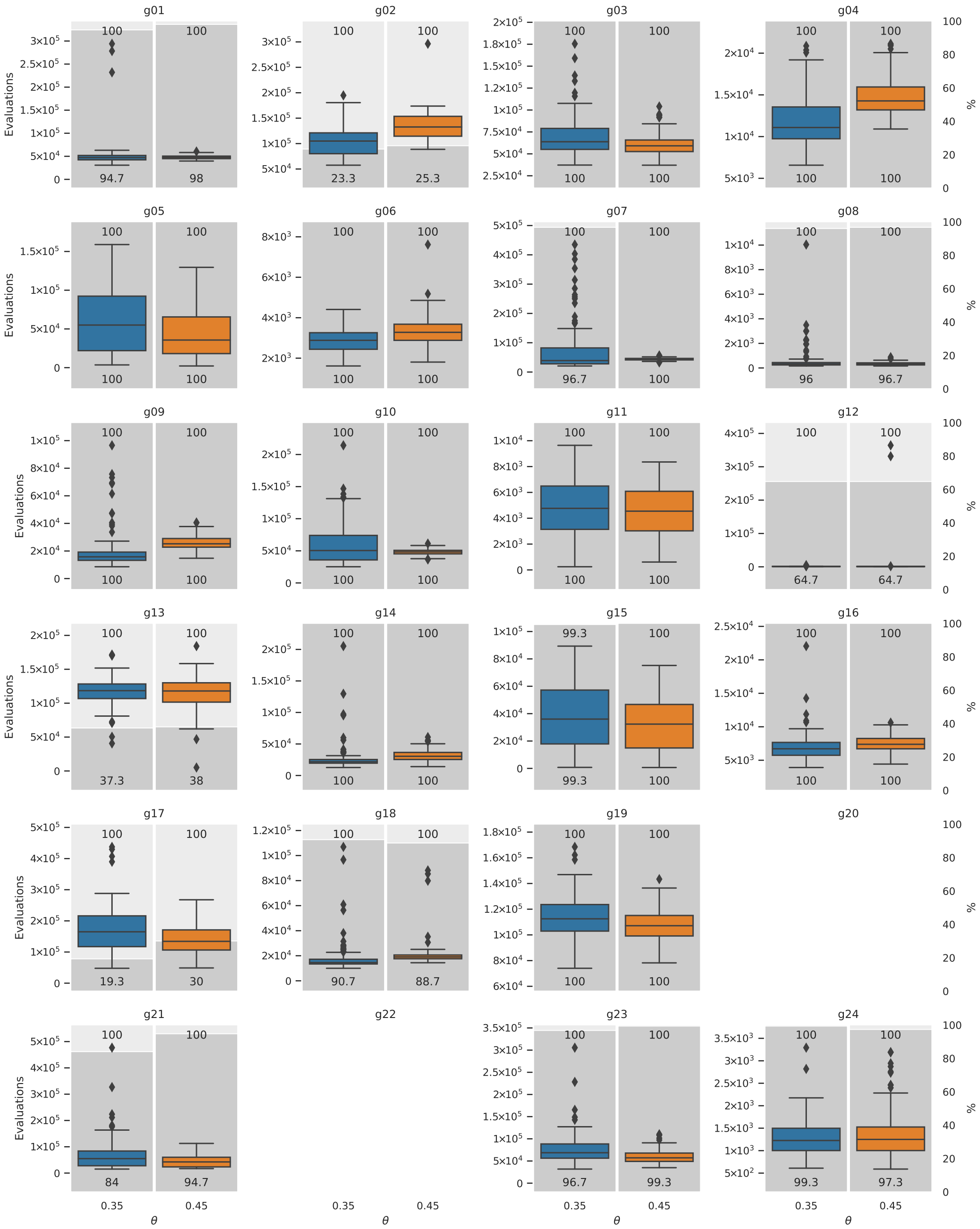


Figure A.13: The effect of the θ_f parameter of the DP-O method over all runs performed.

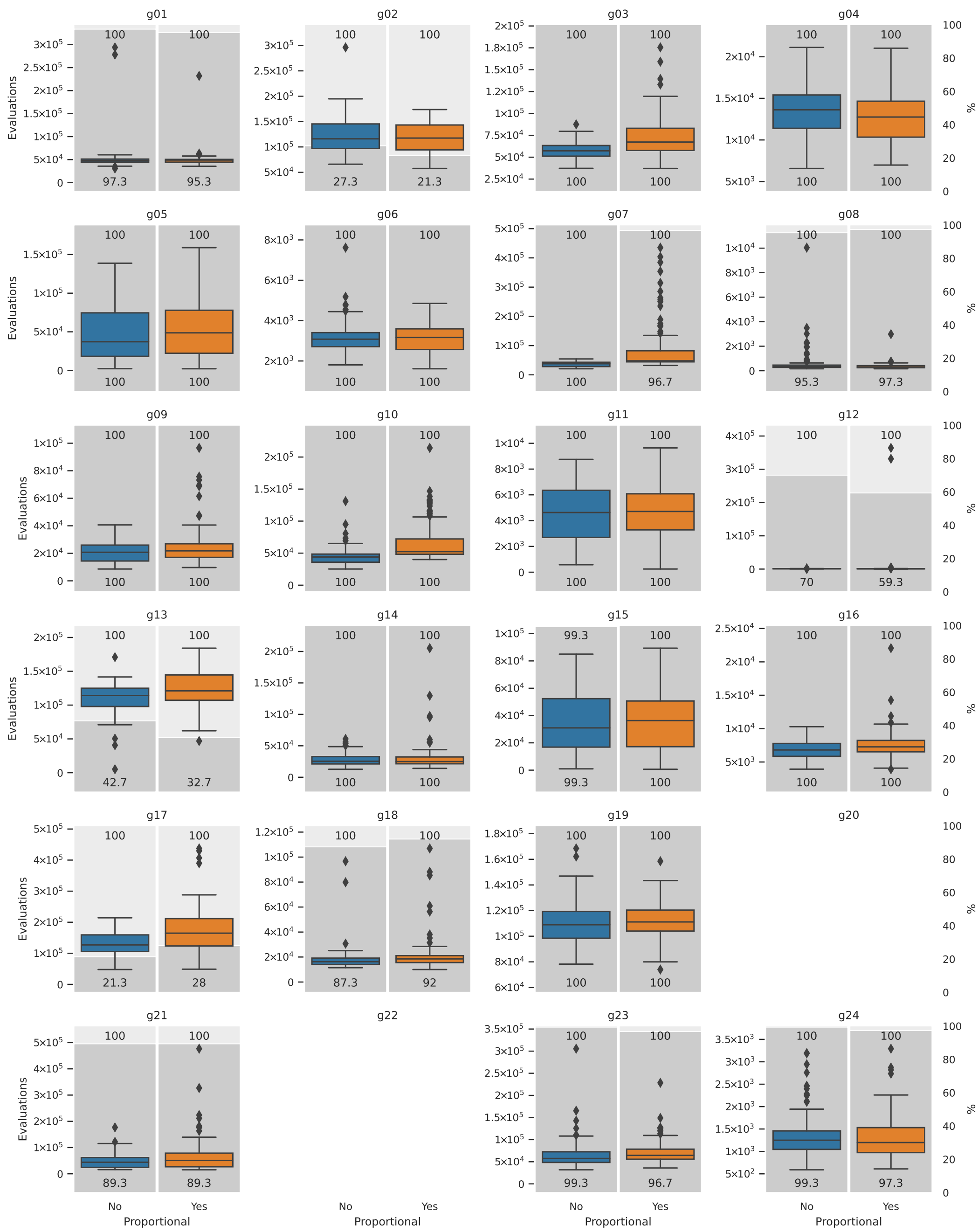


Figure A.14: The effect of the proportional or full scheme for the subpopulations of the DP-O method over all runs performed.

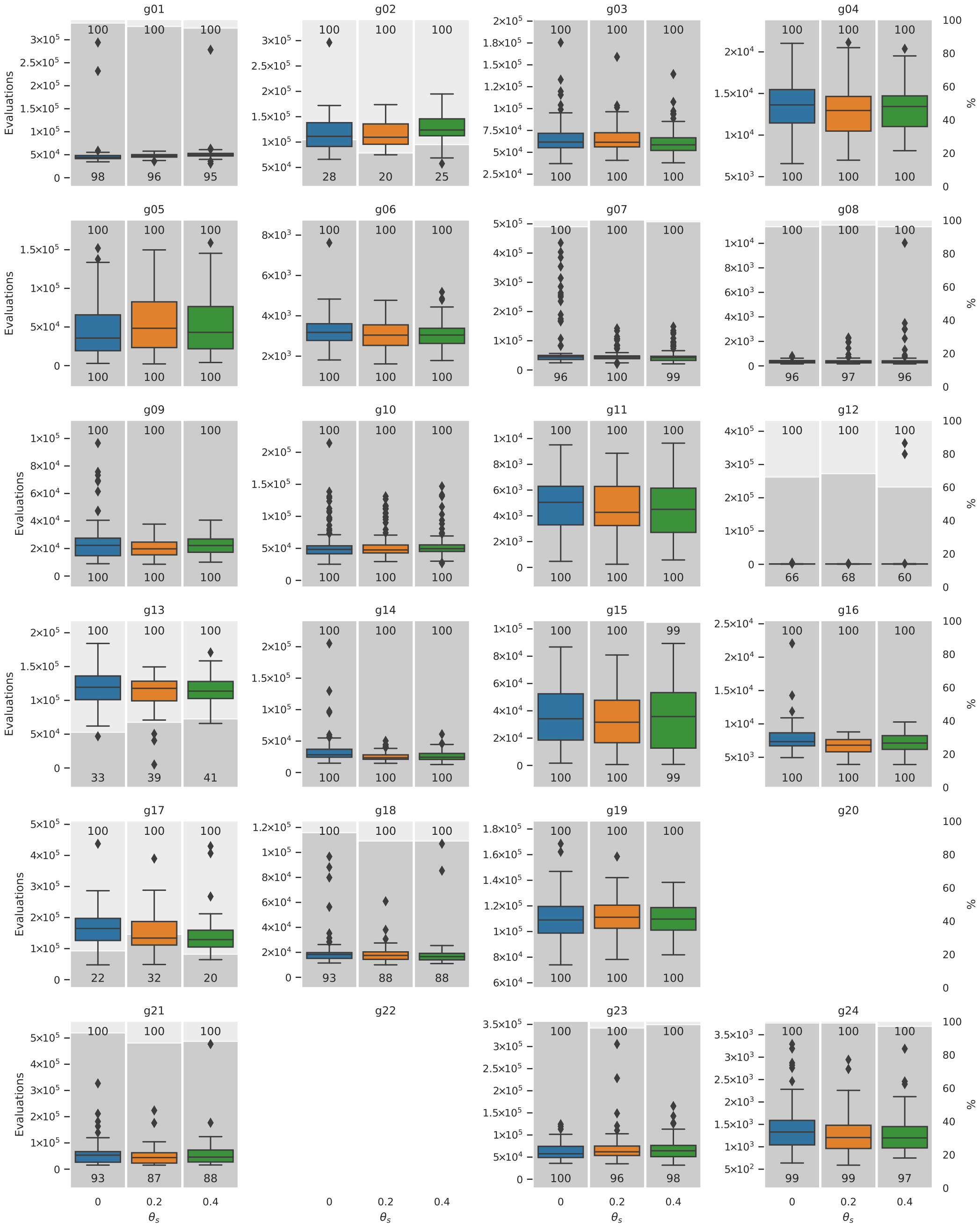


Figure A.15: The effect of the θ_s parameter of the DP-O method over all runs performed.

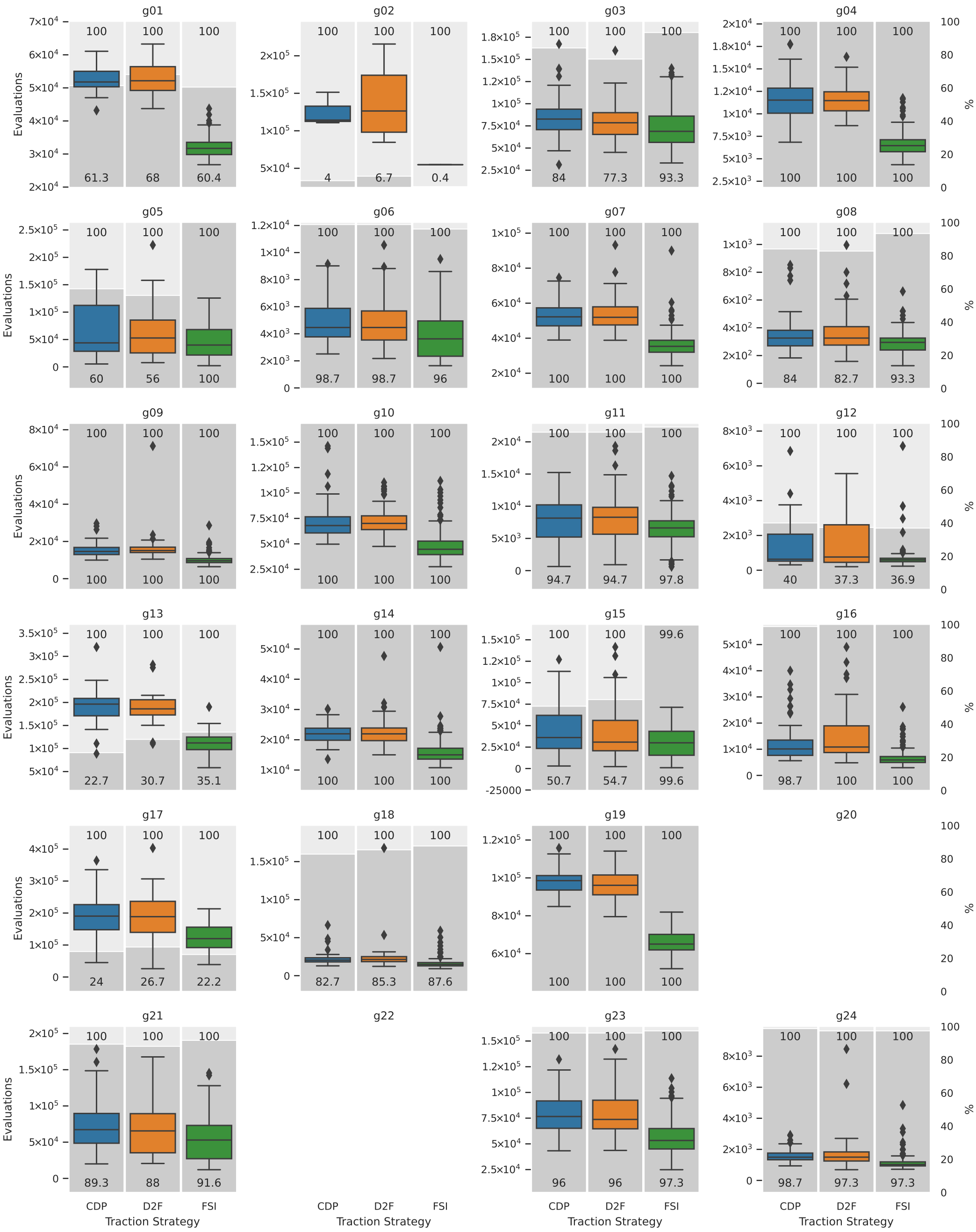


Figure A.16: The effect of the different traction strategies for the DP-T method over all runs performed. CDP is no traction strategy, D2F corresponds to the distance to the feasible population and FSI to the injection of the feasible selection.

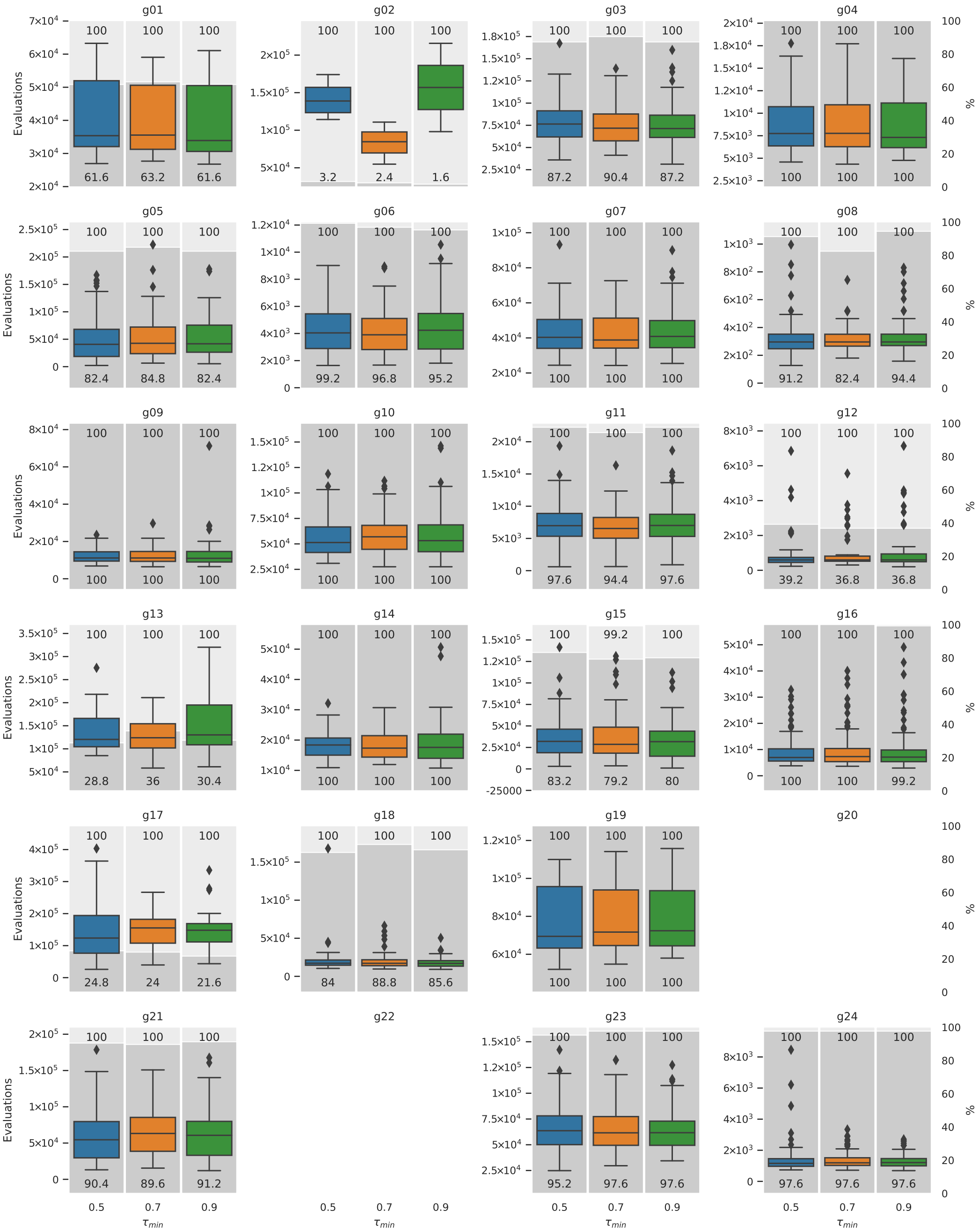


Figure A.17: The effect of the τ_{min} parameter of the DP-T method over all runs performed.

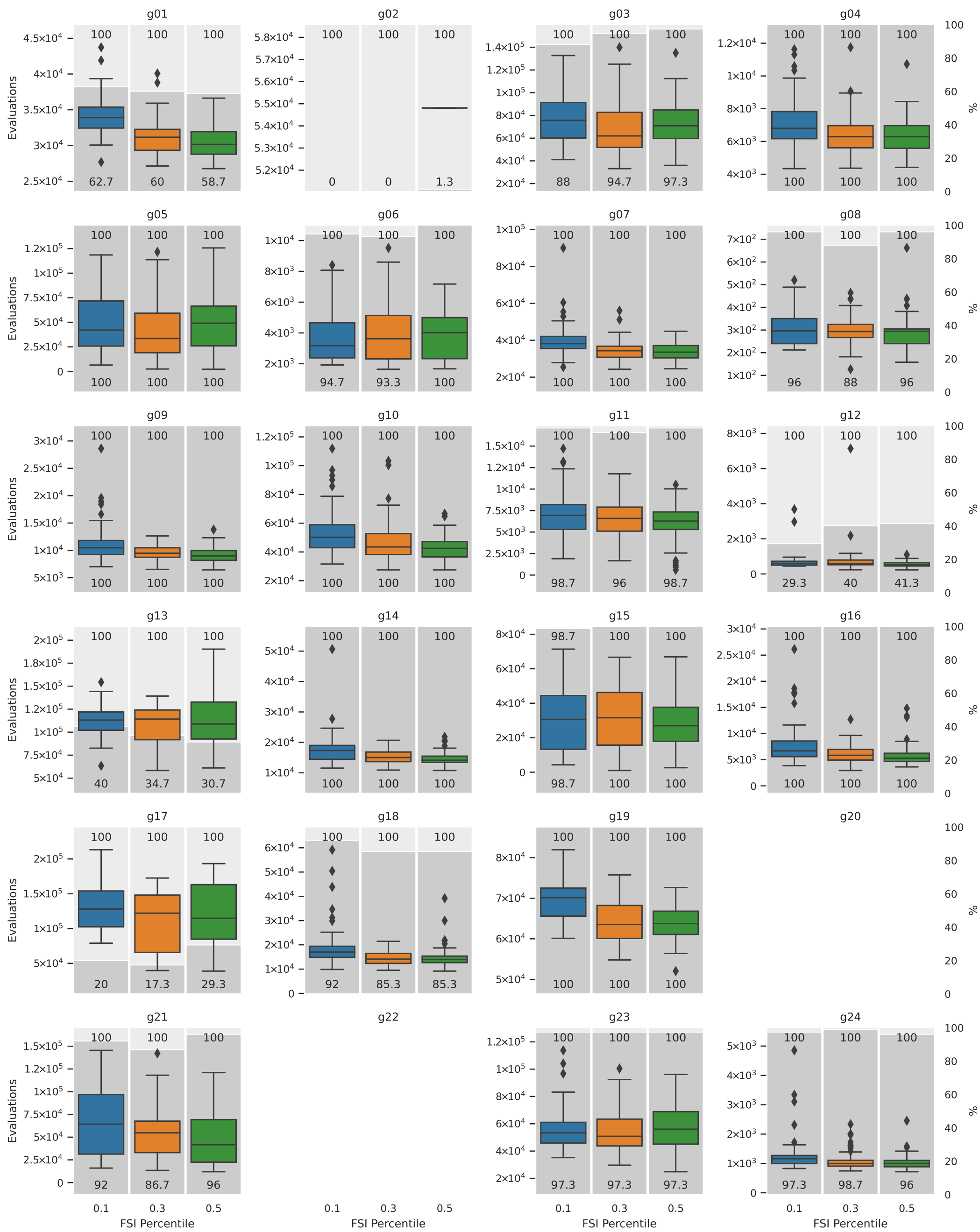


Figure A.18: The effect of the “feasible selection injection percentile”-parameter of the DP-T method over all runs performed.

