# An Instance Space Analysis of Constrained Multi-Objective Optimization Problems

Hanan Alsouly, Michael Kirley, and Mario Andrés Muñoz,

***Abstract*—Constrained multi-objective optimization problems (CMOPs) are generally more challenging than unconstrained problems. This in part can be attributed to the infeasible region generated by the constraint functions, the interaction between constraints and objectives, or both. In this paper, we explore the relationship between the performance of constrained multi-objective evolutionary algorithms (CMOEAs) and the instance characteristics of CMOP using Instance Space Analysis (ISA). To do this, we extend recent work on Landscape Analysis features for characterising CMOPs. Specifically, we introduce new features to describe the multi-objective-violation landscape, formed by the interaction between constraint violation and multi-objective fitness. Detailed evaluation of the algorithm footprints, spanning eight CMOP benchmark suites and fifteen CMOEAs, demonstrates that ISA effectively captures the strength and weakness of the CMOEAs. We conclude that two characteristics, the isolation of non-dominate set and the correlation between constraints and objectives evolvability, have the greatest impact on algorithm performance. However, the current benchmarks problems lack of diversity to represent the real-world problems and to fully reveal the efficacy of CMOEAs evaluated.**

***Index Terms*—constrained multiobjective optimization, problem characterization, landscape analysis, algorithm selection, evolutionary algorithm.**

## I. Introduction

CONSTRAINED multi-objective optimization problems (CMOPs) involve searching for the best trade-off between multiple conflicting objectives subject to one or more constraints. Many real-world optimization problems match this description, in areas as diverse as mechanical design, chemical engineering, and power system optimization [1]. Generally, a CMOP is more challenging than its unconstrained counterpart due to the addition of one or more constraint functions, and the resulting interactions between the constraints and the objectives [2]. Constraints may change the shape and location of the Pareto front $(PF)$, often creating a small and possibly disjoint feasible region, resulting in additional difficulties when attempting to estimate the $PF$. Consequently, several constrained multi-objective evolutionary algorithms (CMOEAs) have been introduced to specifically tackle CMOPs [3]. However, as per many other problem domains, practice has shown that no single algorithm outperforms all other algorithms across all problem instances [1], [4]. Each algorithm has its strengths and weaknesses, and it is difficult to choose the best one for solving a particular instance. Therefore, it is necessary to understand when an algorithm is suitable or not, i.e., when it performs well and when it fails, which requires an understanding of the characteristics of the instances being solved, e.g. multi-modality and variable scaling, and what distinguish the instances from each other [5].

In this paper, we explore the relationship between CMOEA performance and the instances characteristics of CMOP using Instance Space Analysis (ISA). Proposed by Smith-Miles *et al.*, [6], ISA is a methodology for assessing the difficulty of a set of problem instances for a group of algorithms. Figure 1 illustrates ISA's framework, which uses a meta-data set consisting of features that characterize a set of instances, and performance measures of a group of algorithms on those instances. Then, by selecting a subset of uncorrelated features that are predictive of algorithm performance, and using a tailored dimensionality reduction method, the meta-data is projected into a 2-dimensional plane called the *instance space*. Within this, each instance is represented as a point, allowing for the visualization of the similarities and differences between instances, in terms of characteristics and algorithm performance. An examination of the generated instance space can then be used to identify regions of good performance, called *footprints*, where an algorithm is expected to perform well and why.

ISA has been employed successfully on related problem domains. For example, Yap *et al.*, [7] performed an ISA of combinatorial multi-objective optimization problems (MOPs), discovering that MOEA/D is preferred, not only when the number of objectives increased, but also when the degree of conflict between objectives decreased. Similarly, Muñoz and Smith-Miles [8] analyzed the space of continuous single-objective optimization problems, identifying that multi-modal instances with adequate global structure are hard to solve by most studied algorithms with the exception of BIPOP-CMA-ES. In both works, Landscape Analysis *features* [9] were employed to characterize a problem instance. Therefore, a necessary first step when applying ISA for CMOP will be to identify and calculate appropriate, informative Landscape Analysis features.

Recent works have proposed Landscape Analysis features for characterizing MOPs. Kerschke *et al.*, [11] studied the notion of multimodality in MOPs and provided a set of features to quantify it, whilst Liefooghe *et al.*, [12] extended previous

Funding was provided by the Australian Research Council through grants FL140100012 and IC200100009.

H. Alsouly, M. Kirley and M.A. Muñoz are with the School of Computing and Information, University of Melbourne, Melbourne, Australia, and with the ARC Centre in Optimisation Technologies, Integrated Methodologies and Applications (OPTIMA), Australia. e-mail: (halsouly@student.unimelb.edu.au and {mkirley,munoz.m}@unimelb.edu.au).

H. Alsouly is also with College of Computing and Information Sciences, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh, Saudi Arabia

Manuscript received MMM D, 2022; revised MMM D, 2022.

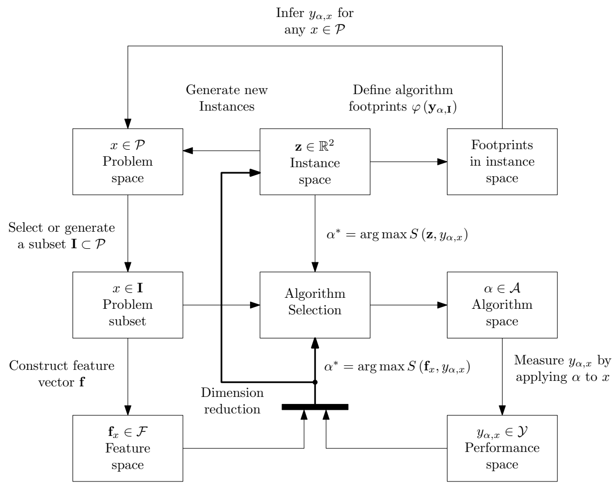



Fig. 1. Summary of the Instance Space Analysis framework [10].

works in the combinatorial MOPs domain to characterize continuous MOPs, focusing on multimodality, evolvability, and ruggedness. Unfortunately, it is not a straight forward task to identify Landscape Analysis features for CMOPs. However, the features described above can be used in the CMOPs domain to help characterize the objective space. Still, features associated with the constraints' violation, and features representing the interaction between objectives and constraints are required.

There have been limited attempts to characterize constrained optimization problems. For example, for single-objective problems (COPs), Malan *et al.*, [13] defined the concept of a violation landscape, proposing four features to characterize the feasible and constrained spaces. In other work, Poursoltan and Neumann [14] introduced a biased sampling technique to quantify the ruggedness of a COP. Picard and Schiffmann [15] focussing on CMOPs, adopted two features from [13], and extended another so that it could be used to measure 'disjointedness' of the feasible region. They also proposed two features to quantify the relationship between the objectives and constraints. Vodopija *et al.*, [16] were the first to introduce violation multimodality in CMOPs, proposing a set of features to characterize violation multimodality, smoothness, and the correlation between the objectives and constraints. They then used those features to compare the characteristics of eight benchmark problem suites against a real-world suite. This work did provide important insights. However, the approach was limited to the violation landscape and did not capture important aspects that need to be quantified, such as the relationship between the constrained and unconstrained PF, or the ruggedness and evolvability of the multi-objective-violation landscape.