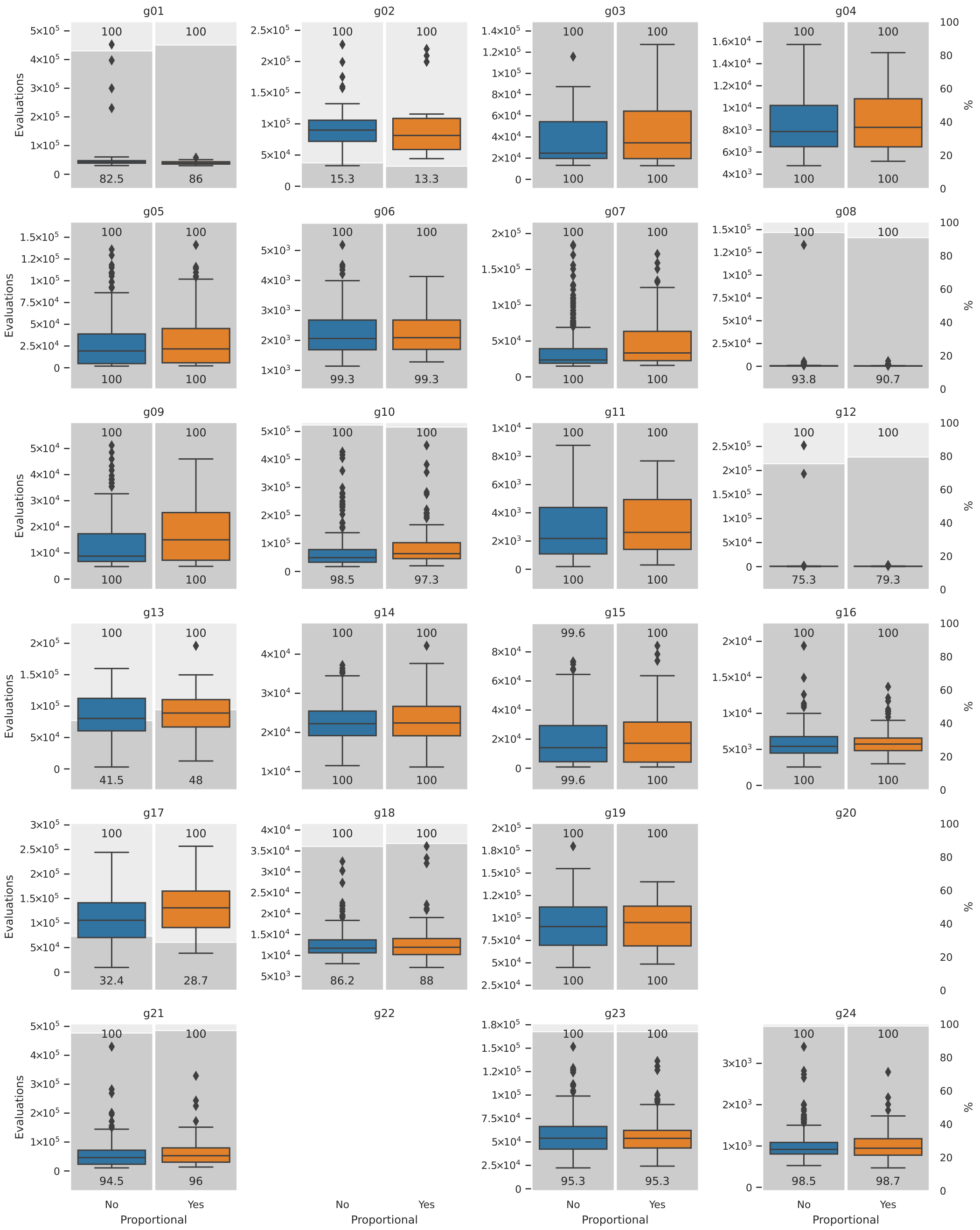


Figure A.19: The effect of the proportional and full subpopulation schemes of the PI method over all runs performed.

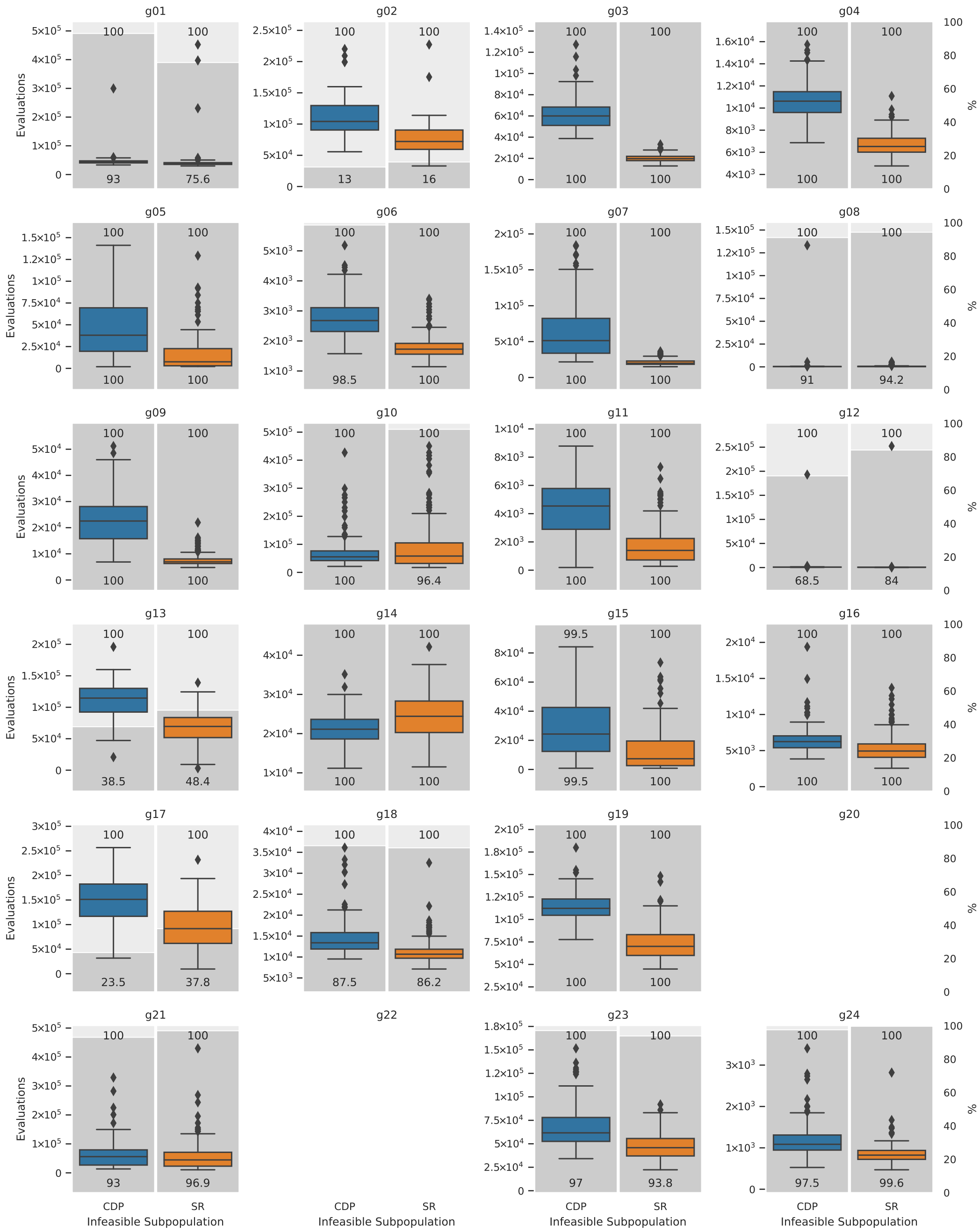


Figure A.20: The effect of the search strategy used in the infeasible (sub)populations of the PI method over all runs performed.

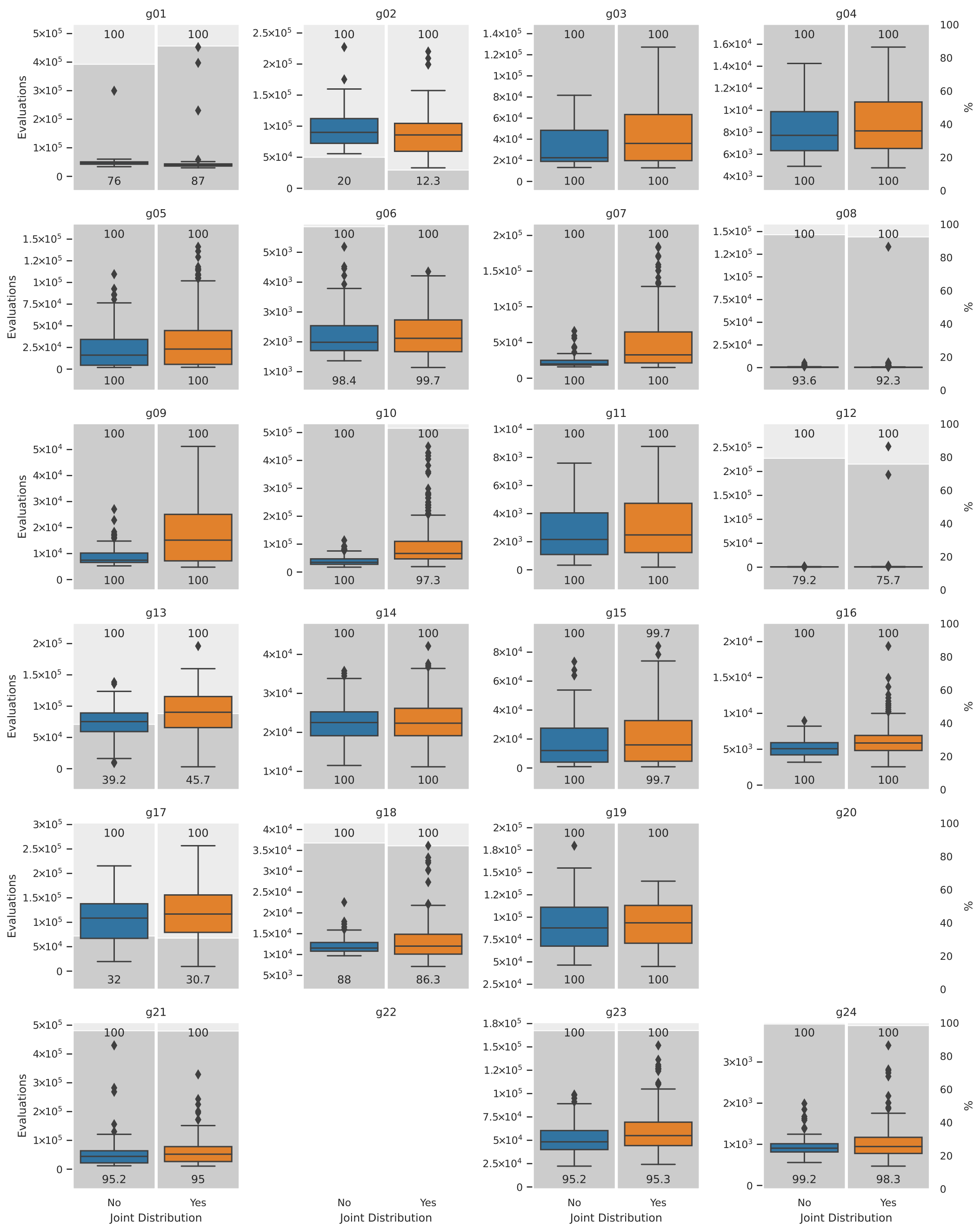


Figure A.21: The effect of using a single joint distribution vs. a distribution per subpopulation for the PI method over all runs performed.

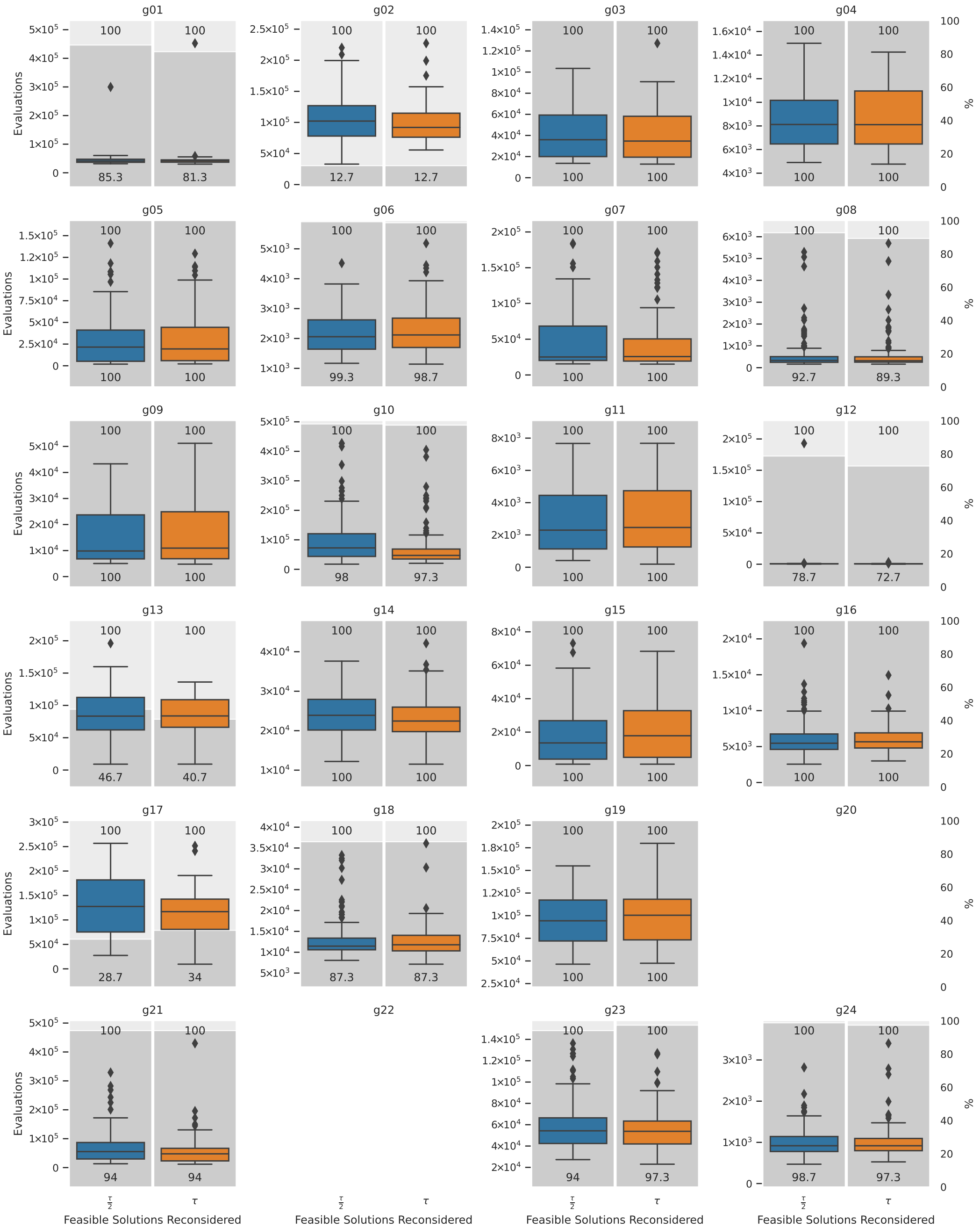


Figure A.22: The effect of reconsidering feasible solutions lost during variation for the PI method over all runs performed.

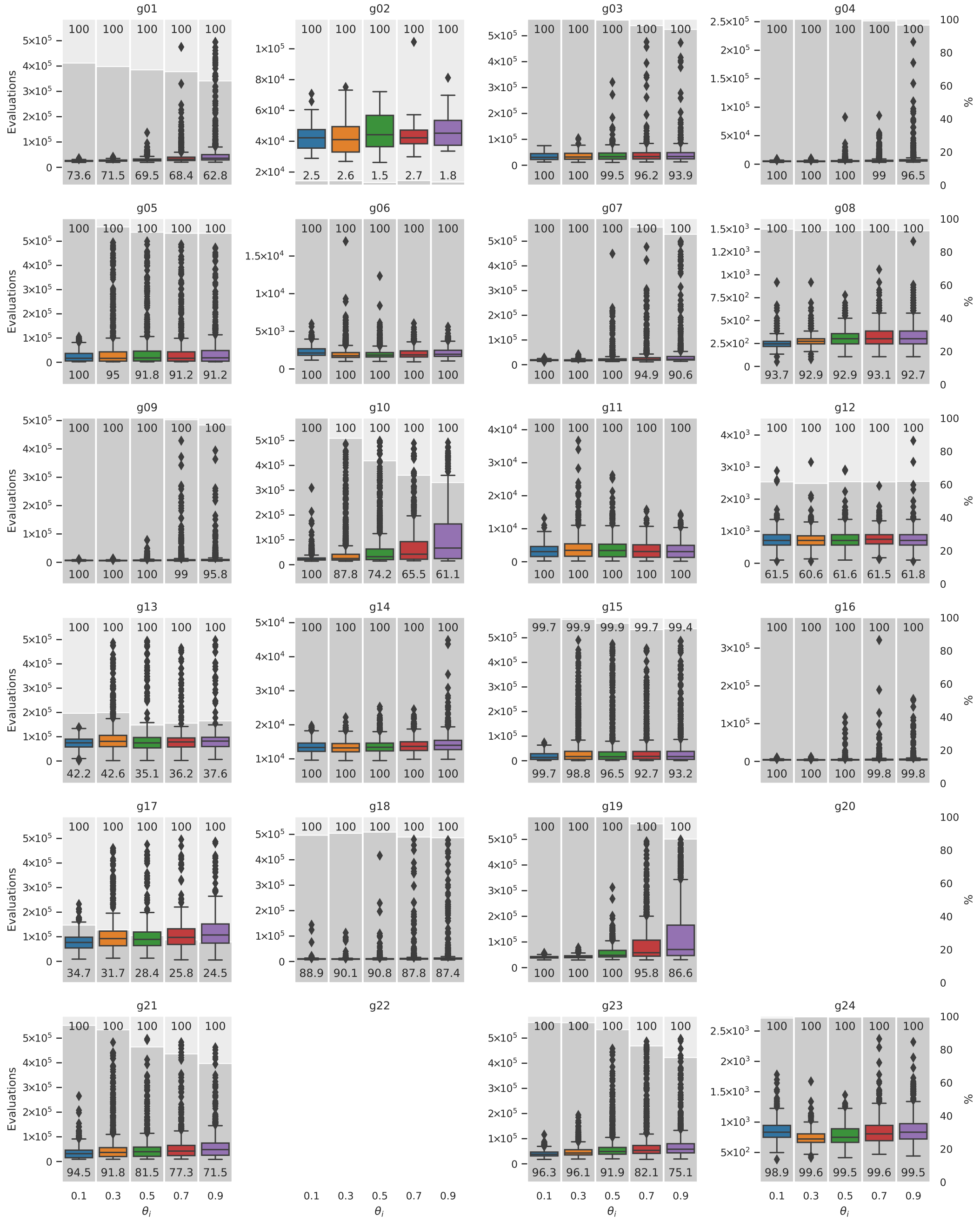


Figure A.23: The effect of the θ_i parameter of the PIS method over all runs performed.

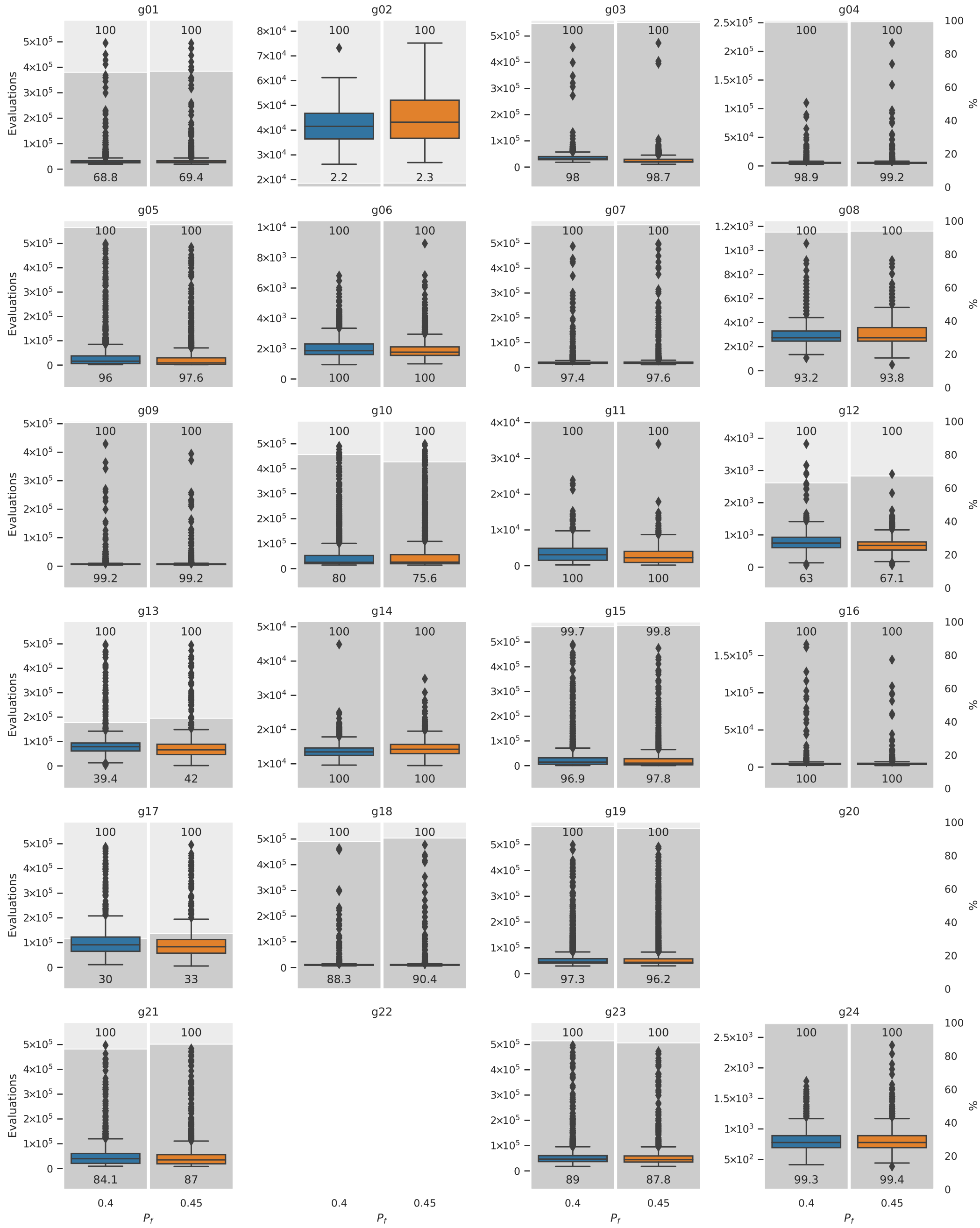


Figure A.24: The effect of the P_f parameter of the PIS method over all runs using SR for either searching in the infeasible region or selecting the infeasible solutions used.