To construct the first ISA for continuous CMOPs, we introduce the multi-objective-violation landscape, formed by the interaction between constraint violation and multi-objective fitness. This requires the introduction of 12 new features and modification of 22 existing features to quantify the characteristics of the violation landscape and multi-objective-violation landscape. The meta-data set is then generated for the instance space by processing the features for eight CMOP benchmark suites and the performance of 15 CMOEAs.

Comprehensive analysis of the generated footprints illustrates that ISA effectively captures the strength and weakness of the CMOEAs. A key observation is that there are two characteristics in particular that affect the performance of most CMOEAs – the isolation of the non-dominate set and the correlation between constraints and objectives. However, the performance of each CMOEA is affected by a different set of features. Importantly, the footprints provide strong supporting visual evidence as to which characteristics are necessary if any new proposed benchmarks are to significantly challenge CMOEAs.

The remainder of the paper is organized as follows: Section II presents a detailed discussion of related work and thoroughly outlines the ISA methodology followed in this study. Specifically, we describe the ISA methodology and its components, which include a definition of CMOPs, benchmark suites, landscape features, CMOEAs, and performance metrics. Section III introduces the multi-objective-violation landscape and describes new Landscape Analysis features designed to help characterize this space. Section IV describes the experimental setup. The results are presented and discussed in Section V. Finally, Section VI concludes the paper.

## II. Instance Space Analysis

Instance Space Analysis (ISA) traces its foundations to Rice's framework for solving the *Algorithm Selection Problem* [17], which suggests the construction of a selection mapping between measurable features of a problem and a set of suitable algorithms; and Wolpert and Macready's *No-Free Lunch theorems* [18], which state that an algorithm is unlikely to outperform all other algorithms on all possible instances. Figure 1 illustrates ISA's framework, which has at its core six component *spaces* or sets [10]: (a) the *problem space*, $\mathcal{P}$, containing all the relevant instances of a problem in an application domain; (b) a *subset of instances*, $\mathbf{I}$, for which we have meta-data from computational experiments; (c) the *feature space*, $\mathcal{F}$, which includes features used to characterize the mathematical and statistical properties of the instances; (d) the *algorithm space*, $\mathcal{A}$, representing the set of algorithms available to solve all instances in $\mathbf{I}$; (e) the *performance space*, $\mathcal{Y}$, composed of a measure of the computational effort to obtain a satisfactory solution; and (f) the *instance space*, a 2-dimensional visualization that aids in the observation of trends in hardness for different algorithms, and facilitates insights into the distribution of existing instances.

The remainder of this section describes the details for each one of the spaces in the ISA framework, tailored specifically for CMOPs. We start with $\mathcal{P}$ by formally defining a CMOP. Then, we present $\mathbf{I}$, drawn from seven commonly used benchmark suites and a real-world suite. Next, we describe $\mathcal{F}$, where we summarize the features used for characterizing CMOPs. We follow by describing $\mathcal{A}$, by briefly presenting the 15 algorithms under test, and $\mathcal{Y}$ by formally defining the *hyper-volume* and *IGD*$^+$, our performance metrics.

This article has been accepted for publication in IEEE Transactions on Evolutionary Computation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2022.3208595



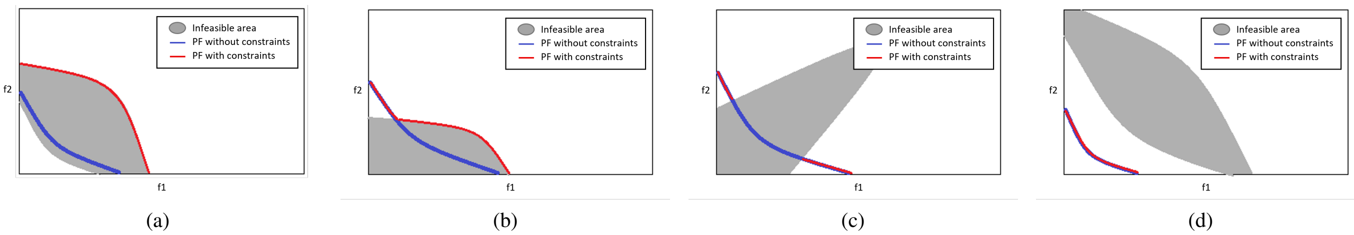



Fig. 2. Effect of constraints on the $PF$: (a) The $UPF$ is no longer feasible, and the true $PF$ lies completely on bounds of the feasible region; (b) part of the $UPF$ is no longer feasible, and parts of the true $PF$ lies on bounds of the feasible region; (c) the true $PF$ is only a proportion of the $UPF$; and (d) an infeasible region blocks the way toward the $PF$.

### A. Problem Space

A CMOP can be defined as finding a vector of decision variables that optimizes a set of objective functions and satisfies a set of restrictions. Without loosing generality, we assume minimization. A CMOP can be mathematically defined as follows:

$$\left.\begin{array}{lll} \min & f_m(x), & m=1,2,\ldots,M \\ \text{s.t.} & g_j(x) \geq 0, & j=1,2,\ldots,J \\ & h_k(x)=0, & k=1,2,\ldots,K \\ & x_i^{(L)} \leq x_i \leq x_i^{(U)}, & i=1,2,\ldots,n \end{array}\right\} \quad (1)$$

where a solution $x \in \mathbb{R}^n$ is a vector of $n$ decision variables, $f(x)$ is a vector of $M$ objectives to be optimized, $g(x)$ is set of $J$ inequality constraints, while $h(x)$ is set of $K$ equality constraints, and $x_i^{(L)}$ and $x_i^{(U)}$ represent respectively the lower and upper bounds of a decision variable $x_i$. The constraints' violation of a solution $x$ can be calculated by using the following equation:

$$CV(x) = \sqrt{\sum_{j=1}^{J} G_j(x)^2 + \sum_{k=1}^{K} H_k(x)^2} \quad (2)$$

where

$$G_j(x) = \max\left(0, g_j(x)\right) \quad (3)$$

and

$$H_k(x) = \max\left(0, \left|h_k(x)\right| - \varepsilon\right) \quad (4)$$

where $\varepsilon$ is a small value $\left(10^{-4}\right)$ to relax the equality constraints. A solution to the problem is *feasible* when $CV(x)=0$, otherwise the solution is *infeasible*.

A solution $x \in \mathbb{R}^n$ is said to be *Pareto optimal* if there is no other feasible solution $y \in \mathbb{R}^n$ such that $f(y) \prec f(x)$, where $\prec$ indicates the Pareto dominance relation. That is, a feasible solution $x$ dominates a feasible solution $y$ if and only if $(\forall m)\, f_m(x) \leq f_m(y)$ and $(\exists m)\, f_m(x) < f_m(y)$. Because of the conflicting nature of the multiple objectives, optimizing one objective function may lead to a degradation in another. Therefore, a single optimal solution may no longer be found, but instead a set of trade-off solutions, the so-called Pareto optimal set ($PS$), i.e., $PS = \{x^* \in \mathbb{R}^n | \nexists x \in \mathbb{R}^n, f(x) \prec f(x^*)\}$. Those set of solutions represent the Pareto optimal front ($PF$) in the objective space, i.e., $PF = \{f(x), x^* \in PS\}$.

When solving a MOP, an algorithm aims to find an estimated front, $\widetilde{PF}$, that has *converged*, i.e., is as close as possible to the $PF$, and is *diverse*, i.e., represents the whole $PF$. In CMOPs, the feasibility of the solution(s) must also be considered. The true, constrained, $PF$ of a CMOP can be determined by the unconstrained $PF$ ($UPF$) and bounds of the feasible region in the objective space. Having a low proportion of feasible region typically adds to the challenge/difficulty of the search process. In addition, the infeasible region may affect the shape of the $PF$ or split it into many segments, which may impact the algorithm's ability to provide diversity in its solutions. That is, the infeasible region may block the trajectory of the search towards $PF$, limiting convergence [2]. Figure 2 illustrates examples of difficulties caused by constraints.

### B. Subset of instances

If the evaluation of the performance of an algorithm is to be meaningful in practice, test problems should cover as many characteristics of real-world problems as possible. Several CMOP benchmarks have been designed with this goal in mind. Ma and Wang [19] proposed a classification of CMOPs depending on the relationship between $UPF$ and the $PF$:

**Type I** where the $PF$ is same as the $UPF$.
**Type II** where the $PF$ is part of the $UPF$.
**Type III** where the $PF$ contains all or part of the $UPF$ and solutions on the boundary of the feasible region.
**Type IV** where the $PF$ is completely located on the boundary of the feasible region.

In this study, we have used a range of diverse benchmark suites with a wide range of characteristics, which are summarized in Table I. Specifically, six inequality constrained benchmark suites: CF [20], C-DTLZ [21], DC-DTLZ [22], LIR-CMOP [23], DAS-CMOP [2], and MWs [19], and an equality constrained suite: Eq-DTLZ [24]. In addition, we have used a real-world suite, RWMOP [1], to compare the characteristics of synthetic benchmarks with the real-world problems.

### C. Feature Space

Landscape Analysis are methods used to quantify the characteristics of a problem's landscape, which is described as a surface in the search space that defines a certain aspect of the problem, such as fitness for each potential solution [5]. Stadler [25] defined a general form of the fitness landscape for a problem as the triplet $(X, N, f)$, where $X$ is a set of potential solutions, $f$ is a fitness function, and $N$ is a notion of

TABLE I

CHARACTERISTICS OF THE BENCHMARKS EMPLOYED IN THIS STUDY. ALL OF THEM HAVE A SCALABLE NUMBER OF DECISION VARIABLES. *Type* DESCRIBES THE RELATIONSHIP BETWEEN *UPF* AND *PF*, *M* IS THE NUMBER OF OBJECTIVES, *CF* IS THE NUMBER OF CONSTRAINTS, *UPF* AND *PF* COLUMNS DEFINE THEIR SHAPE, AND THE SIZE AND CONNECTIVITY OF THE FEASIBLE REGION IS GIVEN IN THE LAST COLUMN. *S* IS SHORT FOR SCALABLE, *C* FOR CONTROLLABLE, *Disconn* FOR DISCONNECTED, *Conn* FOR CONNECTED, *L* FOR LARGE, AND *S* FOR SMALL.

| **Problem** | **Type** | **M** | **CF** | **UPF** | **PF** | **Feasible Region** |
|---|---|---|---|---|---|---|
| CF1 | II | 2 | 1 | Linear | Disconn | L/Conn |
| CF2 | II | 2 | 1 | Convex | Disconn | L/Conn |
| CF3 | II | 2 | 1 | Concave | Disconn | L/Conn |
| CF4 | III | 2 | 1 | Linear | Linear | L/Conn |
| CF5 | III | 2 | 1 | Linear | Linear | L/Conn |
| CF6 | III | 2 | 1 | Convex | Mixed | L/Conn |
| CF7 | III | 2 | 1 | Convex | Mixed | L/Conn |
| CF8 | II | 3 | 1 | Concave | Disconn | L/Conn |
| CF9 | II | 3 | 1 | Concave | Disconn | L/Conn |
| CF10 | II | 3 | 1 | Concave | Disconn | L/Conn |
| C1-DTLZ1 | I | S | 1 | Linear | Linear | S/Conn |
| C2-DTLZ2 | II | S | 1 | Concave | Concave | S/Disconn |
| C1-DTLZ3 | I | S | 1 | Concave | Concave | L/Disconn |
| C3-DTLZ4 | IV | S | M | Concave | Concave | L/Conn |
| DC1-DTLZ1 | II | S | 1 | Linear | Disconn | C/Disconn |
| DC1-DTLZ3 | II | S | 1 | Concave | Disconn | C/Disconn |
| DC2-DTLZ1 | I | S | 2 | Linear | Linear | S/Disconn |
| DC2-DTLZ3 | I | S | 2 | Concave | Concave | S/Disconn |
| DC3-DTLZ1 | II | S | M | Linear | Disconn | C/Disconn |
| DC3-DTLZ3 | II | S | M | Concave | Disconn | C/Disconn |
| Eq1-DTLZ1 | II | S | S | Linear | Linear | Undefined |
| Eq1-DTLZ2 | II | S | S | Concave | Concave | Undefined |
| Eq1-DTLZ3 | II | S | S | Concave | Concave | Undefined |
| Eq1-DTLZ4 | II | S | S | Concave | Concave | Undefined |
| Eq1-iDTLZ1 | II | S | S | Linear | Linear | Undefined |
| Eq1-iDTLZ2 | II | S | S | Convex | Convex | Undefined |
| Eq2-DTLZ1 | II | S | S | Linear | Linear | Undefined |
| Eq2-DTLZ2 | II | S | S | Concave | Concave | Undefined |
| Eq2-DTLZ3 | II | S | S | Concave | Concave | Undefined |
| Eq2-DTLZ4 | II | S | S | Concave | Concave | Undefined |
| Eq2-iDTLZ1 | II | S | S | Linear | Linear | Undefined |
| Eq2-iDTLZ2 | II | S | S | Convex | Convex | Undefined |
| DAS-CMOP1 | C | 2 | 11 | Concave | Disconn | C/C |
| DAS-CMOP2 | C | 2 | 11 | Mixed | Mixed | C/C |
| DAS-CMOP3 | C | 2 | 11 | Disconn | Disconn | C/C |
| DAS-CMOP4 | C | 2 | 11 | Concave | Disconn | C/C |
| DAS-CMOP5 | C | 2 | 11 | Mixed | Mixed | C/C |
| DAS-CMOP6 | C | 2 | 11 | Disconn | Disconn | C/C |
| DAS-CMOP7 | C | 3 | 7 | Linear | Disconn | C/C |
| DAS-CMOP8 | C | 3 | 7 | Concave | Disconn | C/C |
| DAS-CMOP9 | C | 3 | 7 | Concave | Disconn | C/C |
| LIR-CMOP1 | IV | 2 | 2 | Concave | Concave | S/Conn |
| LIR-CMOP2 | IV | 2 | 2 | Convex | Convex | S/Conn |
| LIR-CMOP3 | IV | 2 | 3 | Concave | Disconn | S/Disconn |
| LIR-CMOP4 | IV | 2 | 3 | Convex | Disconn | S/Disconn |
| LIR-CMOP5 | I | 2 | 2 | Convex | Convex | S/Disconn |
| LIR-CMOP6 | I | 2 | 2 | Concave | Concave | S/Disconn |
| LIR-CMOP7 | IV | 2 | 3 | Convex | Concave | S/Disconn |
| LIR-CMOP8 | IV | 2 | 3 | Concave | Concave | S/Disconn |
| LIR-CMOP9 | II | 2 | 2 | Concave | Disconn | S/Disconn |
| LIR-CMOP10 | II | 2 | 2 | Convex | Disconn | S/Disconn |
| LIR-CMOP11 | III | 2 | 2 | Convex | Disconn | S/Disconn |
| LIR-CMOP12 | III | 2 | 2 | Concave | Disconn | S/Disconn |
| LIR-CMOP13 | I | 3 | 2 | Mixed | Mixed | S/Disconn |
| LIR-CMOP14 | II | 3 | 3 | Mixed | Mixed | S/Disconn |
| MW1 | II | 2 | 1 | Linear | Disconn | S/Disconn |
| MW2 | I | 2 | 1 | Linear | Linear | S/Disconn |
| MW3 | III | 2 | 2 | Linear | Mixed | S/Conn |
| MW4 | I | S | 1 | Linear | Linear | S/Conn |
| MW5 | II | 2 | 3 | Concave | Disconn | S/Conn |
| MW6 | II | 2 | 1 | Concave | Disconn | S/Disconn |
| MW7 | III | 2 | 2 | Concave | Disconn | S/Conn |
| MW8 | II | S | 1 | Concave | Disconn | S/Disconn |
| MW9 | IV | 2 | 1 | Convex | Concave | S/Conn |
| MW10 | III | 2 | 3 | Concave | Disconn | S/Disconn |
| MW11 | IV | 2 | 4 | Concave | Disconn | S/Disconn |
| MW12 | IV | 2 | 2 | Mixed | Mixed | S/Disconn |
| MW13 | III | 2 | 2 | Disconn | Disconn | S/Disconn |
| MW14 | I | S | 1 | Disconn | Disconn | S/Conn |