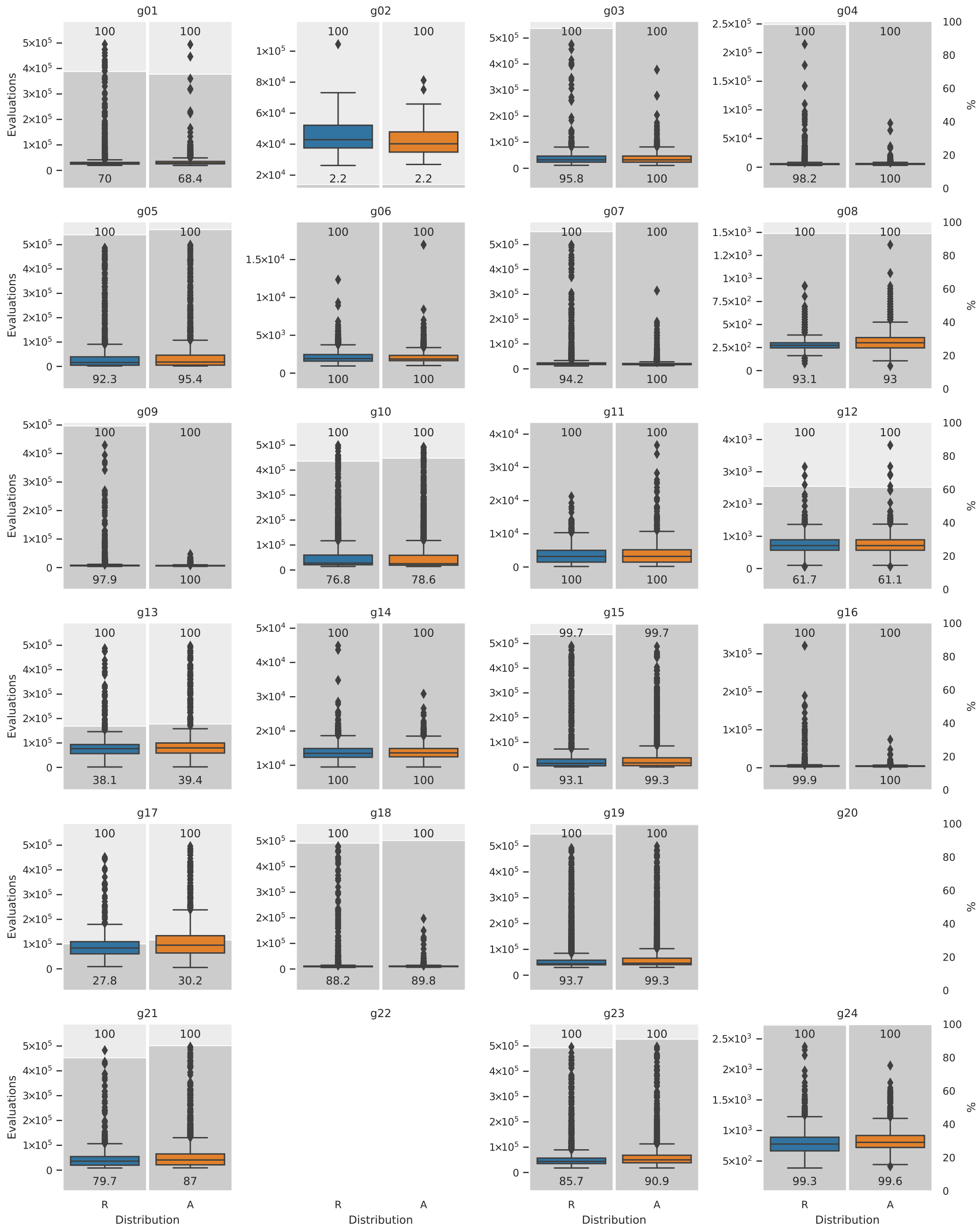


Figure A.25: The effect of replacing feasible solutions or adding the infeasible solutions for PIS method over all runs performed.

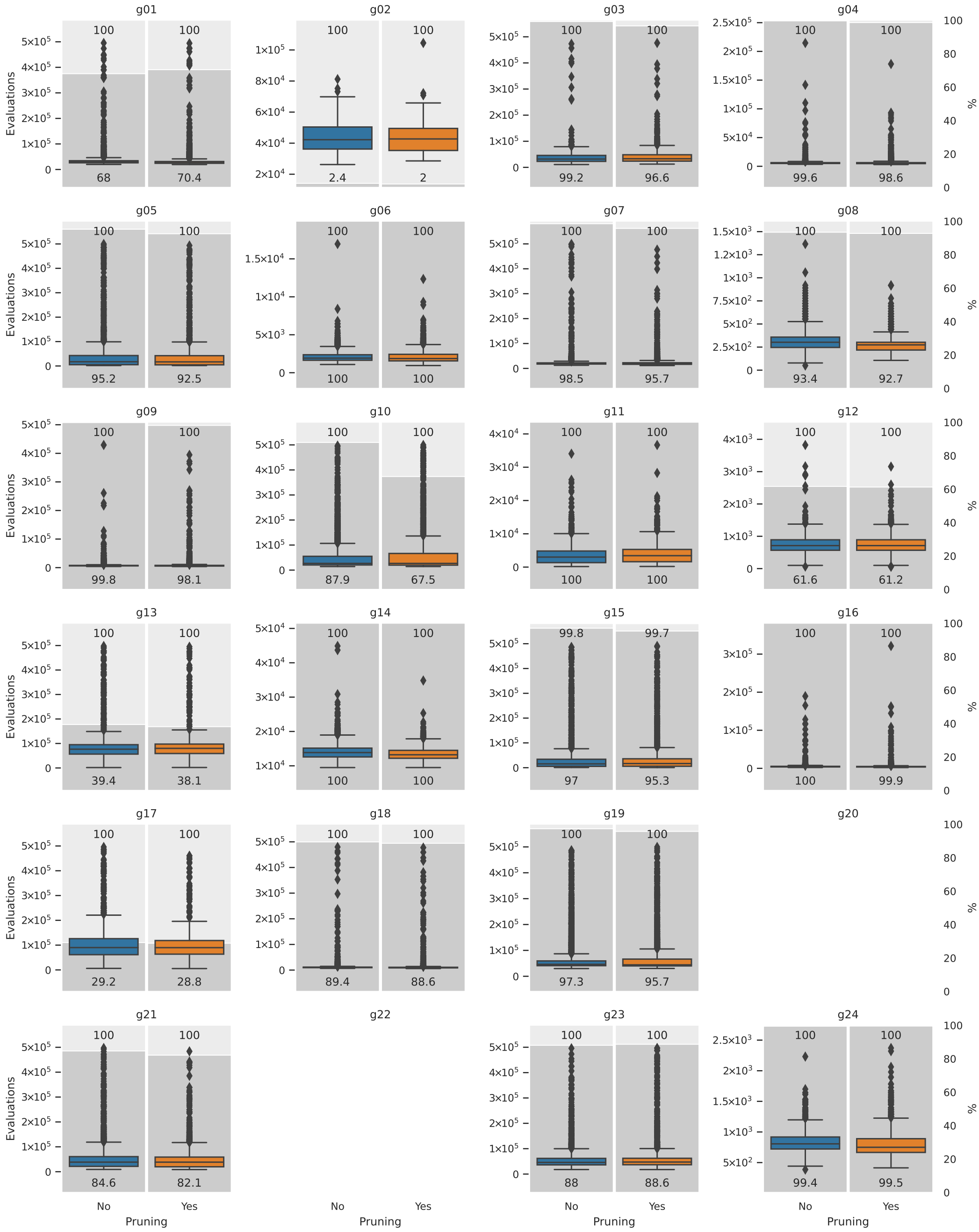


Figure A.26: The effect of pruning for the PIS method over all runs performed.

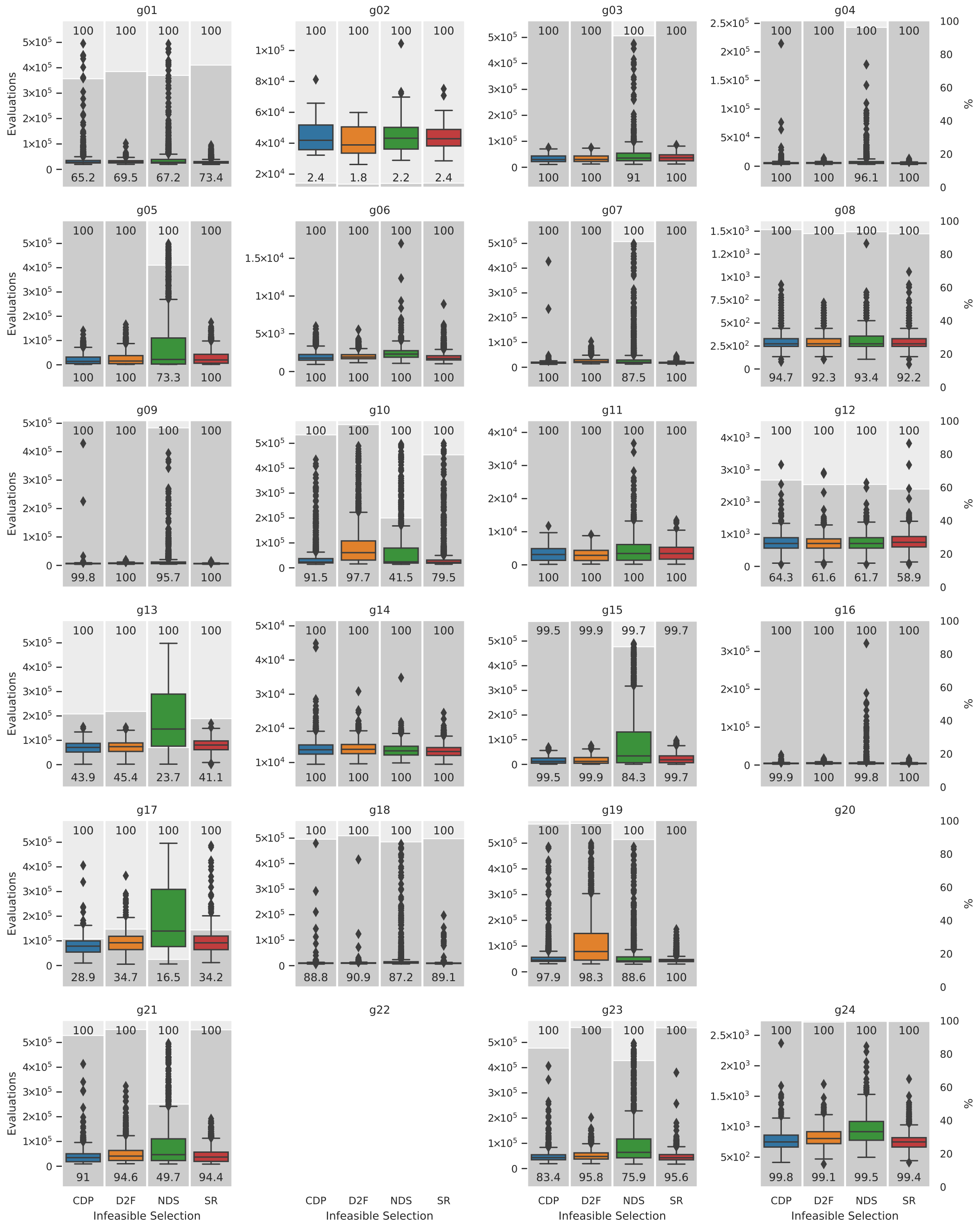


Figure A.27: The effect of the strategy used to select the infeasible solutions for the PIS method over all runs performed. D2F corresponds to selection based on the proximity to the feasible solutions and NDS to non-dominated-sorting.

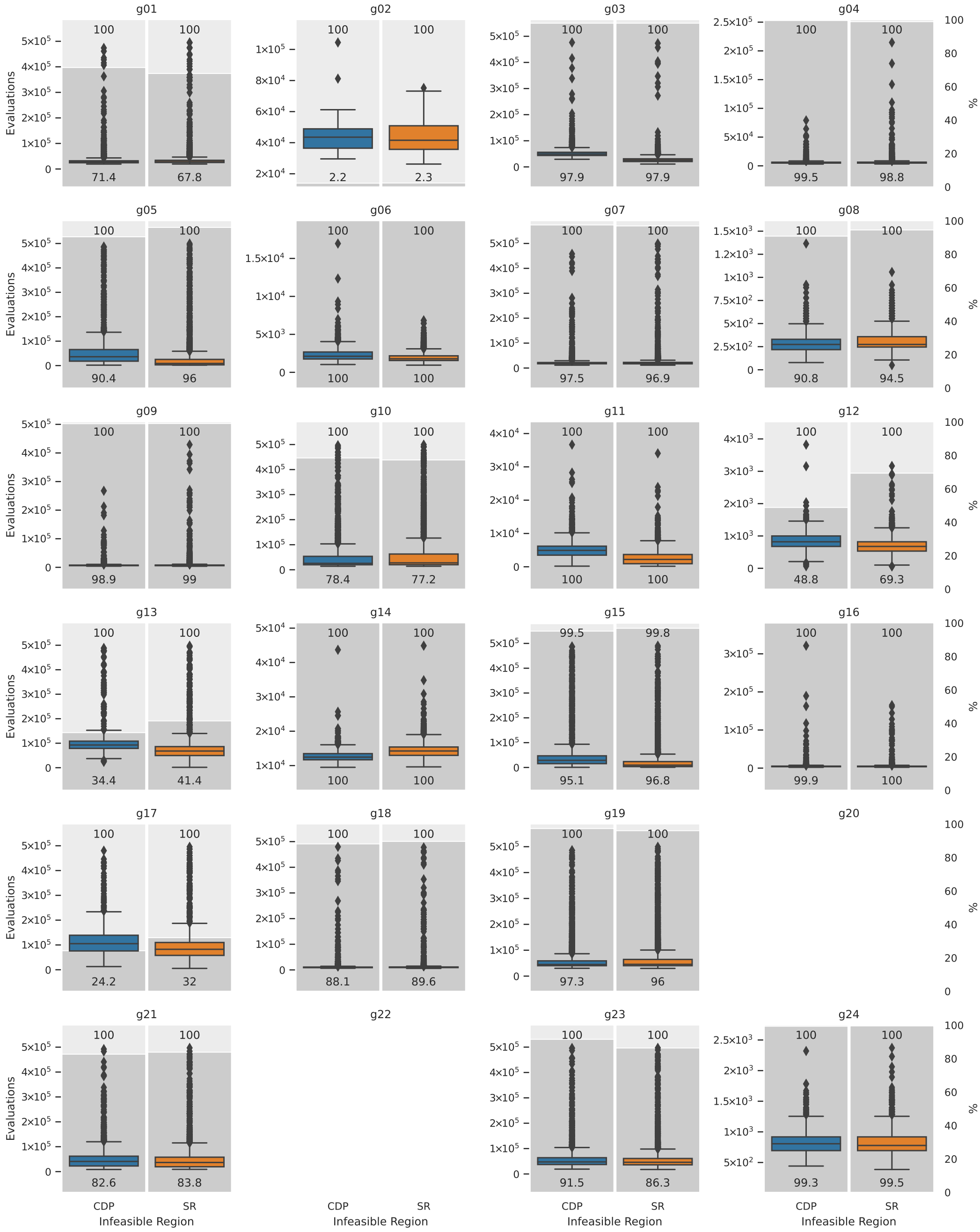


Figure A.28: The effect of using SR in the infeasible region for the PIS method over all runs performed.

Appendix B

CEC2006 Results

	Feasible Rate	Success Rate	Min	Median	Max	Mean	Std	Success Performance
g01	100.0%	88.0%	-14.9999	-14.9999	-13.0000	-14.7930	0.5882	28845.6
g02	100.0%	0.0%	-0.7862	-0.7409	-0.4896	-0.6992	0.1025	-
g03	0.0%	0.0%	-100.0340	-9.9875	-0.0014	-24.0469	27.8272	-
g04	40.0%	0.0%	-30169.1074	-29386.0451	-28584.9035	-29387.7576	383.3031	-
g05	100.0%	100.0%	5126.4967	5126.4968	5126.4968	5126.4968	0.0000	4342.5
g06	96.0%	0.0%	-6823.3214	-6405.2261	-4423.4972	-5951.5698	792.4876	-
g07	100.0%	100.0%	24.3063	24.3063	24.3063	24.3063	0.0000	18835.8
g08	100.0%	84.0%	-0.0958	-0.0958	-0.0258	-0.0889	0.0182	15996.4
g09	100.0%	100.0%	680.6301	680.6301	680.6302	680.6301	0.0000	5582.9
g10	0.0%	0.0%	2100.0000	2100.0000	17744.7398	6003.0405	5205.8310	-
g11	100.0%	100.0%	0.7499	0.7500	0.7500	0.7499	0.0000	836.3
g12	100.0%	92.0%	-1.0000	-0.9999	-0.9944	-0.9995	0.0015	277.0
g13	100.0%	36.0%	0.0540	0.4388	1.0000	0.3227	0.2337	157735.5
g14	100.0%	100.0%	-47.7648	-47.7648	-47.7648	-47.7648	0.0000	11155.2
g15	100.0%	100.0%	961.7150	961.7151	961.7151	961.7151	0.0000	3667.3
g16	100.0%	100.0%	-1.9051	-1.9051	-1.9051	-1.9051	0.0000	4194.8
g17	100.0%	24.0%	8853.5372	8927.5917	8927.5917	8909.8189	32.2792	105107.6
g18	100.0%	92.0%	-0.8660	-0.8659	-0.1275	-0.8076	0.2020	11133.4
g19	100.0%	100.0%	32.6556	32.6557	32.6557	32.6557	0.0000	37341.5
g21	44.0%	44.0%	100.9286	193.7246	382.6867	213.3046	68.9909	26563.4
g22	0.0%	0.0%	874.6729	9316.9878	19999.8274	10128.2893	6337.7855	-
g23	0.0%	0.0%	-1628.1900	-784.3309	2.8079	-765.0414	370.0766	-
g24	100.0%	100.0%	-5.5080	-5.5080	-5.5079	-5.5080	0.0000	857.2

Table B.1: The CEC2006 results for the AL method.

	Feasible Rate	Success Rate	Min	Median	Max	Mean	Std	Success Performance
g01	100.0%	76.0%	-14.9999	-14.9999	-13.0000	-14.6524	0.6606	32534.5
g02	100.0%	0.0%	-0.7947	-0.7649	-0.5888	-0.7403	0.0523	-
g03	100.0%	100.0%	-1.0005	-1.0004	-1.0004	-1.0004	0.0000	48991.6
g04	100.0%	100.0%	-30665.5386	-30665.5386	-30665.5386	-30665.5386	0.0000	5423.7
g05	100.0%	100.0%	5126.4967	5126.4968	5126.4968	5126.4968	0.0000	35441.0
g06	100.0%	100.0%	-6961.8139	-6961.8138	-6961.8138	-6961.8138	0.0000	2300.1
g07	100.0%	100.0%	24.3062	24.3063	24.3063	24.3063	0.0000	20423.8
g08	100.0%	100.0%	-0.0958	-0.0958	-0.0957	-0.0958	0.0000	232.6
g09	100.0%	100.0%	680.6301	680.6301	680.6302	680.6301	0.0000	7422.0
g10	100.0%	100.0%	7049.2481	7049.2481	7049.2481	7049.2481	0.0000	24388.0
g11	100.0%	100.0%	0.7499	0.7500	0.7500	0.7500	0.0000	4083.8
g12	100.0%	44.0%	-1.0000	-0.9944	-0.9694	-0.9942	0.0077	1633.1
g13	100.0%	32.0%	0.0540	0.4388	0.4388	0.3157	0.1832	280714.8
g14	100.0%	100.0%	-47.7648	-47.7648	-47.7648	-47.7648	0.0000	12976.7
g15	100.0%	100.0%	961.7150	961.7151	961.7151	961.7151	0.0000	23149.6
g16	100.0%	100.0%	-1.9051	-1.9051	-1.9051	-1.9051	0.0000	4490.0
g17	100.0%	16.0%	8853.5389	8927.5917	8927.5917	8915.7434	27.7079	473603.1
g18	100.0%	84.0%	-0.8660	-0.8659	-0.5000	-0.8284	0.0931	11794.3
g19	100.0%	100.0%	32.6557	32.6557	32.6557	32.6557	0.0000	42077.3
g21	100.0%	100.0%	193.7246	193.7246	193.7246	193.7246	0.0000	35539.8
g22	0.0%	0.0%	1160.2579	7974.7061	19996.2146	9408.5501	6954.9993	-
g23	100.0%	100.0%	-400.0550	-400.0550	-400.0550	-400.0550	0.0000	39846.2
g24	100.0%	100.0%	-5.5080	-5.5080	-5.5079	-5.5080	0.0000	836.2

Table B.2: The CEC2006 results for the CDP method.

	Feasible Rate	Success Rate	Min	Median	Max	Mean	Std	Success Performance
g01	100.0%	92.0%	-14.9999	-14.9999	-12.4531	-14.8043	0.6777	69020.7
g02	100.0%	20.0%	-0.8035	-0.7781	-0.4156	-0.7530	0.0786	557860.0
g03	100.0%	100.0%	-1.0005	-1.0004	-1.0004	-1.0004	0.0000	62014.8
g04	100.0%	100.0%	-30665.5386	-30665.5386	-30665.5386	-30665.5386	0.0000	11176.2
g05	100.0%	100.0%	5126.4967	5126.4968	5126.4968	5126.4968	0.0000	50261.9
g06	100.0%	100.0%	-6961.8139	-6961.8138	-6961.8138	-6961.8138	0.0000	2938.3
g07	100.0%	100.0%	24.3063	24.3063	24.3063	24.3063	0.0000	36153.3
g08	100.0%	92.0%	-0.0958	-0.0958	-0.0291	-0.0904	0.0185	701.9
g09	100.0%	100.0%	680.6301	680.6301	680.6302	680.6301	0.0000	12865.4
g10	100.0%	100.0%	7049.2481	7049.2481	7049.2481	7049.2481	0.0000	41161.4
g11	100.0%	100.0%	0.7499	0.7500	0.7500	0.7500	0.0000	4654.7
g12	100.0%	72.0%	-1.0000	-0.9999	-0.9694	-0.9974	0.0063	1207.3
g13	100.0%	32.0%	0.0540	0.4388	1.0000	0.3381	0.2278	318559.8
g14	100.0%	100.0%	-47.7648	-47.7648	-47.7648	-47.7648	0.0000	21133.4
g15	100.0%	100.0%	961.7150	961.7151	961.7151	961.7151	0.0000	28744.8
g16	100.0%	100.0%	-1.9051	-1.9051	-1.9051	-1.9051	0.0000	5725.7
g17	100.0%	24.0%	8853.5378	8927.5917	8927.5917	8909.8190	32.2790	533517.4
g18	100.0%	84.0%	-0.8660	-0.8659	-0.6742	-0.8353	0.0715	19425.9
g19	100.0%	100.0%	32.6557	32.6557	32.6557	32.6557	0.0000	104095.6
g21	100.0%	76.0%	193.7246	193.7246	324.7028	198.9761	26.1931	76180.1
g22	0.0%	0.0%	773.3236	8239.2398	19999.8798	9914.5573	8204.8080	-
g23	100.0%	96.0%	-400.0550	-400.0550	-372.7851	-398.9642	5.4540	75848.0
g24	100.0%	100.0%	-5.5080	-5.5079	-5.5079	-5.5079	0.0000	1305.4

Table B.3: The CEC2006 results for the DP-O method.

	Feasible Rate	Success Rate	Min	Median	Max	Mean	Std	Success Performance
g01	100.0%	64.0%	-14.9999	-14.9999	-12.6562	-14.3712	0.9149	48083.0
g02	100.0%	0.0%	-0.7857	-0.7202	-0.4492	-0.6852	0.0992	-
g03	100.0%	100.0%	-1.0004	-1.0004	-1.0004	-1.0004	0.0000	76399.6
g04	100.0%	100.0%	-30665.5386	-30665.5386	-30665.5386	-30665.5386	0.0000	5886.8
g05	100.0%	100.0%	5126.4968	5126.4968	5126.4968	5126.4968	0.0000	53809.5
g06	100.0%	100.0%	-6961.8139	-6961.8138	-6961.8138	-6961.8138	0.0000	3420.3
g07	100.0%	100.0%	24.3063	24.3063	24.3063	24.3063	0.0000	34849.0
g08	100.0%	96.0%	-0.0958	-0.0958	-0.0291	-0.0931	0.0133	299.5
g09	100.0%	100.0%	680.6301	680.6301	680.6302	680.6301	0.0000	9711.4
g10	100.0%	100.0%	7049.2481	7049.2481	7049.2481	7049.2481	0.0000	43585.7
g11	100.0%	100.0%	0.7499	0.7500	0.7500	0.7500	0.0000	5992.2
g12	100.0%	56.0%	-1.0000	-0.9999	-0.9694	-0.9949	0.0079	1758.0
g13	100.0%	56.0%	0.0540	0.0540	0.4388	0.2233	0.1949	202039.3
g14	100.0%	100.0%	-47.7648	-47.7648	-47.7648	-47.7648	0.0000	13939.4
g15	100.0%	100.0%	961.7150	961.7151	961.7151	961.7151	0.0000	26324.1
g16	100.0%	100.0%	-1.9051	-1.9051	-1.9051	-1.9051	0.0000	5817.2
g17	100.0%	36.0%	8853.5374	8927.5917	8927.5917	8900.9326	36.2785	393863.9
g18	100.0%	68.0%	-0.8660	-0.8659	-0.2075	-0.7861	0.1485	21518.6
g19	100.0%	100.0%	32.6557	32.6557	32.6557	32.6557	0.0000	63982.0
g21	100.0%	96.0%	193.7246	193.7246	324.7028	198.9637	26.1956	47120.8
g22	0.0%	0.0%	742.3349	7001.3598	19999.5450	9079.5611	7605.3167	-
g23	100.0%	100.0%	-400.0550	-400.0550	-400.0550	-400.0550	0.0000	59739.2
g24	100.0%	100.0%	-5.5080	-5.5079	-5.5079	-5.5080	0.0000	1033.6

Table B.4: The CEC2006 results for the DP-T method.

	Feasible Rate	Success Rate	Min	Median	Max	Mean	Std	Success Performance
g01	100.0%	72.0%	-14.9999	-14.9999	-12.4531	-14.4843	0.8800	69261.8
g02	100.0%	0.0%	-0.7926	-0.7390	-0.5327	-0.7132	0.0766	-
g03	100.0%	100.0%	-1.0005	-1.0004	-1.0004	-1.0004	0.0000	42305.9
g04	100.0%	100.0%	-30665.5387	-30665.5386	-30665.5386	-30665.5386	0.0000	15208.4
g05	100.0%	100.0%	5126.4967	5126.4968	5126.4968	5126.4968	0.0000	47685.2
g06	72.0%	60.0%	-7950.9618	-6961.8139	-1502.1177	-6782.7563	1497.9954	81187.8
g07	100.0%	100.0%	24.3063	24.3063	24.3063	24.3063	0.0000	54946.4
g08	88.0%	48.0%	-0.0958	-0.0929	0.0055	-0.0644	0.0407	76251.0
g09	100.0%	100.0%	680.6301	680.6301	680.6302	680.6301	0.0000	46183.8
g10	100.0%	100.0%	7049.2481	7049.2481	7049.2481	7049.2481	0.0000	67341.2
g11	100.0%	100.0%	0.7499	0.7499	0.7500	0.7499	0.0000	36948.8
g12	100.0%	80.0%	-1.0000	-1.0000	-0.9944	-0.9988	0.0023	417.5
g13	100.0%	72.0%	0.0539	0.0540	0.4388	0.1617	0.1764	58816.0
g14	100.0%	100.0%	-47.7649	-47.7648	-47.7648	-47.7648	0.0000	47591.5
g15	68.0%	68.0%	961.7150	961.7150	967.9999	963.5532	2.8568	66322.1
g16	92.0%	92.0%	-1.9052	-1.9051	-1.4849	-1.8816	0.0892	50373.3
g17	44.0%	0.0%	4261.3242	8927.5917	10830.9799	8659.4456	1579.9275	-
g18	100.0%	100.0%	-0.8660	-0.8660	-0.8659	-0.8660	0.0000	56121.9
g19	100.0%	100.0%	32.6557	32.6557	32.6557	32.6557	0.0000	41715.4
g21	0.0%	0.0%	0.0000	0.0000	0.0000	0.0000	0.0000	-
g22	0.0%	0.0%	0.0000	2306.3093	19995.7715	5557.2647	7133.0110	-
g23	100.0%	76.0%	-400.0550	-400.0550	-100.0466	-328.0530	130.7706	74124.1
g24	100.0%	100.0%	-5.5080	-5.5079	-5.5079	-5.5080	0.0000	881.0

Table B.5: The CEC2006 results for the EPS method.

	Feasible Rate	Success Rate	Min	Median	Max	Mean	Std	Success Performance
g01	100.0%	60.0%	-14.9999	-14.9999	-12.6562	-14.3049	0.9220	81869.4
g02	100.0%	4.0%	-0.8035	-0.7357	-0.4441	-0.7229	0.0897	1247600.0
g03	100.0%	100.0%	-1.0005	-1.0004	-1.0004	-1.0004	0.0000	42230.6
g04	100.0%	100.0%	-30665.5387	-30665.5387	-30665.5386	-30665.5386	0.0000	46091.2
g05	100.0%	100.0%	5126.4967	5126.4967	5126.4968	5126.4968	0.0000	47402.7
g06	88.0%	88.0%	-7889.1022	-6961.8138	-805.2953	-6747.9738	1252.2152	55273.1
g07	100.0%	100.0%	24.3063	24.3063	24.3063	24.3063	0.0000	55357.8
g08	92.0%	56.0%	-0.0958	-0.0958	0.0547	-0.0658	0.0426	63779.1
g09	100.0%	100.0%	680.6301	680.6301	680.6302	680.6301	0.0000	45823.2
g10	100.0%	100.0%	7049.2481	7049.2481	7049.2481	7049.2481	0.0000	67014.8
g11	100.0%	100.0%	0.7499	0.7499	0.7500	0.7499	0.0000	35805.1
g12	100.0%	88.0%	-1.0000	-1.0000	-0.9864	-0.9990	0.0030	387.0
g13	100.0%	64.0%	0.0539	0.0540	0.4388	0.1925	0.1885	65765.9
g14	100.0%	100.0%	-47.7649	-47.7648	-47.7648	-47.7648	0.0000	48605.2
g15	60.0%	60.0%	958.9629	961.7150	971.1010	963.5488	3.0949	76320.0
g16	88.0%	88.0%	-1.9052	-1.9051	-0.8536	-1.8215	0.2503	53165.8
g17	40.0%	0.0%	5886.4351	8927.5917	11260.0404	8729.9659	1275.8442	-
g18	100.0%	100.0%	-0.8660	-0.8660	-0.8659	-0.8660	0.0000	67053.1
g19	100.0%	100.0%	32.6557	32.6557	32.6557	32.6557	0.0000	55055.8
g21	0.0%	0.0%	0.0000	0.0000	0.0000	0.0000	0.0000	-
g22	0.0%	0.0%	0.0000	195.1632	11967.5605	1882.7024	3061.8720	-
g23	100.0%	76.0%	-400.0550	-400.0550	-100.0466	-328.0530	130.7706	69606.2
g24	100.0%	100.0%	-5.5080	-5.5080	-5.5080	-5.5080	0.0000	44604.2

Table B.6: The CEC2006 results for the IEPS method.

	Feasible Rate	Success Rate	Min	Median	Max	Mean	Std	Success Performance
g01	100.0%	68.0%	-14.9999	-14.9999	-11.8281	-14.4649	0.8959	35442.7
g02	100.0%	8.0%	-0.8035	-0.7363	-0.5385	-0.7279	0.0640	565356.2
g03	100.0%	100.0%	-1.0005	-1.0004	-1.0004	-1.0004	0.0000	48844.8
g04	100.0%	100.0%	-30665.5386	-30665.5386	-30665.5386	-30665.5386	0.0000	5108.5
g05	100.0%	100.0%	5126.4968	5126.4968	5126.4968	5126.4968	0.0000	31839.2
g06	100.0%	100.0%	-6961.8139	-6961.8138	-6961.8138	-6961.8138	0.0000	2615.2
g07	100.0%	100.0%	24.3063	24.3063	24.3063	24.3063	0.0000	17652.1
g08	100.0%	92.0%	-0.0958	-0.0958	-0.0291	-0.0905	0.0185	260.5
g09	100.0%	100.0%	680.6301	680.6301	680.6302	680.6301	0.0000	6976.6
g10	100.0%	100.0%	7049.2481	7049.2481	7049.2481	7049.2481	0.0000	22266.8
g11	100.0%	100.0%	0.7499	0.7500	0.7500	0.7500	0.0000	5017.7
g12	100.0%	48.0%	-1.0000	-0.9944	-0.9694	-0.9954	0.0068	1726.9
g13	100.0%	36.0%	0.0540	0.4388	0.4388	0.3003	0.1885	312176.2
g14	100.0%	100.0%	-47.7649	-47.7648	-47.7648	-47.7648	0.0000	12225.8
g15	100.0%	100.0%	961.7150	961.7151	961.7151	961.7151	0.0000	21947.8
g16	100.0%	100.0%	-1.9051	-1.9051	-1.9051	-1.9051	0.0000	4349.1
g17	100.0%	28.0%	8853.5357	8927.5917	8927.5917	8906.8566	33.9358	380642.9
g18	100.0%	76.0%	-0.8660	-0.8659	-0.6750	-0.8201	0.0832	12809.0
g19	100.0%	100.0%	32.6557	32.6557	32.6557	32.6557	0.0000	39519.7
g21	100.0%	88.0%	193.7246	193.7246	324.7028	209.4420	43.4406	41402.5
g22	0.0%	0.0%	725.7620	11589.0201	19999.9993	10580.0386	6452.2904	-
g23	100.0%	100.0%	-400.0550	-400.0550	-400.0550	-400.0550	0.0000	43020.8
g24	100.0%	96.0%	-5.5080	-5.5079	-4.4200	-5.4644	0.2176	852.6

Table B.7: The CEC2006 results for the R-BS method.

	Feasible Rate	Success Rate	Min	Median	Max	Mean	Std	Success Performance
g01	100.0%	84.0%	-14.9999	-14.9999	-11.2813	-14.6555	0.9151	33550.9
g02	100.0%	4.0%	-0.8035	-0.7649	-0.4416	-0.7203	0.1014	1155600.0
g03	100.0%	100.0%	-1.0004	-1.0004	-1.0004	-1.0004	0.0000	57152.8
g04	100.0%	100.0%	-30665.5386	-30665.5386	-30665.5386	-30665.5386	0.0000	6059.8
g05	100.0%	100.0%	5126.4968	5126.4968	5126.4968	5126.4968	0.0000	47221.4
g06	100.0%	100.0%	-6961.8139	-6961.8138	-6961.8138	-6961.8138	0.0000	2815.1
g07	100.0%	100.0%	24.3063	24.3063	24.3063	24.3063	0.0000	27016.0
g08	100.0%	88.0%	-0.0958	-0.0958	-0.0258	-0.0875	0.0228	290.4
g09	100.0%	100.0%	680.6301	680.6301	680.6302	680.6301	0.0000	9284.3
g10	100.0%	100.0%	7049.2481	7049.2481	7049.2481	7049.2481	0.0000	27568.0
g11	100.0%	100.0%	0.7499	0.7499	0.7500	0.7499	0.0000	5168.0
g12	100.0%	68.0%	-1.0000	-0.9999	-0.9694	-0.9962	0.0084	1294.4
g13	100.0%	28.0%	0.0540	0.4388	0.4388	0.3311	0.1763	441991.8
g14	100.0%	100.0%	-47.7649	-47.7648	-47.7648	-47.7648	0.0000	14590.1
g15	100.0%	100.0%	961.7150	961.7151	961.7151	961.7151	0.0000	32188.8
g16	100.0%	100.0%	-1.9051	-1.9051	-1.9051	-1.9051	0.0000	5360.6
g17	100.0%	24.0%	8853.5386	8927.5917	8927.5917	8909.8191	32.2788	603310.4
g18	100.0%	92.0%	-0.8660	-0.8659	-0.6750	-0.8507	0.0529	13908.1
g19	100.0%	100.0%	32.6557	32.6557	32.6557	32.6557	0.0000	51850.5
g21	100.0%	96.0%	193.7246	193.7246	324.7028	198.9637	26.1956	49774.4
g22	0.0%	0.0%	712.6385	9475.3043	19988.3437	10342.7557	7523.7896	-
g23	100.0%	96.0%	-400.0550	-400.0550	-100.0466	-388.0547	60.0017	49666.3
g24	100.0%	100.0%	-5.5080	-5.5079	-5.5079	-5.5079	0.0000	991.7

Table B.8: The CEC2006 results for the R-R method.

	Feasible Rate	Success Rate	Min	Median	Max	Mean	Std	Success Performance
g01	100.0%	80.0%	-14.9999	-14.9999	-13.8281	-14.7656	0.4784	32305.1
g02	100.0%	4.0%	-0.8035	-0.7422	-0.4511	-0.7037	0.1020	1122350.0
g03	100.0%	100.0%	-1.0004	-1.0004	-1.0004	-1.0004	0.0000	20043.7
g04	100.0%	100.0%	-30665.5386	-30665.5386	-30665.5386	-30665.5386	0.0000	5039.8
g05	100.0%	100.0%	5126.4968	5126.4968	5126.4968	5126.4968	0.0000	10908.9
g06	100.0%	100.0%	-6961.8139	-6961.8138	-6961.8138	-6961.8138	0.0000	1839.1
g07	100.0%	100.0%	24.3063	24.3063	24.3063	24.3063	0.0000	20200.4
g08	100.0%	96.0%	-0.0958	-0.0958	-0.0291	-0.0931	0.0133	274.5
g09	100.0%	100.0%	680.6301	680.6301	680.6302	680.6301	0.0000	7160.1
g10	100.0%	100.0%	7049.2481	7049.2481	7049.2481	7049.2481	0.0000	21621.0
g11	100.0%	100.0%	0.7499	0.7500	0.7500	0.7500	0.0000	1148.2
g12	100.0%	84.0%	-1.0000	-0.9999	-0.9864	-0.9987	0.0032	839.7
g13	100.0%	52.0%	0.0540	0.0540	0.4388	0.2387	0.1962	70190.5
g14	100.0%	100.0%	-47.7648	-47.7648	-47.7648	-47.7648	0.0000	15784.0
g15	96.0%	96.0%	961.7151	961.7151	967.9999	961.9665	1.2570	7200.4
g16	100.0%	100.0%	-1.9051	-1.9051	-1.9051	-1.9051	0.0000	4449.8
g17	100.0%	48.0%	8853.5374	8927.5917	8927.5917	8892.0463	37.7598	128697.6
g18	100.0%	84.0%	-0.8660	-0.8659	-0.6750	-0.8354	0.0714	12342.7
g19	100.0%	100.0%	32.6557	32.6557	32.6557	32.6557	0.0000	42052.8
g21	100.0%	96.0%	193.7246	193.7246	324.7028	198.9637	26.1956	28786.4
g22	0.0%	0.0%	0.0000	0.0000	0.0000	0.0000	0.0000	-
g23	100.0%	100.0%	-400.0550	-400.0550	-400.0550	-400.0550	0.0000	37625.1
g24	100.0%	100.0%	-5.5080	-5.5079	-5.5079	-5.5079	0.0000	787.0

Table B.9: The CEC2006 results for the SR method.

	Feasible Rate	Success Rate	Min	Median	Max	Mean	Std	Success Performance
g01	100.0%	84.0%	-14.9999	-14.9999	-12.6562	-14.7655	0.5859	31127.6
g02	100.0%	4.0%	-0.8035	-0.7502	-0.4619	-0.6964	0.1156	722775.0
g03	100.0%	100.0%	-1.0005	-1.0004	-1.0004	-1.0004	0.0000	19398.3
g04	100.0%	100.0%	-30665.5386	-30665.5386	-30665.5386	-30665.5386	0.0000	4622.8
g05	100.0%	100.0%	5126.4968	5126.4968	5126.4968	5126.4968	0.0000	8028.4
g06	100.0%	100.0%	-6961.8139	-6961.8138	-6961.8138	-6961.8138	0.0000	2151.1
g07	100.0%	100.0%	24.3063	24.3063	24.3063	24.3063	0.0000	16047.0
g08	100.0%	88.0%	-0.0958	-0.0958	-0.0258	-0.0877	0.0225	321.5
g09	100.0%	100.0%	680.6301	680.6301	680.6302	680.6301	0.0000	5677.4
g10	100.0%	100.0%	7049.2481	7049.2481	7049.2481	7049.2481	0.0000	30181.4
g11	100.0%	100.0%	0.7499	0.7500	0.7500	0.7499	0.0000	2389.4
g12	100.0%	76.0%	-1.0000	-0.9999	-0.9944	-0.9986	0.0024	774.8
g13	100.0%	60.0%	0.0540	0.0540	0.4388	0.2079	0.1924	95605.2
g14	100.0%	100.0%	-47.7648	-47.7648	-47.7648	-47.7648	0.0000	14287.4
g15	100.0%	100.0%	961.7151	961.7151	961.7151	961.7151	0.0000	11969.8
g16	100.0%	100.0%	-1.9051	-1.9051	-1.9051	-1.9051	0.0000	3885.2
g17	100.0%	56.0%	8853.5371	8853.5397	8927.5917	8886.1221	37.5169	124345.3
g18	100.0%	88.0%	-0.8660	-0.8659	-0.5000	-0.8360	0.0877	10214.6
g19	100.0%	100.0%	32.6557	32.6557	32.6557	32.6557	0.0000	39431.6
g21	100.0%	92.0%	193.7246	193.7246	324.7028	204.2029	36.2662	34987.1
g22	0.0%	0.0%	0.0000	0.0000	0.0000	0.0000	0.0000	-
g23	100.0%	96.0%	-400.0550	-400.0550	-100.0466	-388.0547	60.0017	39718.3
g24	100.0%	100.0%	-5.5080	-5.5080	-5.5079	-5.5079	0.0000	670.5

Table B.10: The CEC2006 results for the PIS method.