neighborhood relation. The Euclidean distance is usually used in continuous optimization to quantify the solutions' relations.

CMOP involves multiple fitness and constraint functions; hence, Stadler's definition of fitness landscape cannot be directly applied in this work. Verel *et al*., [26] defined the multi-objective landscape, and Malan *et al*., [13] introduced the violation landscape. However, these two landscapes treat constraints and objectives independently. Therefore, they do not capture the interaction between them, which is essential for CMOPs. In Section III, we formally describe our proposed solution to this problem. Specifically, we propose the multi-objective-violation landscape. However, before details are presented, we position our contribution within Smith-Miles's ISA framework by providing a summary of the three landscapes and their features below. It should be noted that the Landscape Analysis features used do not require knowledge of the *PS*, apart from the HV-based features, which require a reference point.

**Multi-Objective Landscape** In multi-objective optimization, we are dealing with multiple fitness functions and a set of optimal solutions. Therefore, we use the definition of multi-objective landscape proposed by Verel *et al*., [26]. The Landscape Analysis features used to characterize the multi-objective landscape has been adopted from the literature and summarized in Table II.

**Violation Landscape** To characterize COP, Malan *et al*., [13] introduced the violation landscape, which uses the violation function to quantify a solution fitness. Here, we use the norm of constraints violation vector as calculated in Equation 2. We propose an extended set of features in section III-B to capture the underlying characteristics. The violation landscape features are summarized in Table III.

**Multi-Objective-Violation Landscape** In Section III-A, we define the multi-objective-violation landscape, which is constructed based on the interaction between constraint violation and multi-objective fitness. We also propose a set of features to characterize this landscape. Table IV lists the features of the multi-objective-violation landscape. The table includes new, modified, and adopted features.

### D. Algorithm Space

Specialized versions of multi-objective evolutionary algorithms (MOEAs) have been designed with constraints handling techniques, so called constrained multi-objective evolutionary algorithms (CMOEAs), to maintain the necessary balance between optimizing objectives and satisfying constraints in CMOP. There are three categories of CMOEAs [33]:

**Prioritize constraints** Methods in this category pressure the search toward a feasible region. However, algorithms may get trapped in a small part of the feasible region because of the bias toward infeasible solutions. Representative methods of this category include using the principle of constraint dominance such as NSGAII, DCMOEAD [34], and ANSGAIII [21], a relaxed version of constraint dominance such as $\varepsilon$-constraint [23]. ECNSGAII and

TABLE II
THE FEATURES USED TO CHARACTERIZE THE MULTI-OBJECTIVES LANDSCAPE OF CMOP.

| Type | Feature | Description | Source | Focus |
|---|---|---|---|---|
| Global | upo_n | Proportion of unconstrained PO solutions | [27] | Set-Cardinality |
| | uhv | Hypervolume-value of the $\widetilde{UPF}$ | [28] | Set-Distribution |
| | corr_obj | correlation between objective values | [29] | evolvability |
| | mean_f | Average of unconstrained ranks | [12] | y-distribution |
| | std_f | Standard deviation of unconstrained ranks | [5] | y-distribution |
| | max_f | Maximum of unconstrained ranks | [5] | y-distribution |
| | skew_f | Skewness of unconstrained ranks | [5] | y-distribution |
| | kurt_f | Kurtosis of unconstrained ranks | [5] | y-distribution |
| | kurt_avg | Average of objectives kurtosis | [5] | y-distribution |
| | kurt_min | Minimum of objectives kurtosis | [5] | y-distribution |
| | kurt_max | Maximum of objectives kurtosis | [5] | y-distribution |
| | kurt_rnge | Range of objectives kurtosis | [5] | y-distribution |
| | skew_avg | Average of objectives skewness | [5] | y-distribution |
| | skew_min | Minimum of objectives skewness | [5] | y-distribution |
| | skew_max | Maximum of objectives skewness | [5] | y-distribution |
| | skew_rnge | Range of objectives skewness | [5] | y-distribution |
| | f_mdl_r2 | Adjusted coefficient of determination of a linear regression model for varibles and unconstrained ranks | [5] | variable scaling |
| | f_range_coeff | Difference between maximum and minimum of the absolute value of the linear model coefficients | [5] | variable scaling |
| Random Walk | dist_f_avg_rws | Average distance from neighbours in the objective space | [12] | evolvability |
| | dist_f_r1_rws | First autocorrelation coefficient of dist_f_avg_rws | [12] | ruggedness |
| | dist_f_dist_x_avg_rws | Ratio of dist_f_avg_rws to dist_x_avg_rws | [12] | evolvability |
| | dist_f_dist_x_avg_r1 | First autocorrelation coefficient of dist_f_dist_x_avg_rws | [12] | ruggedness |
| | nuhv_avg_rws | Average unconstrained hypervolume-value of neighborhood's solutions | [29] | evolvability |
| | nuhv_r1_rws | First autocorrelation coefficient of nuhv_avg_rws | [29] | ruggedness |

TABLE III
THE FEATURES USED TO CHARACTERIZE THE VIOLATION LANDSCAPE OF CMOP. THE PROPOSED FEATURES MARKED AS NEW, WHILE THE (*) INDICATES THAT THE FEATURE HAS BEEN MODIFIED TO CHARACTERIZE CMOP.

| Type | Feature | Description | Source | Focus |
|---|---|---|---|---|
| Global | min_cv | Minimum of constraints violations | [5] * | y-distribution |
| | skew_cv | Skewness of constraints violations | [5] * | y-distribution |
| | kurt_cv | Kurtosis of constraints violations | [5] * | y-distribution |
| | cv_mdl_r2 | Adjusted coefficient of determination of a linear regression model for varibles and violations | [5] * | variable scaling |
| | cv_range_coeff | Difference between maximum and minimum of the absolute value of the linear model coefficients | [5] * | variable scaling |
| | dist_c_corr | Violation-distance correlation | [30] * | deception |
| Random Walk | dist_c_avg_rws | Average distance from neighbours in the constraints space | [12] * | evolvability |
| | dist_c_r1_rws | first autocorrelation coefficient of dist_c_avg_rws | [12] * | ruggedness |
| | dist_c_dist_x_avg_rws | Ratio of dist_c_avg_rws to dist_x_avg_rws | [12] * | evolvability |
| | dist_c_dist_x_r1_rws | First autocorrelation coefficient of dist_c_dist_x_avg_rws | [12] * | ruggedness |
| | ncv_avg_rws | Average single solution's violation-value | New | evolvability |
| | ncv_r1_rws | first autocorrelation coefficient of ncv_avg_rws | New | ruggedness |
| | nncv_avg_rws | Average neighborhood's violation-value | New | evolvability |
| | nncv_r1_rws | first autocorrelation coefficient of nncv_avg_rws | New | ruggedness |
| | bncv_avg_rws | Average violation-value of neighborhood's non-dominated solutions | New | evolvability |
| | bncv_r1_rws | first autocorrelation coefficient of bncv_avg_rws | New | ruggedness |

ECMOEAD apply an improved version of $\varepsilon$-constraint which has been proposed in [35].

**Consider objectives and constraints equally** A method belonging to this category treats constraints as part of the objective functions [36] by including static or dynamic penalty factor in the objectives [37] such as $\varepsilon$-constraint dynamic penalty [38], which has been used in PECNSGAII and PECMOEAD. Other methods objectivize the constraints [39], or switch between dominance relation to compare constraints and dominance to compare objectives by using stochastic ranking [40], which has been implemented in SRNSGAII and SRCMOEAD. Alternatively they use the status of the search such as CMOEA_MS [33]. These approaches provide good balance between exploring feasible and non-feasible regions, however, they may suffer in convergence.

**Hyper-strategies** Methods in this category use different strategies in different populations or stages. They aim to balance objectives and constraints by favoring one or both in each stage of the search or in different populations. For example, CTAEA [22] uses two archives, one to maintain convergence by optimizing both constraints and objectives, while the second archive is used to maintain diversity, and it considers optimizing objectives only. On the other hand, CCMO [41] uses two populations, one to solve the original CMOP and another to solve a helper problem derived from the original one. Another approach is to use multiple stages of the search, MOEADDAE [38] uses the first stage to push the search toward feasible solutions by prioritizing constraints and the second stage

TABLE IV
THE FEATURES USED TO CHARACTERIZE THE MULTI-OBJECTIVES-VIOLATION LANDSCAPE OF CMOP. THE PROPOSED FEATURES MARKED AS NEW, WHILE THE (*) INDICATES THAT THE FEATURE HAS BEEN MODIFIED TO CHARACTERIZE CMOP.

| Type | Feature | Description | Source | Focus |
|---|---|---|---|---|
| Global | fsr | Feasibility ratio | [13] | Set-Cardinality |
| | po_n | Proportion of PO solutions | [27] | Set-Cardinality |
| | hv | Hypervolume-value of the $\widetilde{PF}$ | [28] | Set-Distribution |
| | cpo_upo_n | Proportion of $\widetilde{PF}$ to $\widetilde{UPF}$ | New | PF and UPF correlation |
| | hv_uhv_n | Proportion of HV to unconstrained HV | New | PF and UPF correlation |
| | GD_cpo_upo | distance between $\widetilde{PF}$ and $\widetilde{UPF}$ | New | PF and UPF correlation |
| | cover_cpo_upo | Proportion of $\widetilde{UPF}$ covered by $\widetilde{PF}$ | New | PF and UPF correlation |
| | corr_cobj_min | Minimum constraints and objectives correlation | [16] | evolvability |
| | corr_cobj_max | Maximum constraints and objectives correlation | [16] | evolvability |
| | corr_cf | Constraints and ranks correlation | [13] * | evolvability |
| | piz_ob_min | Minimum proportion of solutions in ideal zone per objectives | [13] * | Optima isolation |
| | piz_ob_max | Maximum proportion of solutions in ideal zone per objectives | [13] * | Optima isolation |
| | piz_f | Proportion of solutions in ideal zone | [13] * | Optima isolation |
| | ps_dist_max | Maximum distance across PS | [27] | PS connectivity |
| | ps_dist_mean | Average distance across PS | [31] | PS connectivity |
| | ps_dist_iqr_mean | Average difference between 75th and 25th percentiles of distances across PS | [31] | PS connectivity |
| | pf_dist_max | Maximum distance across PF | [32] | PF discontinuouty |
| | pf_dist_mean | Average distance across PF | [32] | PF discontinuouty |
| | pf_dist_iqr_mean | Average difference between 75th and 25th percentiles of distances across PF | [32] | PF discontinuouty |
| Random Walk | sup_avg_rws | Average proportion of neighbors dominating the current solution | [29] | evolvability |
| | sup_r1_rws | First autocorrelation coefficient of sup_avg_rws | [29] | ruggedness |
| | inf_avg_rws | Average proportion of neighbors dominated by the current solution | [29] | evolvability |
| | inf_r1_rws | First autocorrelation coefficient of inf_avg_rws | [29] | ruggedness |
| | inc_avg_rws | Average proportion of neighbors incomparable to the current solution | [29] | evolvability |
| | inc_r1_rws | First autocorrelation coefficient of inc_avg_rws | [29] | ruggedness |
| | lnd_avg_rws | Average proportion of locally non-dominated solutions in the neighborhood | [29] | evolvability |
| | lnd_r1_rws | First autocorrelation coefficient of lnd_avg_rws | [29] | ruggedness |
| | dist_x_avg_rws | Average distance from neighbours in the variable space | [12] | evolvability |
| | dist_x_r1_rws | First autocorrelation coefficient of dist_x_avg_rws | [12] | ruggedness |
| | dist_f_c_avg_rws | Average distance from neighbours in the objective-constraints space | [12] * | evolvability |
| | dist_f_c_r1_rws | First autocorrelation coefficient of dist_f_c_avg_rws | [12] * | ruggedness |
| | dist_f_c_dist_x_avg_rws | Ratio of dist_f_c_avg_rws to dist_x_avg_rws | [12] * | evolvability |
| | dist_f_c_dist_x_avg_r1 | First autocorrelation coefficient of dist_f_c_dist_x_avg_rws | [12] * | ruggedness |
| | nhv_avg_rws | Average hypervolume-value of feasible neighborhood's solutions | [29] * | evolvability |
| | nhv_r1_rws | First autocorrelation coefficient of nhv_avg_rws | [29] * | ruggedness |
| | bhv_avg_rws | Average hypervolume-value of neighborhood's non-dominated solutions | [29] * | evolvability |
| | bhv_r1_rws | First autocorrelation coefficient of bhv_avg_rws | [29] * | ruggedness |
| | nfronts_avg_rws | Average number of ranks | New | evolvability |
| | nfronts_r1_rws | first autocorrelation coefficient of nfronts_avg_rws | New | ruggedness |
| | rfbx_rws_avg | Average ratio of feasible boundary crossings | [13] | Dispersion of the feasible regions |