	Feasible Rate	Success Rate	Min	Median	Max	Mean	Std	Success Performance
g01	100.0%	80.0%	-15.0000	-14.9999	-12.4531	-14.7106	0.6418	70895.8
g02	100.0%	16.0%	-0.8036	-0.7704	-0.4307	-0.7336	0.1076	293550.0
g03	100.0%	100.0%	-1.0005	-1.0004	-1.0004	-1.0004	0.0000	20271.7
g04	100.0%	100.0%	-30665.5386	-30665.5386	-30665.5386	-30665.5386	0.0000	6694.3
g05	100.0%	100.0%	5126.4967	5126.4968	5126.4968	5126.4968	0.0000	13086.0
g06	100.0%	100.0%	-6961.8139	-6961.8138	-6961.8138	-6961.8138	0.0000	1743.2
g07	100.0%	100.0%	24.3063	24.3063	24.3063	24.3063	0.0000	22414.2
g08	100.0%	92.0%	-0.0958	-0.0958	-0.0258	-0.0903	0.0189	645.6
g09	100.0%	100.0%	680.6301	680.6301	680.6302	680.6301	0.0000	7080.9
g10	100.0%	100.0%	7049.2481	7049.2481	7049.2481	7049.2481	0.0000	128690.0
g11	100.0%	100.0%	0.7499	0.7500	0.7500	0.7500	0.0000	2031.8
g12	100.0%	84.0%	-1.0000	-0.9999	-0.9944	-0.9991	0.0021	625.2
g13	100.0%	48.0%	0.0540	0.4388	0.4388	0.2541	0.1962	146451.9
g14	100.0%	100.0%	-47.7648	-47.7648	-47.7648	-47.7648	0.0000	22727.6
g15	100.0%	100.0%	961.7151	961.7151	961.7151	961.7151	0.0000	9256.2
g16	100.0%	100.0%	-1.9051	-1.9051	-1.9051	-1.9051	0.0000	5077.7
g17	100.0%	40.0%	8853.5369	8927.5917	8927.5917	8897.9706	37.0265	193848.2
g18	100.0%	96.0%	-0.8660	-0.8659	-0.6750	-0.8583	0.0382	10699.7
g19	100.0%	100.0%	32.6557	32.6557	32.6557	32.6557	0.0000	66860.7
g21	100.0%	100.0%	193.7246	193.7246	193.7246	193.7246	0.0000	52343.8
g22	0.0%	0.0%	0.0000	0.0000	0.0000	0.0000	0.0000	-
g23	100.0%	96.0%	-400.0550	-400.0550	-100.0466	-388.0547	60.0017	43860.1
g24	100.0%	100.0%	-5.5080	-5.5079	-5.5079	-5.5079	0.0000	932.4

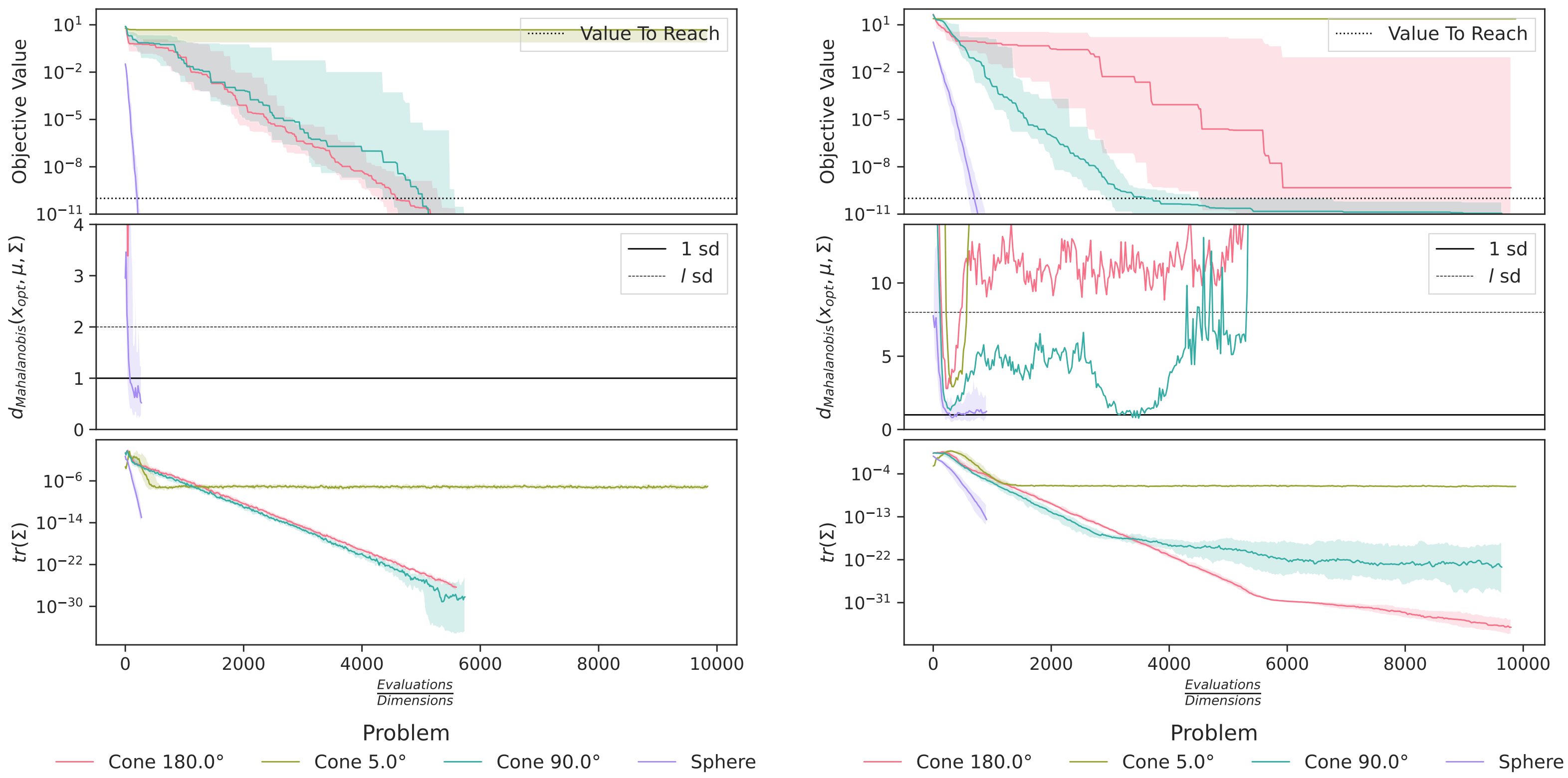
Table B.11: The CEC2006 results for the PI method.

	Feasible Rate	**Success Rate**	**Min**	**Median**	**Max**	**Mean**	**Std**	**Success Performance**
g01	100.0%	84.0%	-14.9999	-14.9999	-13.8281	-14.8124	0.4384	52165.9
g02	100.0%	24.0%	-0.8035	-0.7862	-0.4701	-0.7500	0.0927	425000.0
g03	100.0%	100.0%	-1.0004	-1.0004	-1.0004	-1.0004	0.0000	20217.0
g04	100.0%	100.0%	-30665.5386	-30665.5386	-30665.5386	-30665.5386	0.0000	6571.7
g05	100.0%	100.0%	5126.4968	5126.4968	5126.4968	5126.4968	0.0000	19371.8
g06	100.0%	100.0%	-6961.8139	-6961.8138	-6961.8138	-6961.8138	0.0000	1793.6
g07	100.0%	100.0%	24.3063	24.3063	24.3063	24.3063	0.0000	18706.2
g08	100.0%	96.0%	-0.0958	-0.0958	-0.0956	-0.0958	0.0000	792.8
g09	100.0%	100.0%	680.6301	680.6301	680.6302	680.6301	0.0000	6544.2
g10	100.0%	100.0%	7049.2481	7049.2481	7049.2481	7049.2481	0.0000	37638.4
g11	100.0%	100.0%	0.7499	0.7500	0.7500	0.7500	0.0000	1859.5
g12	100.0%	84.0%	-1.0000	-0.9999	-0.9944	-0.9991	0.0021	731.8
g13	100.0%	40.0%	0.0540	0.4388	0.4388	0.2849	0.1924	146339.8
g14	100.0%	100.0%	-47.7648	-47.7648	-47.7648	-47.7648	0.0000	24463.9
g15	100.0%	100.0%	961.7150	961.7151	961.7151	961.7151	0.0000	10954.9
g16	100.0%	100.0%	-1.9051	-1.9051	-1.9051	-1.9051	0.0000	4968.8
g17	100.0%	60.0%	8853.5361	8853.5390	8927.5917	8883.1597	37.0267	167183.0
g18	100.0%	84.0%	-0.8660	-0.8659	-0.6750	-0.8354	0.0714	13886.2
g19	100.0%	100.0%	32.6557	32.6557	32.6557	32.6557	0.0000	61539.2
g21	100.0%	96.0%	193.7246	193.7246	324.7028	198.9637	26.1957	37228.0
g22	0.0%	0.0%	0.0000	0.0000	0.0000	0.0000	0.0000	-
g23	100.0%	88.0%	-400.0550	-400.0550	-100.0466	-364.0540	99.5015	68676.5
g24	100.0%	100.0%	-5.5080	-5.5079	-5.5079	-5.5080	0.0000	851.9

Table B.12: The CEC2006 results for the DP-PI method.

Appendix C

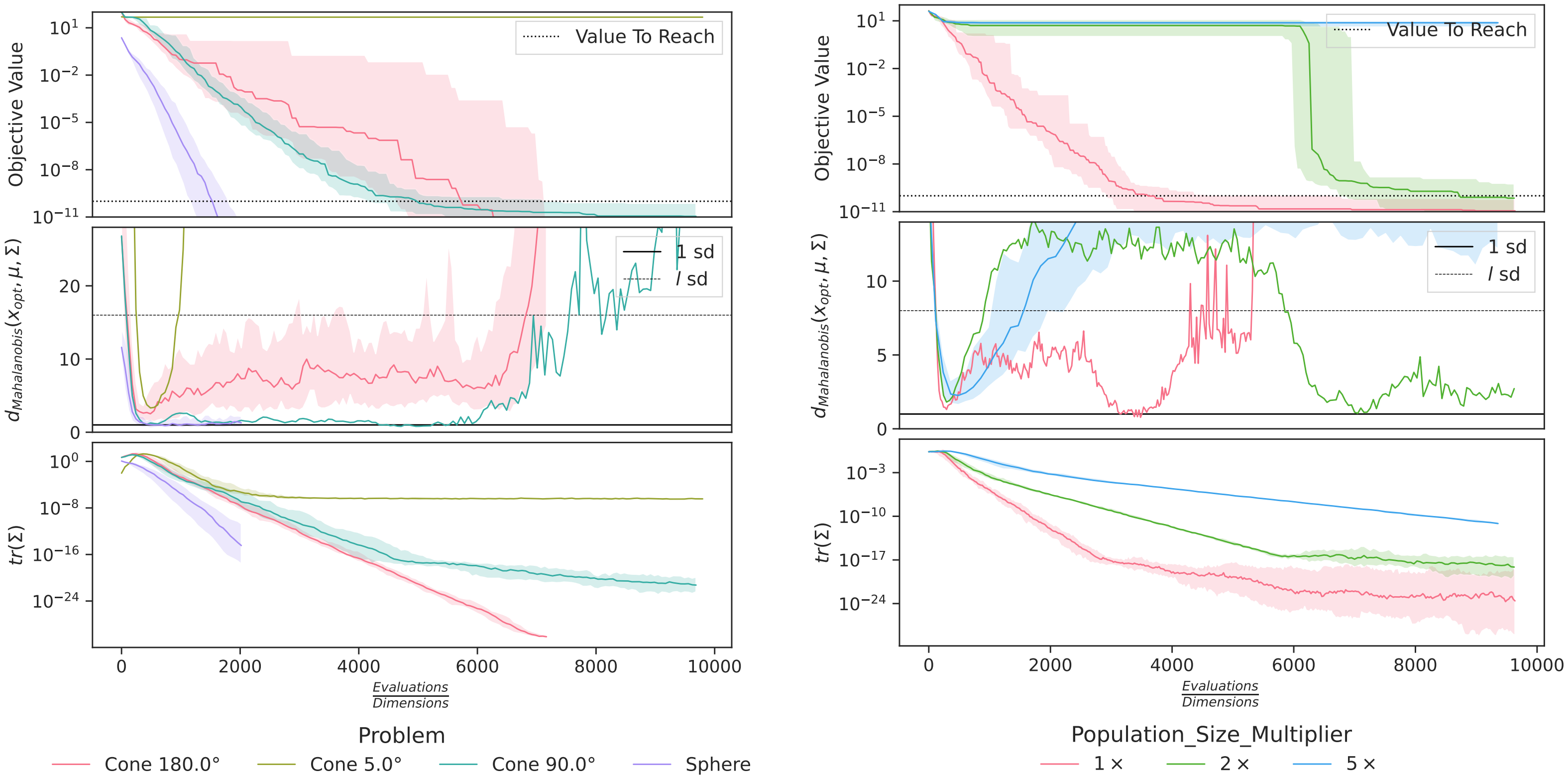
Effectiveness Results



(a) All 2D problems using AL

(b) All 8D problems using AL

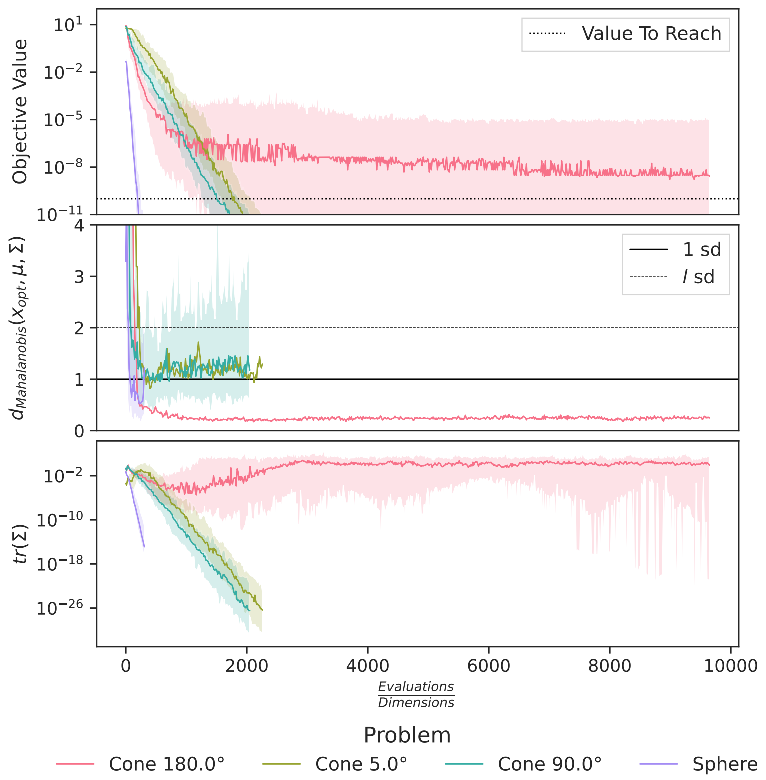
Figure C.1: The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs.



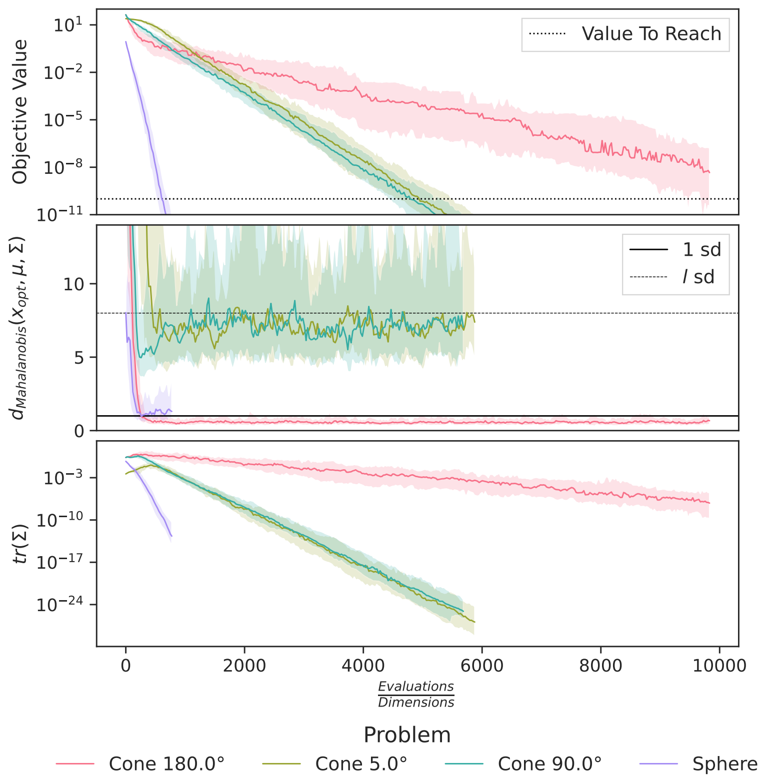
(a) All 16D problems using AL

(b) 8D Cone 90° problem using AL

Figure C.2: The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs.

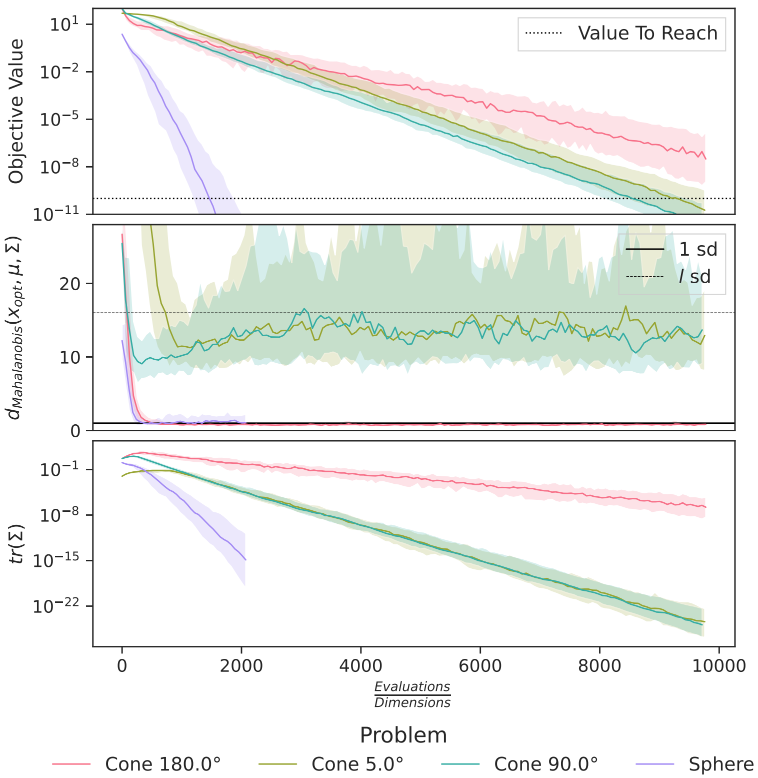


(a) All 2D problems using DP-O

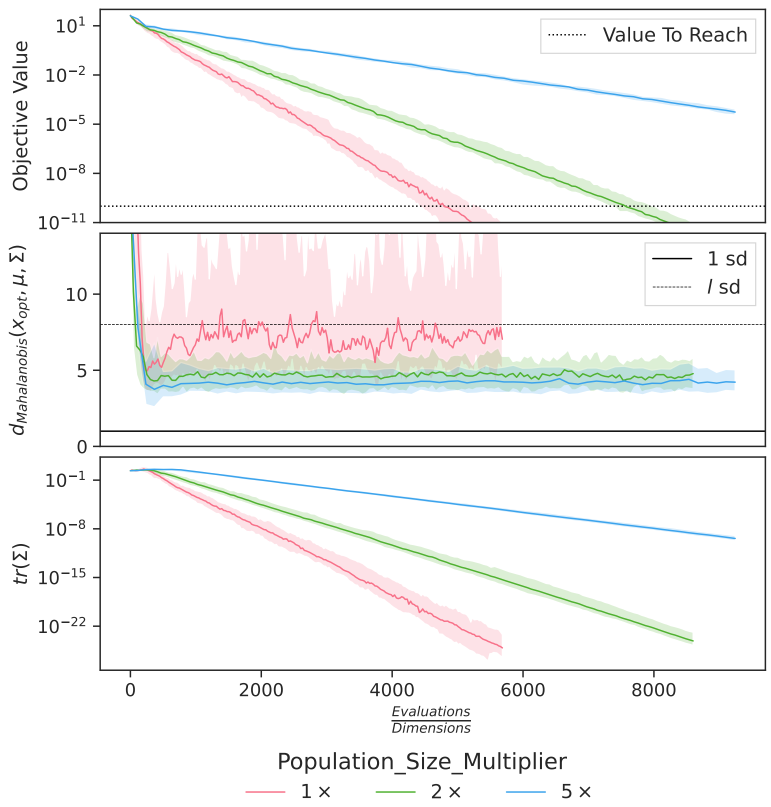


(b) All 8D problems using DP-O

Figure C.3: The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs.