to favor objectives in order to escape local optima, whilst PPS [35] pushes the search towards $UPF$, then, pulls it to the feasible region. ToP [42] converts CMOP into COP in the first stage, then uses a CMOEA in the second stage.

### E. Performance Space

The most commonly used performance indicators when optimizing CMOPs are the hypervolume ($HV$) [1], [43], and the inverted generational distance ($IGD^+$) [44], which evaluate the convergence and diversity of the $\widetilde{PF}$. The $HV$ quantifies the volume of the objective space covered by $\widetilde{PF}$ and a reference point to measure $\widetilde{PF}$ convergence and distribution. The reference point, $r$, is a vector that has objective values worse than any values in the $\widetilde{PF}$. To overcome $HV$ bias, a common reference point $r = (1.1, \ldots, 1.1)^{(T)}$ is used with the normalized $PF$ and objectives [45]. The larger the value of the $HV$, the better the approximation of the true $PF$. The second performance indicator, $IGD^+$, evaluates the convergence and diversity of $\widetilde{PF}$ by measuring the distance between a $PF$ and the dominated points in its approximation. The closer the value to zero the better. $IGD^+$ requires a reference set and can only be used in problems with known $PF$.

## III. CHARACTERIZING CMOP LANDSCAPES

In this section, we describe the concept of the multi-objective-violation landscape in detail. We also propose a set of local structured-based Landscape Analysis features collected by random walks, and global unstructured-based Landscape Analysis features approximated by random samples [46] for the multi-objective-violation landscapes and the violation landscapes.

### A. Multi-Objective-Violation Landscapes

The multi-objective-violation landscape replaces Stadler's fitness function, $f$, by using the constraint domination principle [34] to measure the quality of solutions in the search space. Given two solutions $x$ and $y$, $x$ is said to have better quality or higher rank than $y$ if any of the following conditions is true: (a) the solution $x$ is feasible and the solution $y$ is not;

(b) both solutions are infeasible but $x$ has smaller constraint violation norm; or (c) $x$ and $y$ are both feasible or have similar constraint violation norm, but $x$ dominates $y$ w.r.t. objectives only.

Here, we propose six features to characterize a multi-objective-violation landscape. As described in section II-B, the relationship between the $PF$ and the $UPF$ may cause some difficulty; therefore, four new features have been proposed to quantify the relationship between an approximation of the $PF$ ($\widetilde{PF}$) and an approximation of the $UPF$ ($\widetilde{UPF}$) by using a random sample:

1) Proportion of $PF$ to $UPF$ (cpo_upo_n) approximates the size of the constrained non-dominated solutions set in relation to the unconstrained non-dominated solutions set. Given a random sample of $n$ points, $cpo\_upo\_n$ is defined as:
$$cpo\_upo\_n = \frac{|po|}{|upo|} \tag{5}$$
2) Proportion of HV to unconstrained HV (hv_uhv_n) measures the $hv$ in relation to the volume of the objective space covered by the $\widetilde{UPF}$ (uhv). Given a random sample of $n$ points, $hv\_uhv\_n$ is defined as:
$$hv\_uhv\_n = \frac{hv}{uhv} \tag{6}$$
3) Distance between $PF$ and $UPF$ (GD_cpo_upo) measures the distance between $\widetilde{PF}$ and $\widetilde{UPF}$ by using the generational distance metric [47] as follows:
$$GD\_cpo\_upo = \frac{1}{|po|}\left(\sum_{s'' \in po} d_{s''}^2\right)^{\frac{1}{2}} \tag{7}$$
where
$$d_{s''} = min_{s' \in upo}|F(s'') - F(s')| \tag{8}$$
where $F(s'')$ and $F(s')$ are vectors of solutions objectives, and $d_s''$ is the smallest distance from each solution in the $\widetilde{PF}$ to the nearest solution in the $\widetilde{UPF}$.
4) Proportion of unconstrained PF covered by PF (cover_cpo_upo) approximates how many solutions in $\widetilde{UPF}$ are dominated or equal to solutions in $\widetilde{PF}$. Given a random sample of $n$ points, $cover\_cpo\_upo$ is defined as:
$$cover\_cpo\_upo = \frac{|\{s' \in \widetilde{UPF}; \exists s'' \in \widetilde{PF} : s' \preceq s''\}|}{|\widetilde{UPF}|} \tag{9}$$
if $cover\_cpo\_upo$ = 1, that means all the solutions in $\widetilde{UPF}$ are equal to the solutions in $\widetilde{PF}$, while $cover\_cpo\_upo$ = 0, means the $\widetilde{UPF}$ strictly dominates the $\widetilde{PF}$ [48].

The remaining two features required to help characterize the multi-objective-violation landscape are collected by a random walk as measures of evolvability and ruggedness of the landscape. They are the average number of the solutions' ranks based on constraints domination principle [34] in the neighborhood (nfronts_rws) and its first auto-correlation coefficient.

We also adapt the following features from the multi-objective or the violation landscapes in order to include multi-objective and constraints concepts together. The features are divided into four groups:

1) Constraints and objectives correlation [13]: given a random sample of $n$ points, the correlation between the solutions' rank based on constraints domination principle [34] and the solutions' $CV$ (corr_cf) is calculated using Spearman's rank correlation coefficient, where the range of the correlation coefficients are between [-1,1].
2) Proportion of solutions in ideal zone (piz) [13]: quantifies the proportion of points in the lower quadrant of a fitness-violation scatter plot. The lower quadrant is bounded by the ideal point which is a pair of ideal fitness and ideal $CV$. The ideal fitness or $CV$ (id) is given by the following formula:
$$id = min(S) + (0.25(max(S) - min(S))) \tag{10}$$
where $S$ is the set of solution's fitness or violation. The (piz) is calculated for each objective and for the solutions' rank. Then, the feature (piz_ob_min) quantifies the minimum proportion of solutions in the ideal zone per objectives, while (piz_ob_max) gives the maximum value. Also, (piz_f) approximates the proportion of overall good solutions in ideal zone.
3) Distance among neighbours in the objective-violation space and the variable space [12]: the average euclidean distance from each solution to its neighbours in the objective-violation space (dist_f_c_rws) and in the variable space are calculated, as well as the ratio of (dist_f_c_rws) to (dist_x_rws) (dist_f_c_dist_x_rws). The average value as well as the first autocorrelation coefficient of these features are measured.
4) Hypervolume-value of the neighborhood [12]: in a random walk, the hypervolume-value of the feasible set in each neighborhood is quantified (nhv_rws), and hypervolume-value of neighborhood's non-dominated set (bhv_rws). Then, both the average value and the first autocorrelation coefficient for each feature are measured.

## B. Violation Landscapes

To better characterize constrained optimization problems, we propose six features to measure ruggedness and evolvability. From a random walk, we propose calculating the average and first auto-correlation coefficient for each of the following:

1) Single solution's violation-value (ncv_rws) simply measures the $CV$ of the current solution in a sample collected by random walk.
2) Average neighborhood's violation-value (nncv_rws) is given by:
$$nncv\_rws = \frac{\sum_{x \in S} CV(x)}{|S|} \tag{11}$$
where $|S|$ is the set of solutions in the neighborhood.
3) Average violation-value of neighborhood's non-dominated solutions (bncv_rws) is given by:
$$bncv\_rws = \frac{\sum_{x \in S'} CV(x)}{|S'|} \tag{12}$$

where $|S^{'}|$ is the set of non-dominated solutions in the neighborhood.

In addition, we modified the following features from the fitness landscape domain to quantify the characteristics of the violation landscape. The features are divided into four groups:

1) Solutions' constraints violations distribution measures [5]: for a random sample of $n$ points, the solutions' $CV$ are calculated, then, a set of y-distribution features are calculated, which are: minimum (min_cv), skewness (skew_cv), and kurtosis (kurt_cv) of the $CVs$.
2) Linear model coefficients [5]: linear regression model is fitted to solutions' $CV$ and decision variables. The adjusted coefficient of determination of the model (cv_mdl_r2), and difference between maximum and minimum of the absolute value of the linear model coefficients (cv_range_coeff) are calculated.
3) Violation-distance correlation [30]: given a random sample of $n$ points, for each point a pair of $CV$ and the euclidean distance to the nearest global optima is calculated. Then, the Spearman's rank correlation coefficient is calculated for the set of ($CV$, distance) pairs to measure (dist_c_corr).
4) Distance among neighbours in the violation space [12]: in a sample collected by random walk, The average euclidean distance from each solution to its neighbours in the violation space (dist_c_rws) is calculated, as well as the ratio of (dist_c_rws) to the average distance in the decision space (dist_c_dist_x_rws). We compute both the average value as well as the first autocorrelation coefficient of these features.

## IV. Experimental Setup

We have used a total of 443 bi-objective instances to explore the characteristics of CMOPs and to study their impact on the performance of CMOEAs. Instances belong to the eight benchmark suites described in Section II-B, with $n = \in \{2, 5, 10\}$, with the exception of for CF, where $n = 2$ cannot be used, and for RWMOP, where $n$ can not be controlled. For the DAS-CMOP suite, 15 instances were generated from each problem at each value of $n$ by varying the constraints parameters to adjust difficulty.

To extract the features, for each dimension, 30 samples sets of size $n \times 10^3$ were generated. The average of features were then calculated. Global features used random sampling, while local features depend on random walks. A random walk of neighborhood size $N = (2 \times n) + 1$, length $(n/N) \times 10^3$, and step size of 2% of the range of the instance domain. Then, the features were processed using the Yeo-Johnson power transform method, which resulted in a distribution closer to Gaussian.

We have tested 15 algorithms, five from each category of the CHTs described in II-D. NSGAII, ANSGAIII, CMOEA_MS, CTAEA, CCMO, MOEADDAE, PPS, and ToP are available through the PlatEMO [49] library, while DCMOEAD, ECNSGAII, ECMOEAD, PECNSGAII, PECMOEAD, SRNSGAII, and SRCMOEAD have been implemented as described in Section II-D. All MOEAD-based algorithms used Tchebycheff approach. We have used algorithms' default parameters. The population size set to be 200 with all instances, while number of evaluations is set to be $2 \times 10^4$ for $n = 2$ instances and $5x10^4$ for $n = \{5, 10\}$. For each algorithm and each instance, 30 independent runs were conducted. The average Max-Min normalized value has been calculated for the performance metric. Indicators values are normalized to be between [0,1] where 1 is the best value. We use a binary concept to define the 'goodness' of the measured performance with respect to others [7]. We consider the performance of an algorithm as a 'good' performance if the performance metric ,normalized $HV$ or $IGD^{+}$, is greater than zero and within 1% of the best algorithm on the same instance.

After collecting the meta-data, a subset of the features that impact algorithms' performance were selected. Our selection strategy is to filter out features that are weakly correlated with all algorithms, i.e., when an absolute value of the Pearson correlation is less than 0.3 [10] with all algorithms. Then, when a feature is highly predictive to another feature, one of them will be eliminated to reduce redundancy, i.e., the absolute value of Pearson correlation of two features is greater than 0.85. We then use random forest regressor (RF) to keep only the features that are predictive of algorithms' performance. Hyperparameter tuning and 3-fold cross validation were used to build more accurate and stable RF models. Finally, to construct the instance space, we make use of the publicly available web tools in MATILDA [50].

## V. Results

### A. Instance Space

Figure 3 illustrates the 2-dimensional CMOP instance space, where each instance is represented as a point. The location of each instance is defined by the following projection matrix:

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 0.2559 & 0.1348 \\ -0.2469 & -0.1649 \\ -0.0257 & -0.2703 \\ 0.2938 & -0.2278 \\ -0.2148 & -0.1338 \\ -0.1935 & -0.2210 \\ -0.1651 & 0.2998 \\ -0.2150 & 0.3137 \\ 0.3067 & 0.1382 \\ 0.0709 & 0.3047 \\ 0.2032 & -0.0515 \\ 0.1436 & 0.2869 \\ 0.1940 & 0.1154 \\ -0.0508 & -0.2466 \end{bmatrix}^{\top} \begin{bmatrix} corr\_cf \\ f\_mdl\_r2 \\ dist\_c\_corr \\ min\_cv \\ bhv\_avg\_rws \\ skew\_rnge \\ piz\_ob\_min \\ ps\_dist\_iqr\_mean \\ dist\_c\_dist\_x\_avg\_rws \\ cpo\_upo\_n \\ cv\_range\_coeff \\ corr\_obj \\ dist\_f\_dist\_x\_avg\_rws \\ cv\_mdl\_r2 \end{bmatrix} \quad (13)$$

which uses the features with the highest correlation with algorithm performance. The list of features in Equation 13 corresponds to the common features identified when using both the HV and IGD$^{+}$ performance metrics.