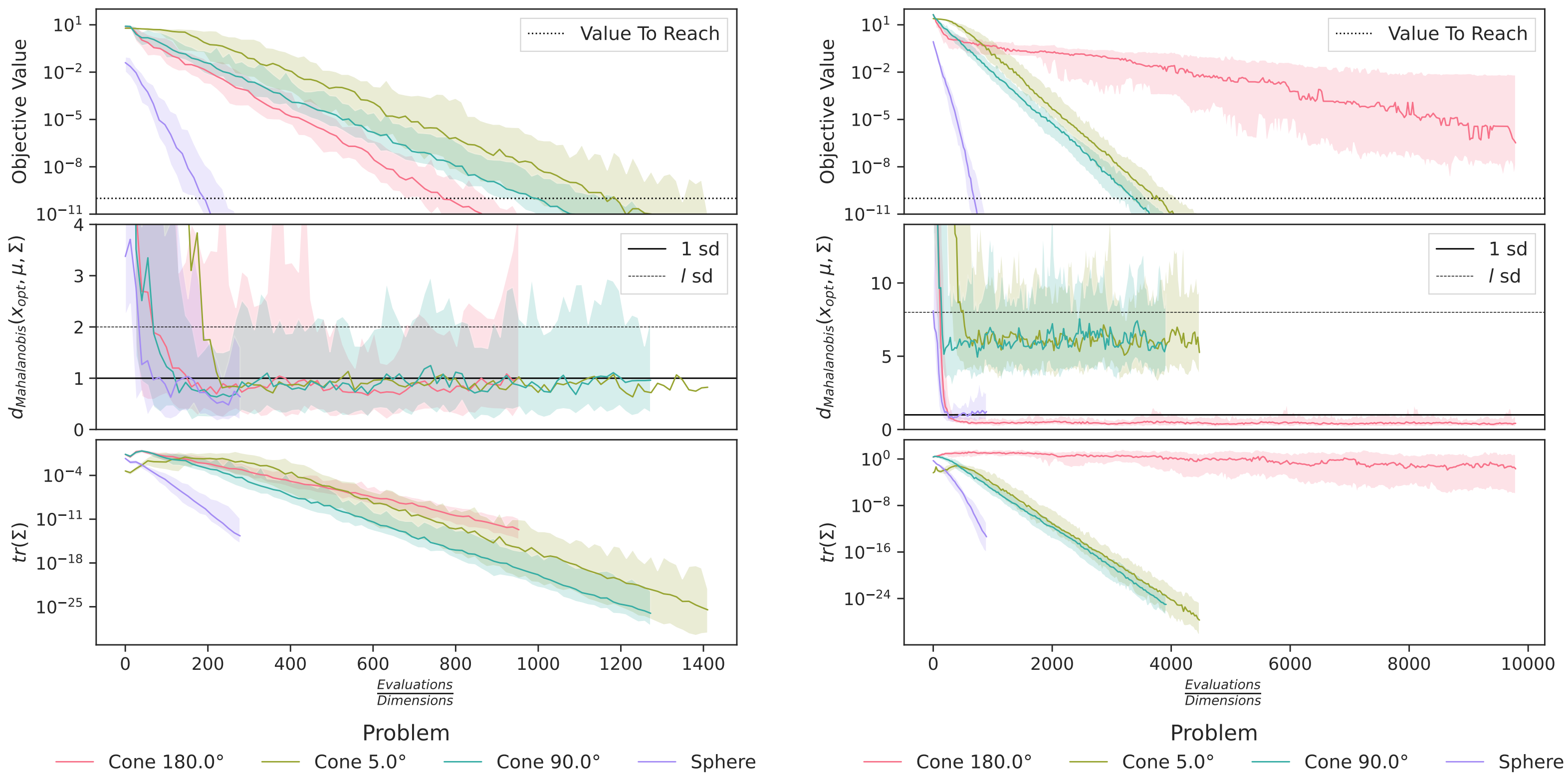


(a) All 16D problems using DP-O



(b) 8D Cone 90° problem using DP-O

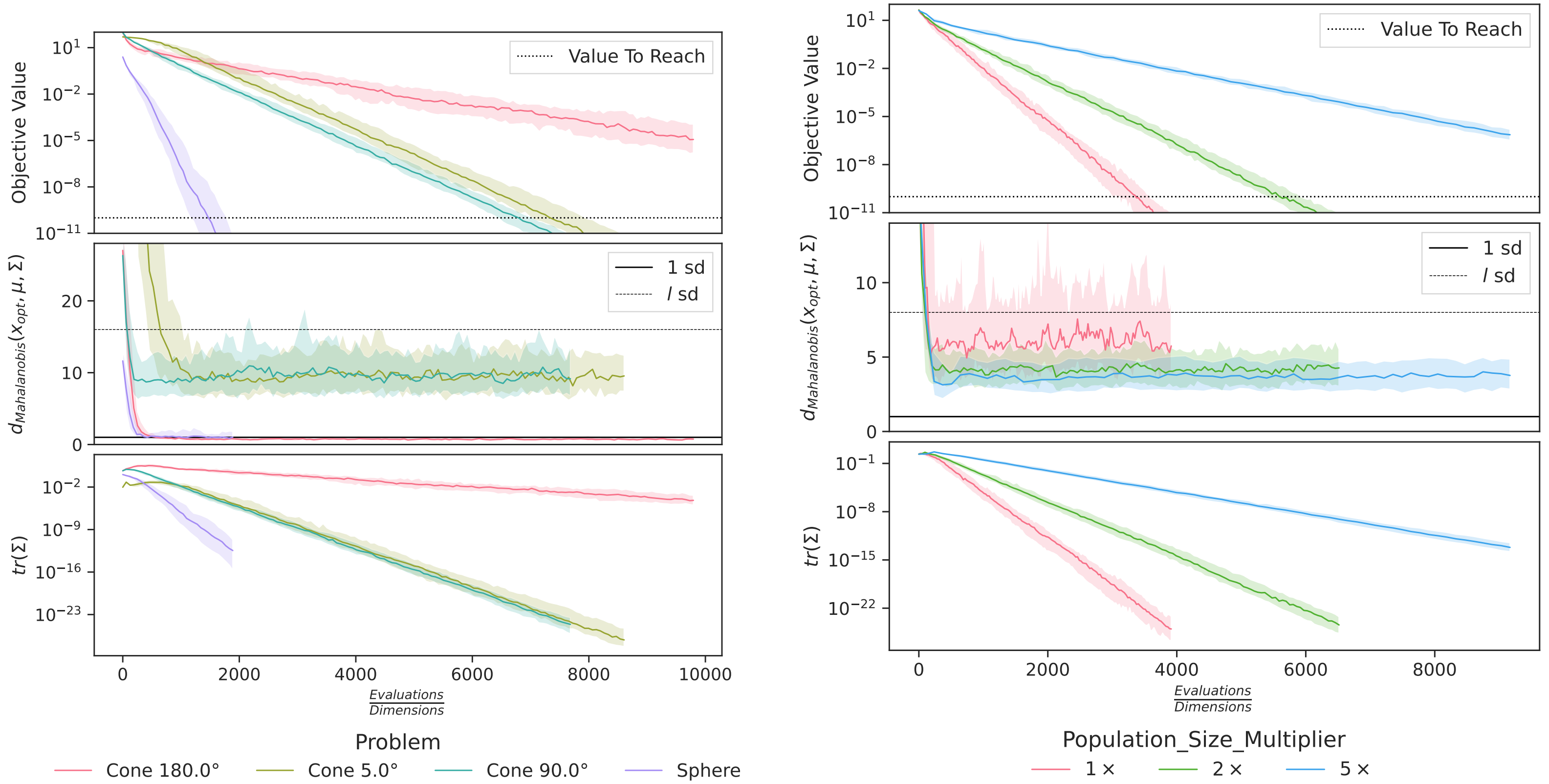
Figure C.4: The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs.



(a) All 2D problems using DP-PI

(b) All 8D problems using DP-PI

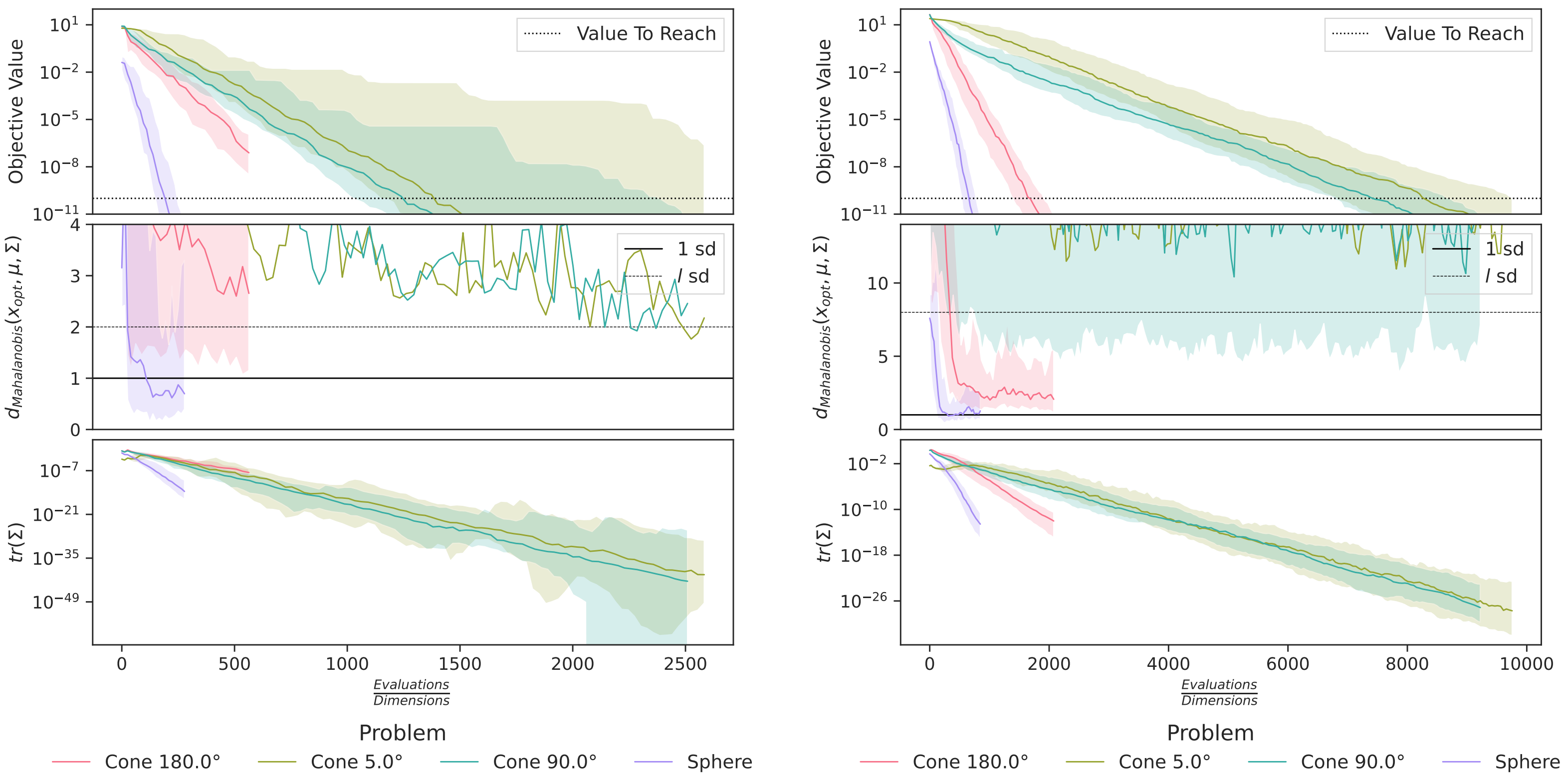
Figure C.5: The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs.



(a) All 16D problems using DP-PI

(b) 8D Cone 90° problem using DP-PI

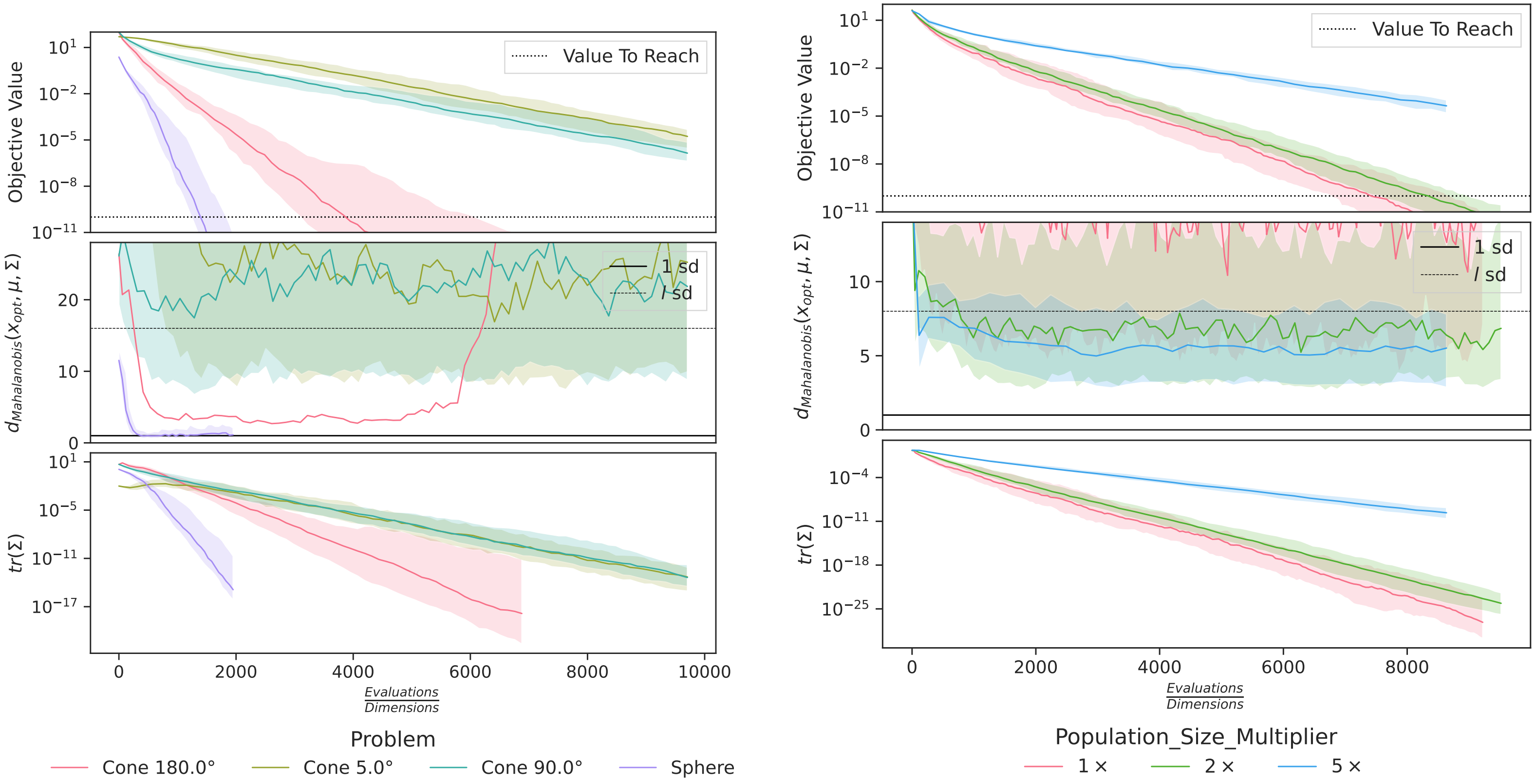
Figure C.6: The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs.



(a) All 2D problems using DP-T

(b) All 8D problems using DP-T

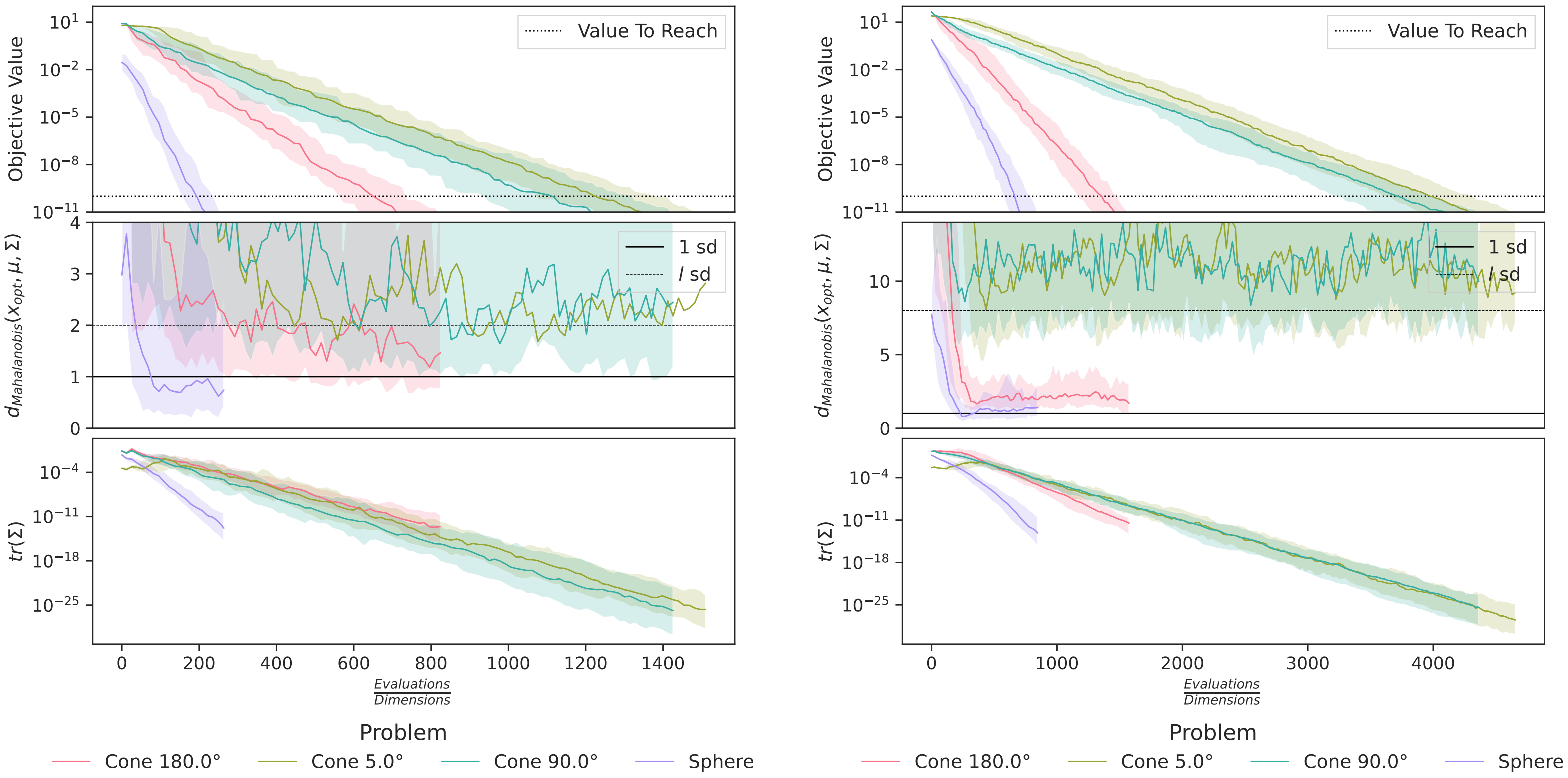
Figure C.7: The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs.



(a) All 16D problems using DP-T

(b) 8D Cone 90° problem using DP-T

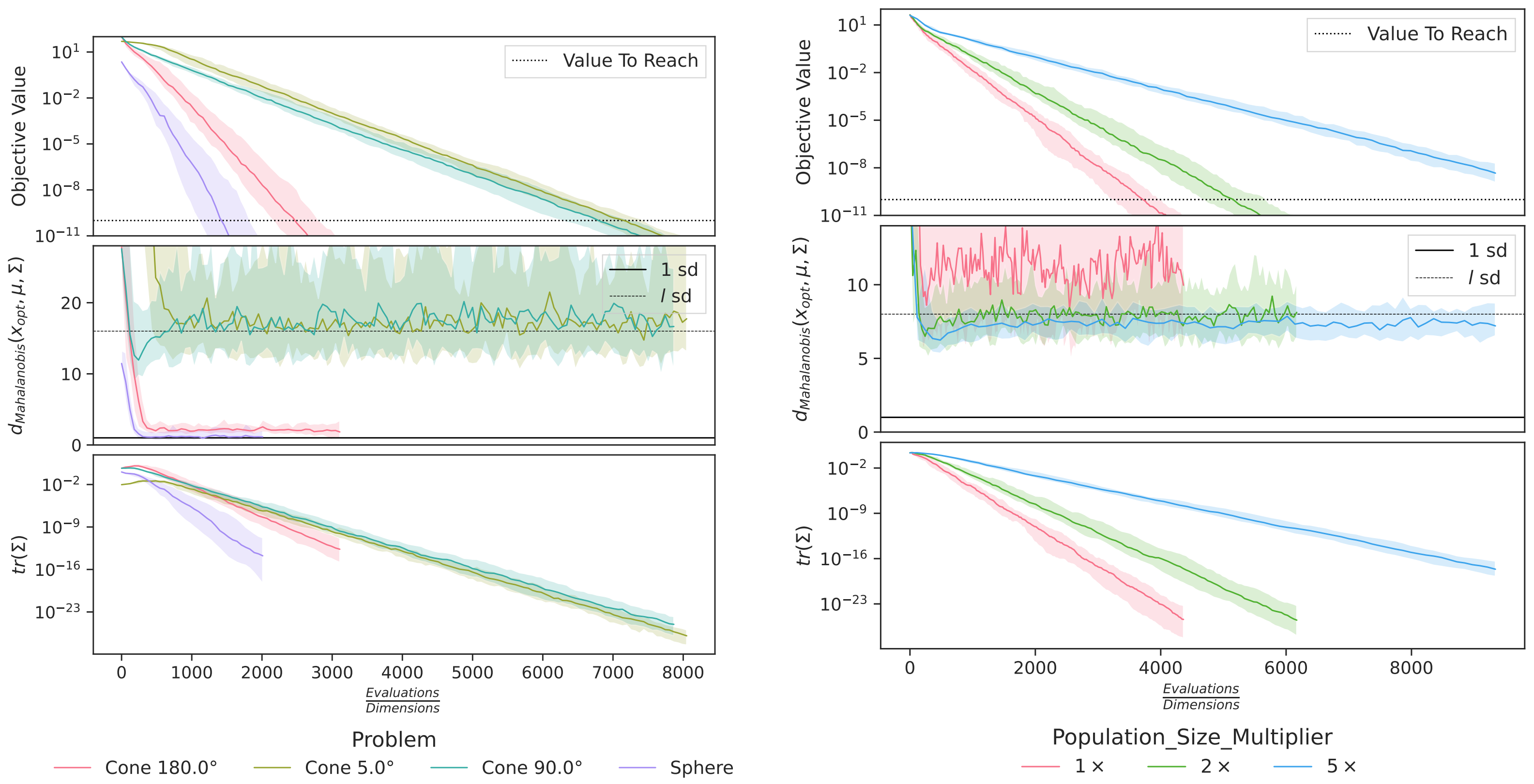
Figure C.8: The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs.



(a) All 2D problems using EPS

(b) All 8D problems using EPS

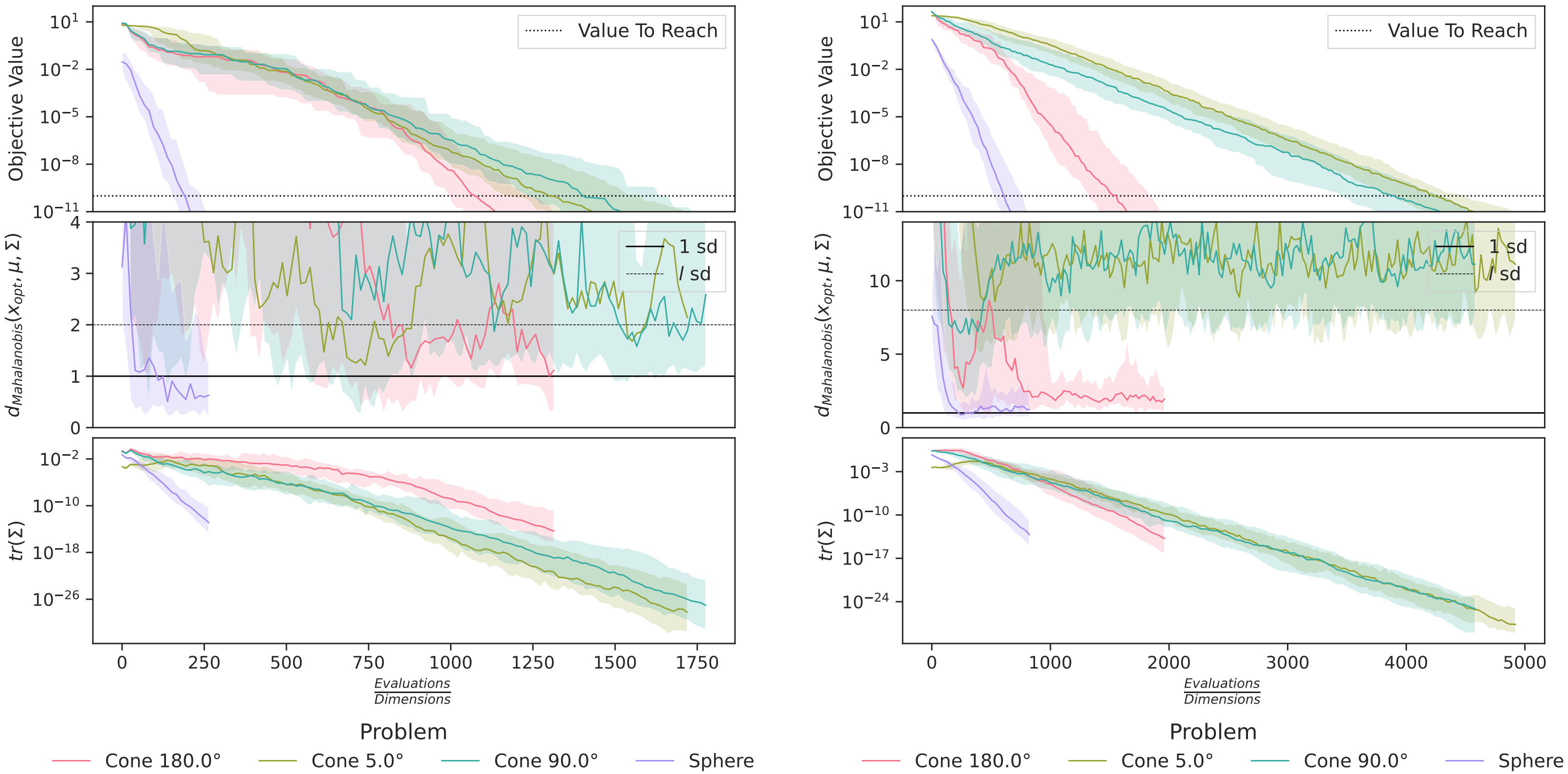
Figure C.9: The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs.



(a) All 16D problems using EPS

(b) 8D Cone 90° problem using EPS

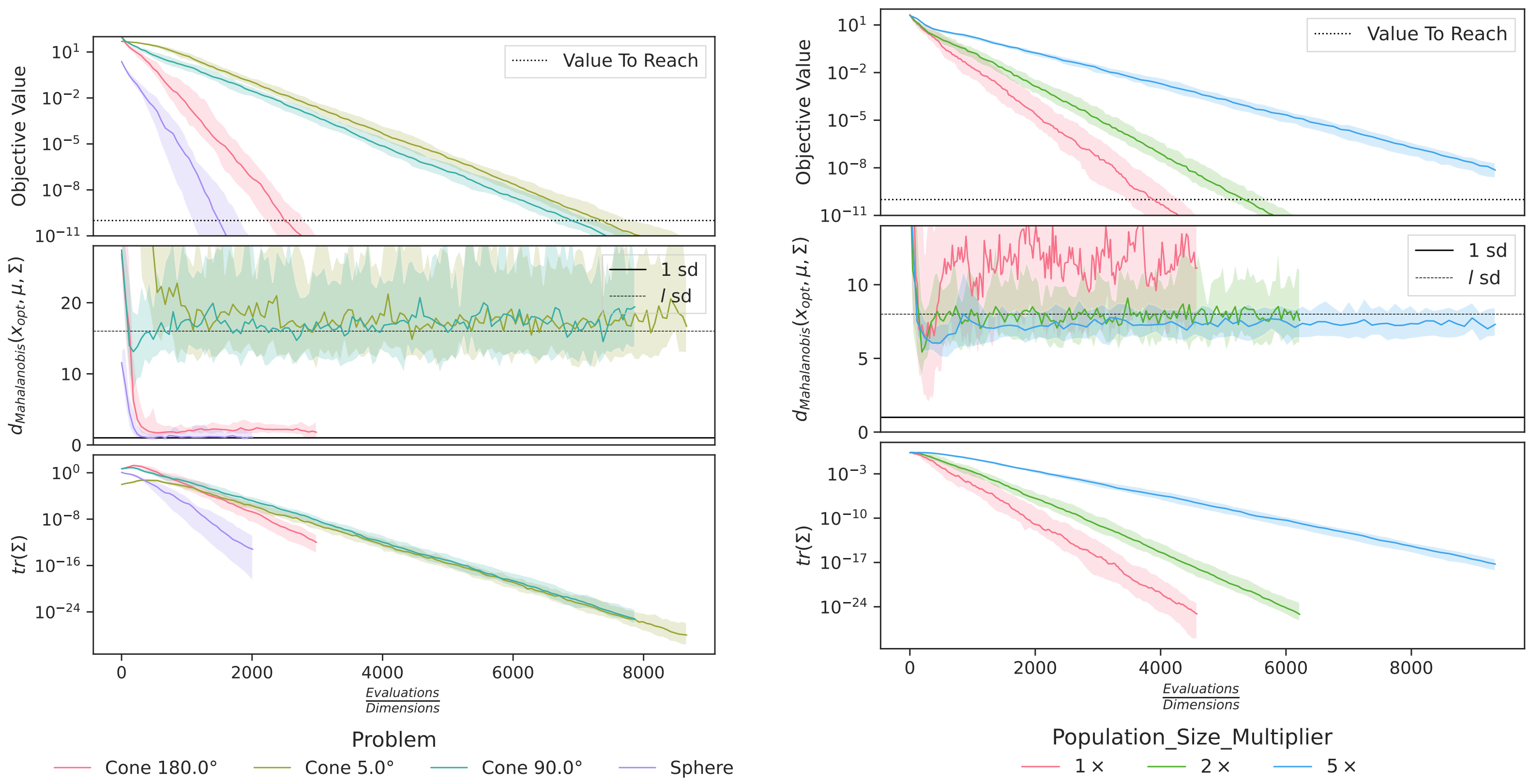
Figure C.10: The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs.



(a) All 2D problems using IEPS

(b) All 8D problems using IEPS

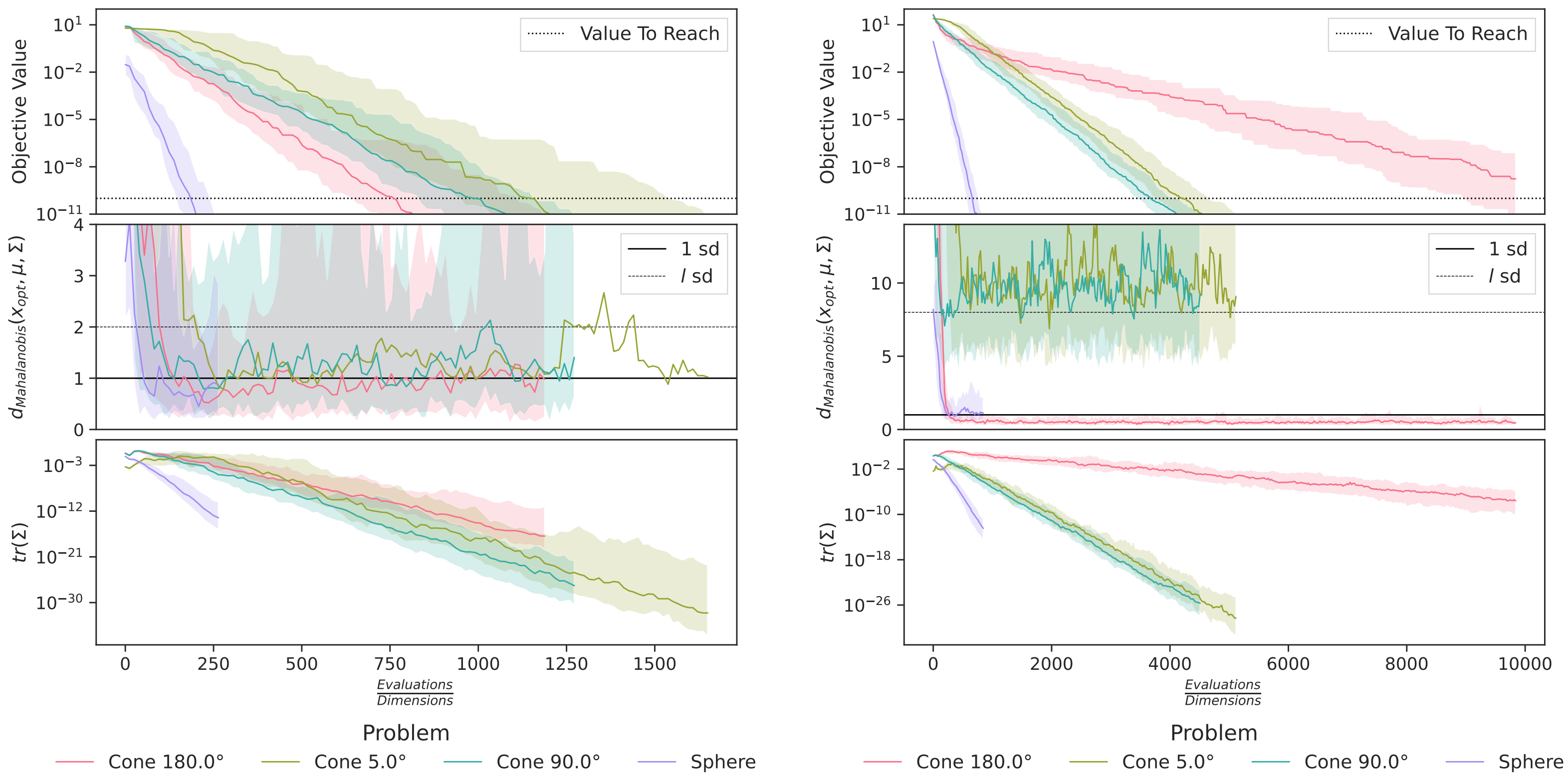
Figure C.11: The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs.



(a) All 16D problems using IEPS

(b) 8D Cone 90° problem using IEPS

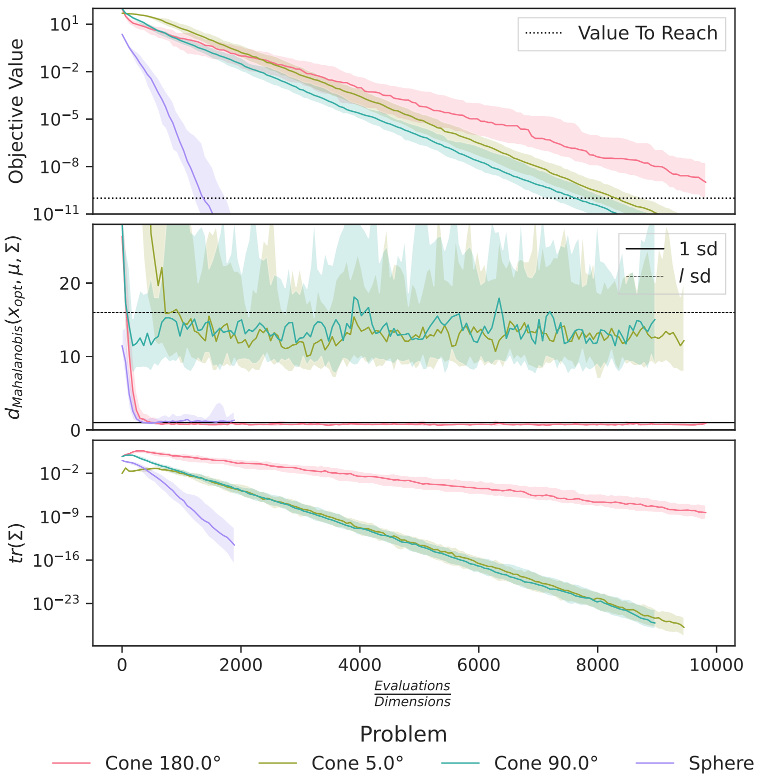
Figure C.12: The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs.



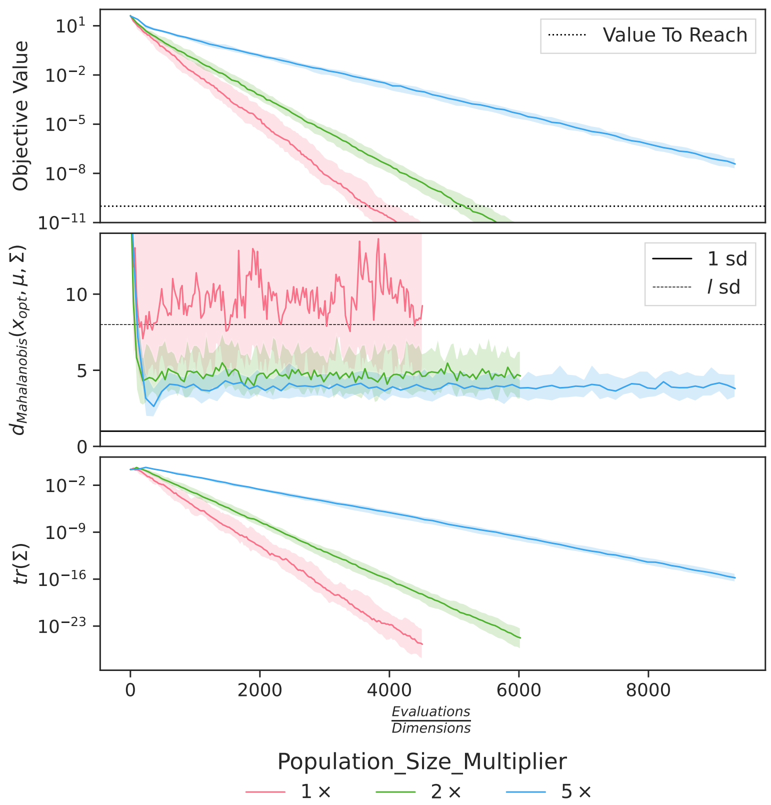
(a) All 2D problems using PI

(b) All 8D problems using PI

Figure C.13: The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs.

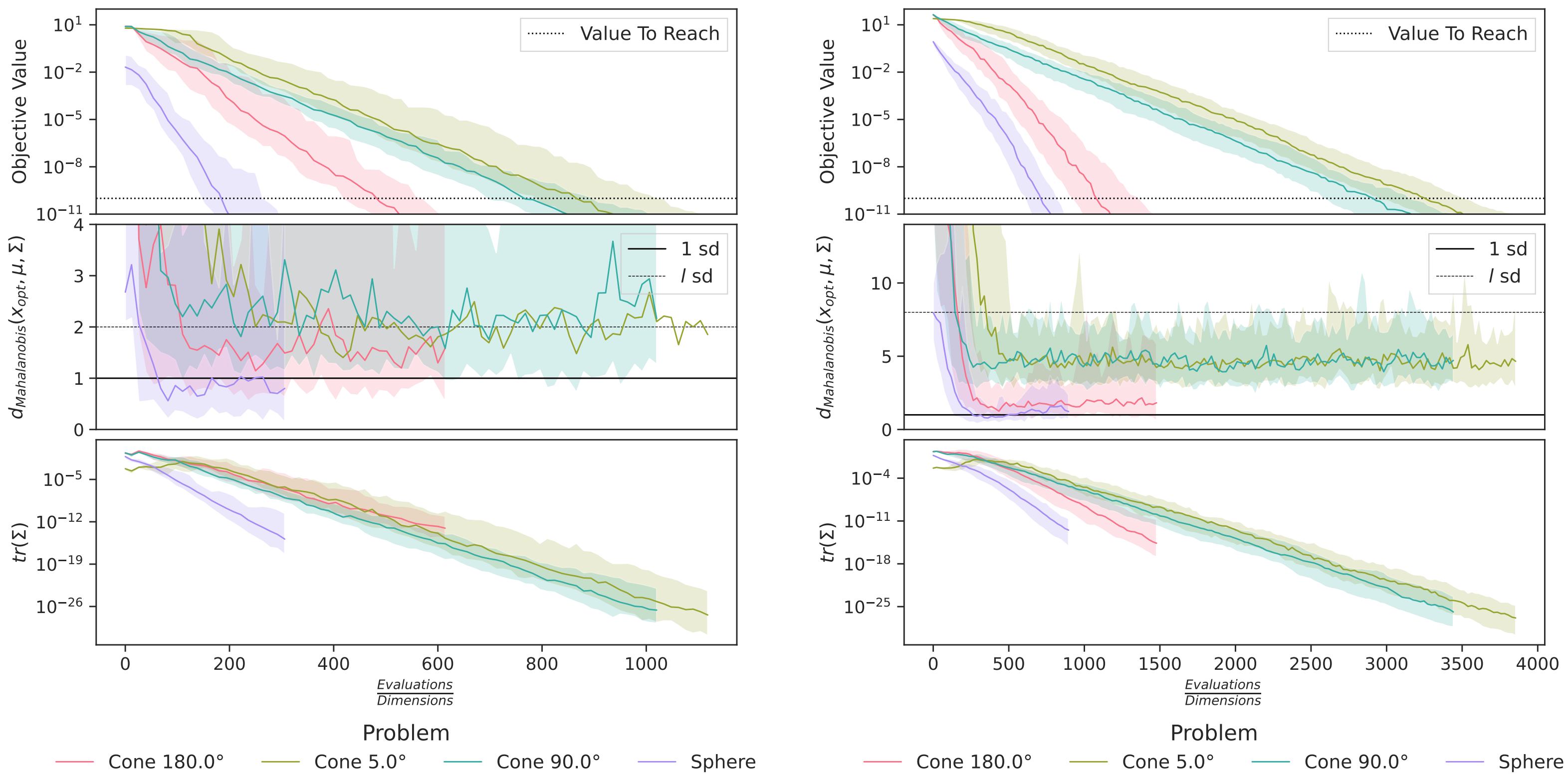


(a) All 16D problems using PI



(b) 8D Cone 90° problem using PI

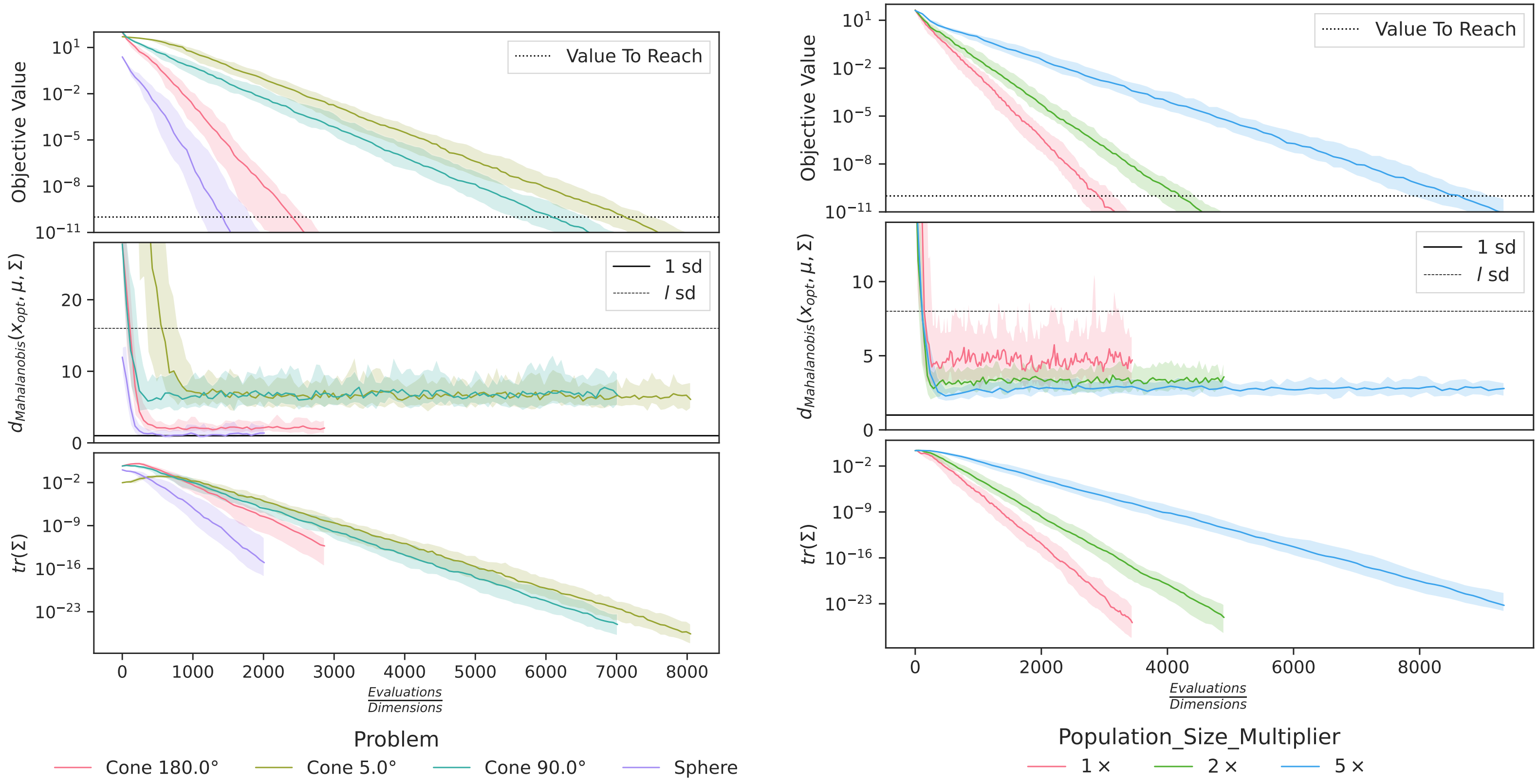
Figure C.14: The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs.



(a) All 2D problems using PIS

(b) All 8D problems using PIS

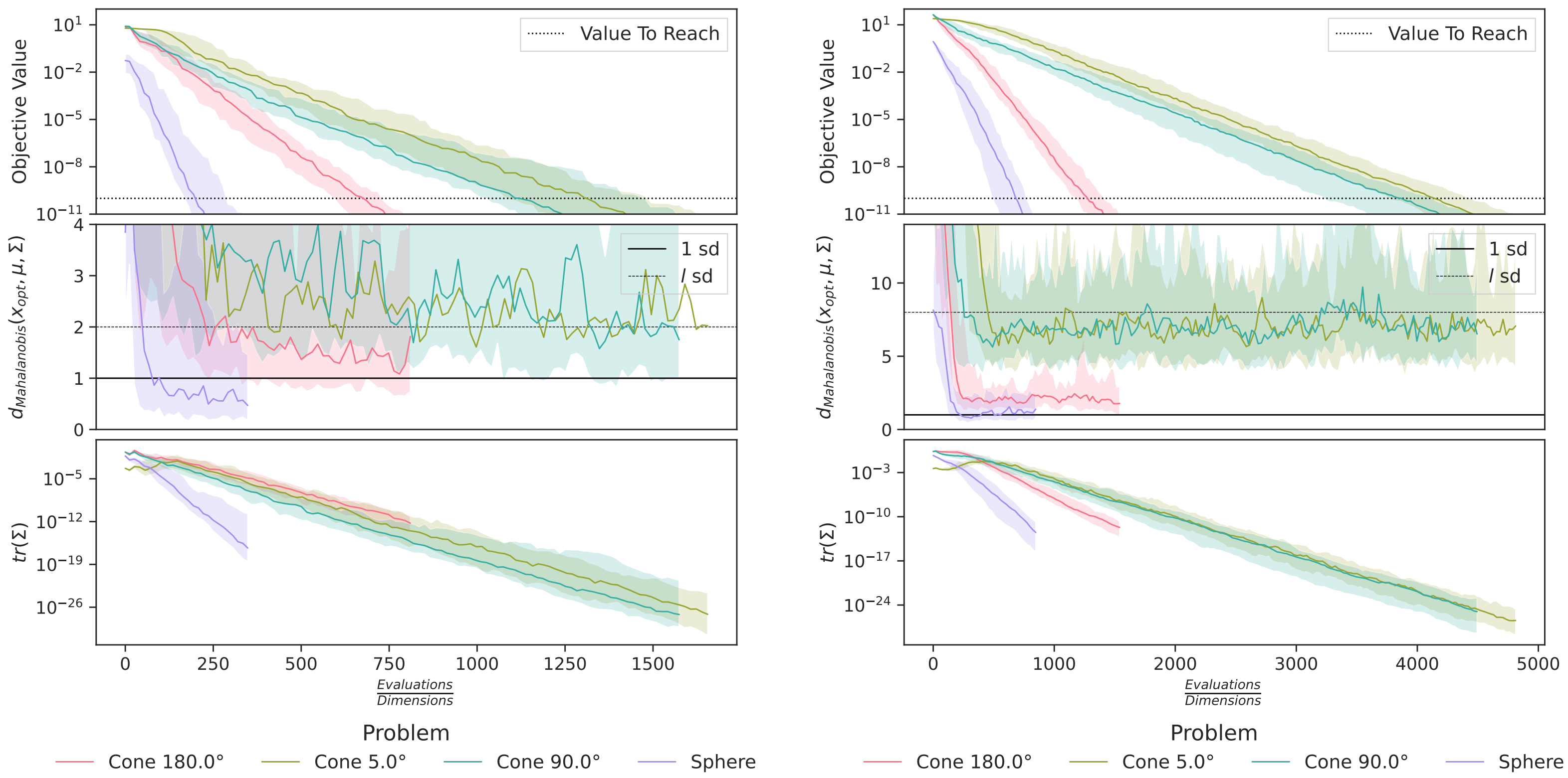
Figure C.15: The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs.



(a) All 16D problems using PIS

(b) 8D Cone 90° problem using PIS

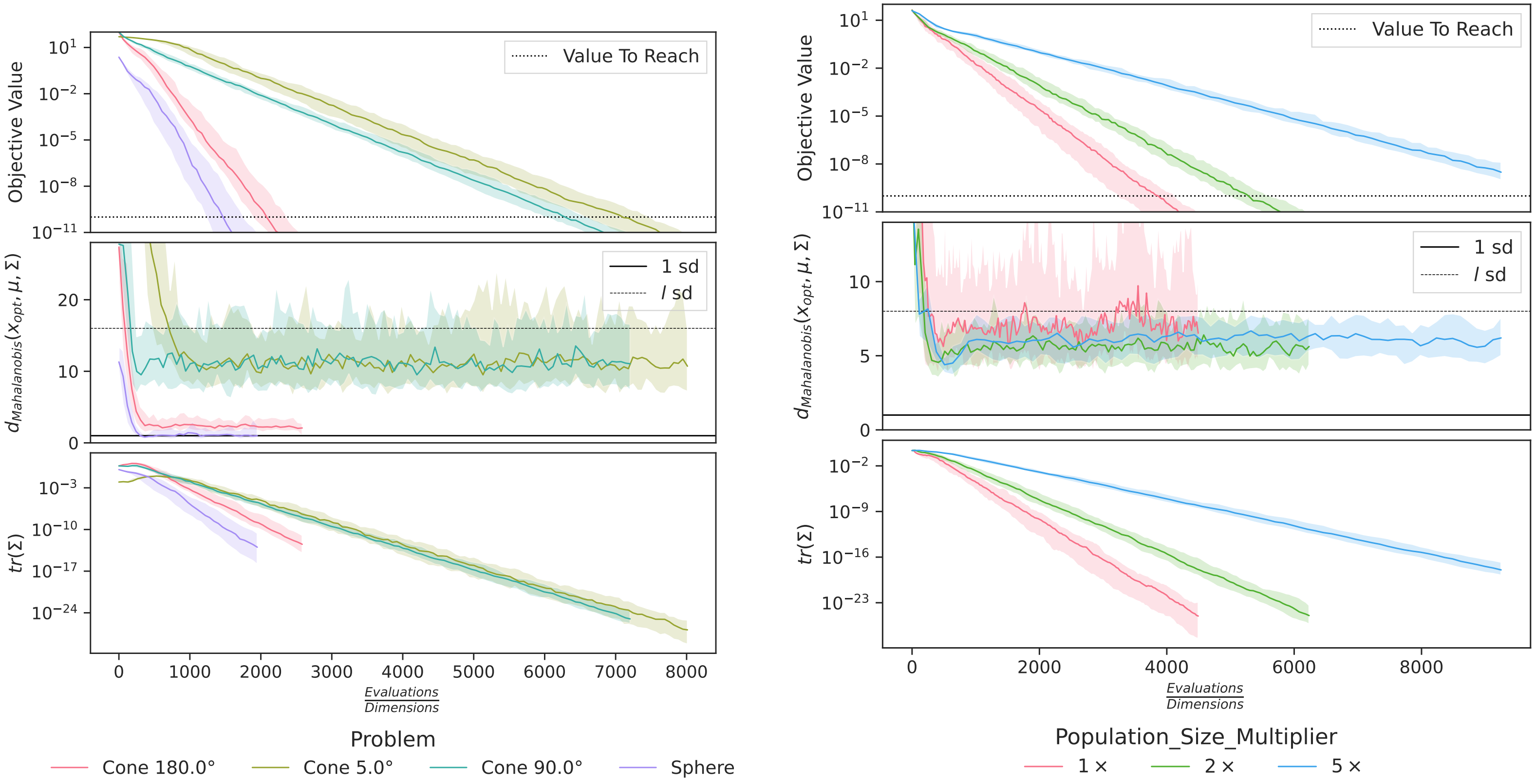
Figure C.16: The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs.



(a) All 2D problems using R-BS

(b) All 8D problems using R-BS

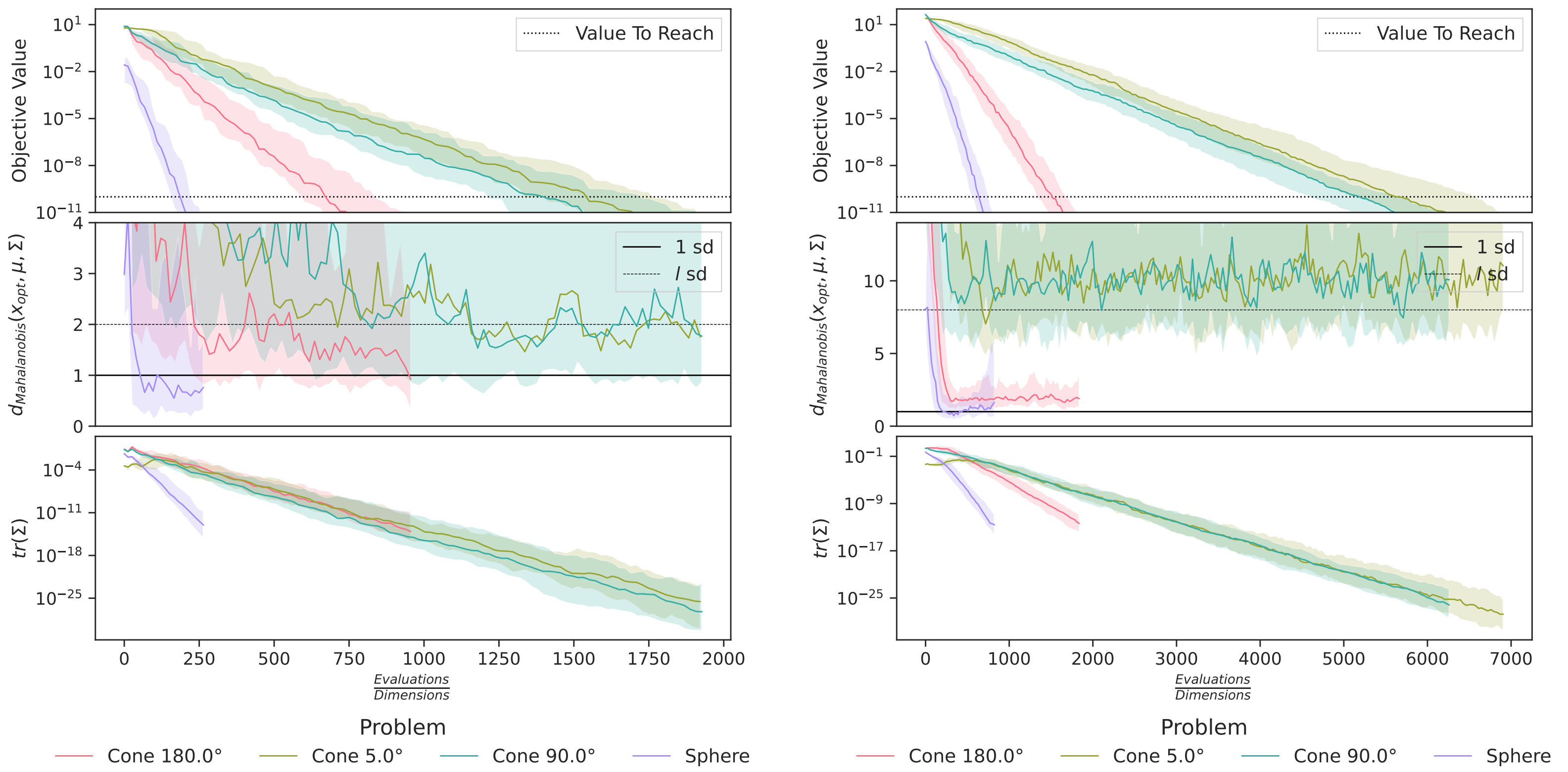
Figure C.17: The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs.



(a) All 16D problems using R-BS

(b) 8D Cone 90° problem using R-BS

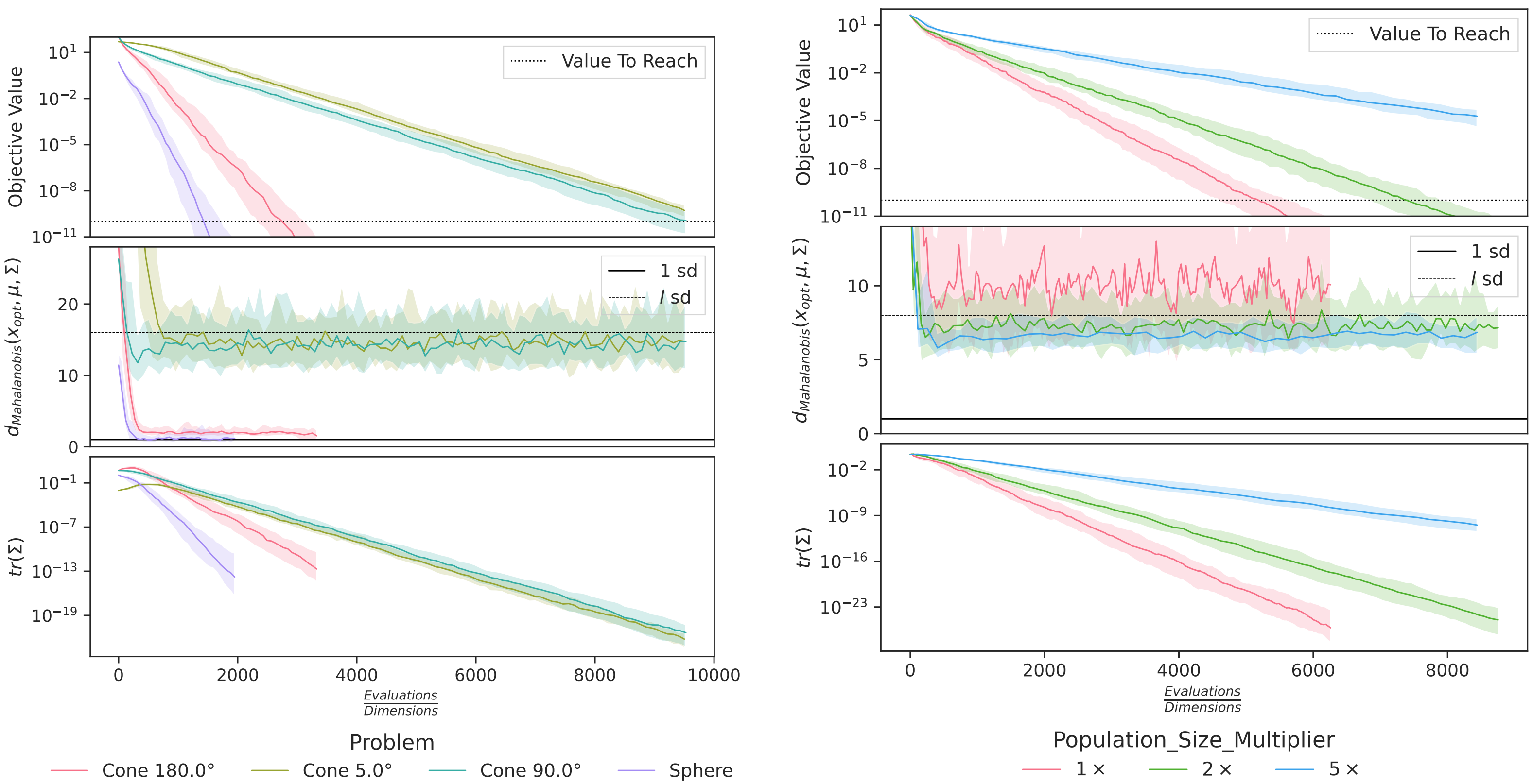
Figure C.18: The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs.



(a) All 2D problems using R-R

(b) All 8D problems using R-R

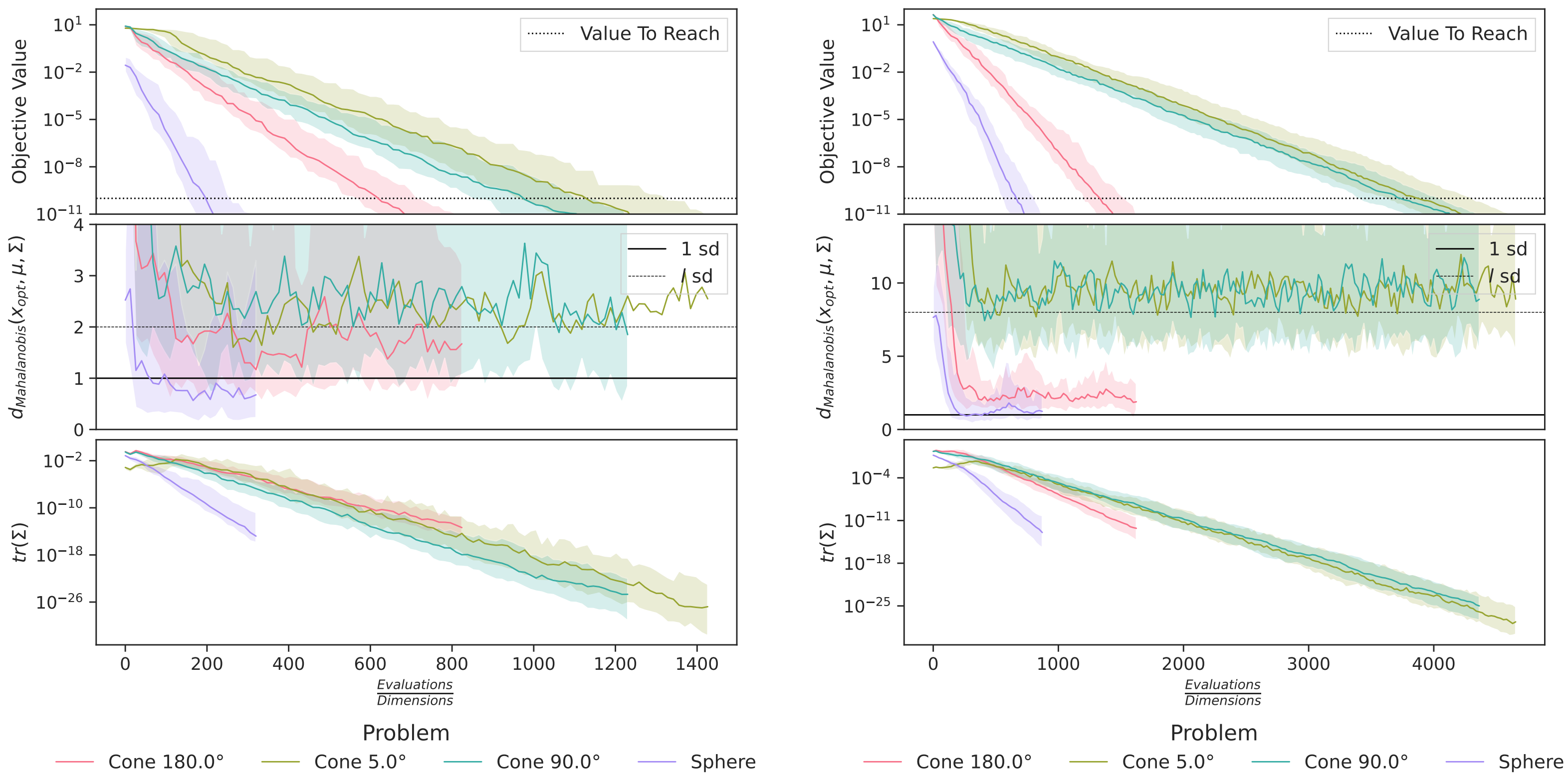
Figure C.19: The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs.



(a) All 16D problems using R-R

(b) 8D Cone 90° problem using R-R

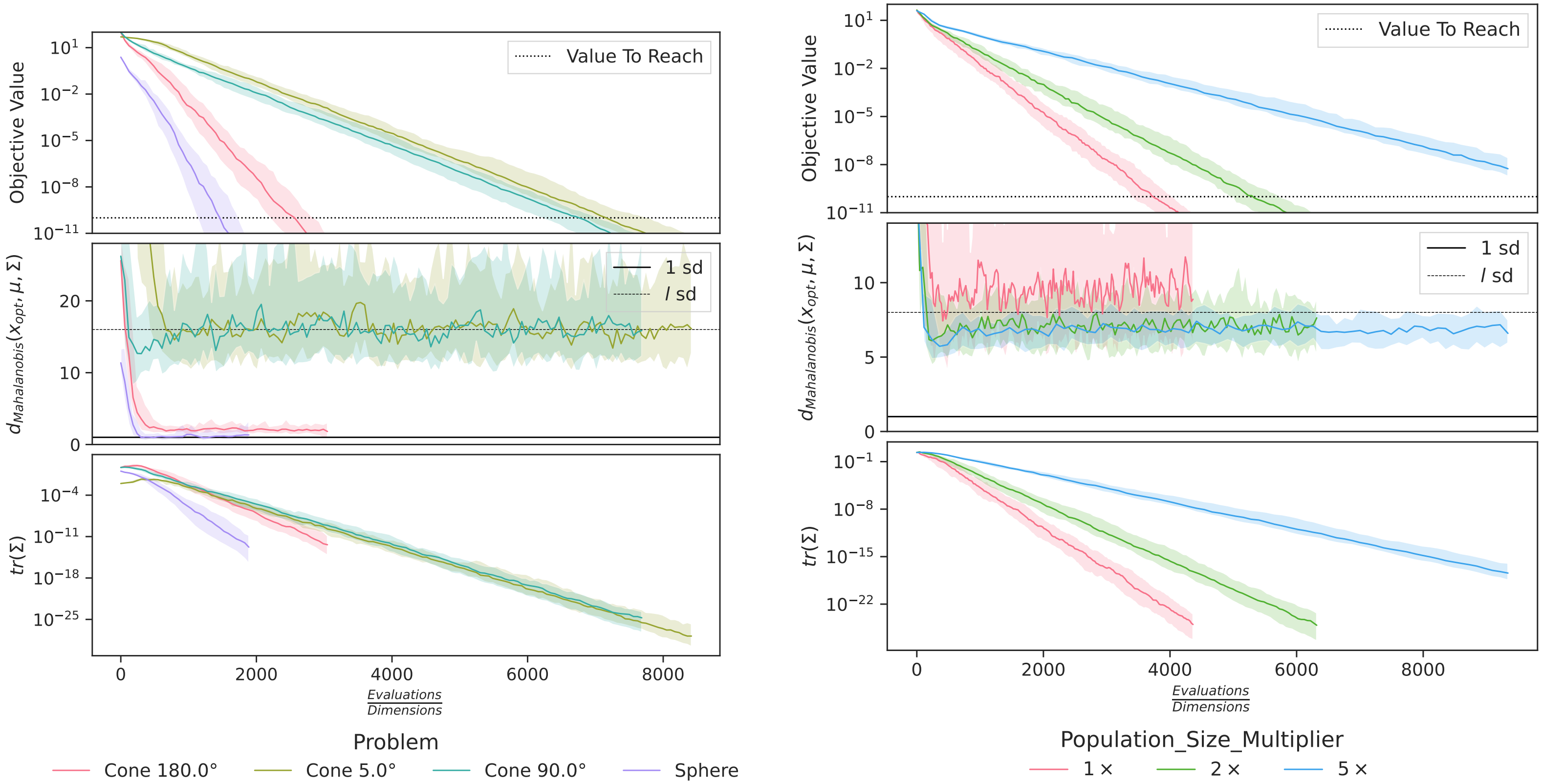
Figure C.20: The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs.



(a) All 2D problems using SR

(b) All 8D problems using SR

Figure C.21: The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs.



(a) All 16D problems using SR

(b) 8D Cone 90° problem using SR

Figure C.22: The median and interdecile range of the objective value, the Mahalanobis distance between the constrained optimum and the estimated distribution, as well as the trace of the covariance matrix per evaluation over 31 runs.