An inspection of the figure reveals that the real-world problems are distributed throughout the instance space, with the exception of the upper-right area, that suggests instances in that part are not representative of real-world problems. Furthermore, most test suites are distributed over specific parts of the instance space. This is expected, as instances in the same suite usually share similar objective or constraint functions.

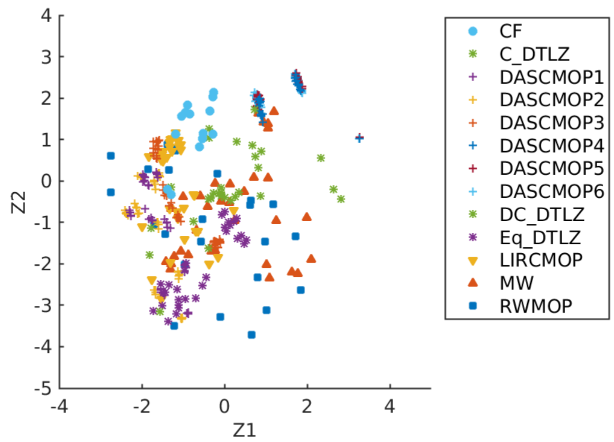



Fig. 3. Distribution of the instances in 2D space by using the projection matrix in Equation (13). Instances are color and shape coded based on the source.

For example, instances from DAS-CMOP1, DAS-CMOP2, and DAS-CMOP3 have similar constraint functions, while DC-DTLZ instances have either DTLZ1 or DTLZ3 objective functions. We observe that there is a paucity of instances in the bottom-right side of the space, that indicates a lack of diversity in some features. That is despite the existence of real-world problems in that area. By observing the features' distribution in figure 6, we note that there is a lack of instances with highly negative $corr\_cf$ and instances that have high $cpo\_upo\_n$. Also, there is high density of instances in some regions, attributed to the DAS-CMOP suite. We note that changing the constraints parameters of such problems does not have impact on the difficulty or diversity of instances. However, changing $n$ does have an impact.

### B. Comparing Performance Metrics

Figures 4a and 4b present a comparison between performance metrics, HV vs IGD$^+$. Each plot illustrates the number of algorithms that performed well on each instance as a proxy of their difficulty. Darker colors of the points in the plots corresponds to fewer algorithms performing well. The plots suggests that the instances in the top-right area are generally easier to solve by most of the algorithms, especially for the instances near the origin of the instance space. It is important to note that the equality benchmark suite resides in the hard to solve' area, which is to be expected as equality constraints are known to be challenging. As observed in the plots, $IGD^+$ has a slightly larger easy instances percentage.

Given that insights gleaned from the performance metrics are not significantly different, the remaining analysis will be based on HV results, consistent with the approach used by Zhou and co-workers [51].

### C. Algorithms Footprints

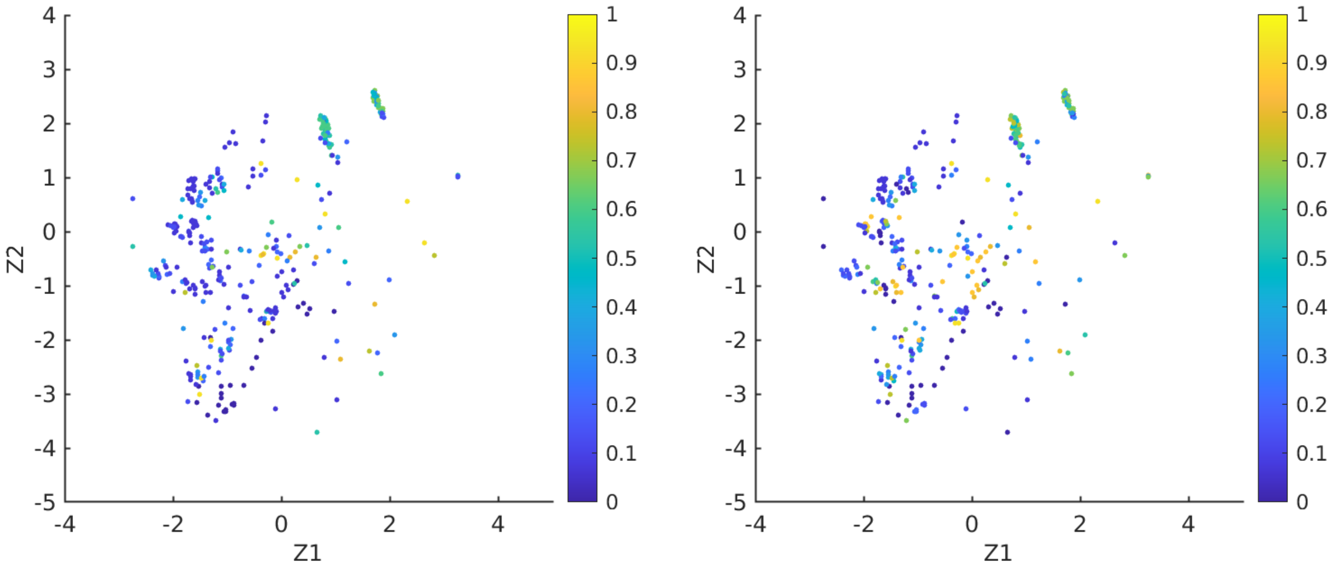
Figure 5 shows the footprints of the algorithms in the instance space. A grey point means the algorithm performed badly compared to others on such instances, while dark blue represents good performance. We limit our analysis to only eight algorithms, as footprints reveal high similarity between many of them. ANSGAIII, NSGAII and MOEAD with the principle of constraint dominance, $\varepsilon$-constraint, and stochastic ranking share similar footprints. The footprints of this group are represented by the footprints of NSGAII in 5a. The figure shows that they are capable of providing relatively good performance in only a third of the instance space. The footprints of CMOEA_MS in 5b, which depends on the principle of constraint dominance but sometimes includes the constraint violation as an objective, matches the good area of NSGAII and has a good performance in part of the instances in the left area. Penalty based algorithms (PECNSGAII, PECMOEAD) in figure 5c have similar footprints; both of them matched the best algorithm in instances located near the origin or on the upper-left area. Whilst, MOEADDAE in 5d, which uses two stages to relax the penalty factor, performed well in almost all the area covered by penalty based algorithms, and matched part of the first group footprints.

On the other hand, CTAEA, ToP, CCMO, and PPS have distinctive footprints. CTAEA and ToP have a low proportion of good performance, but they have different footprints. CTAEA in Figure 5e seems only capable of providing high quality solutions in easy to solve instances, while ToP in 5f targeted instances that are rarely solved by previous algorithms. CCMO and PPS, in figures 5g and 5h respectively, appear to be the only algorithms that performed well in a wide area of the instance space. Moreover, they have almost opposite footprints. Both algorithms use two strategies to handle constraints, the first strategy is considering objectives only, but for the second CCMO uses the principle of constraint dominance while PPS uses $\varepsilon$-constraint. CCMO uses the two strategies in parallel by having two populations, while PPS uses them sequentially by applying the first strategy for several generations, then, applying the second strategy.

### D. Features Impact

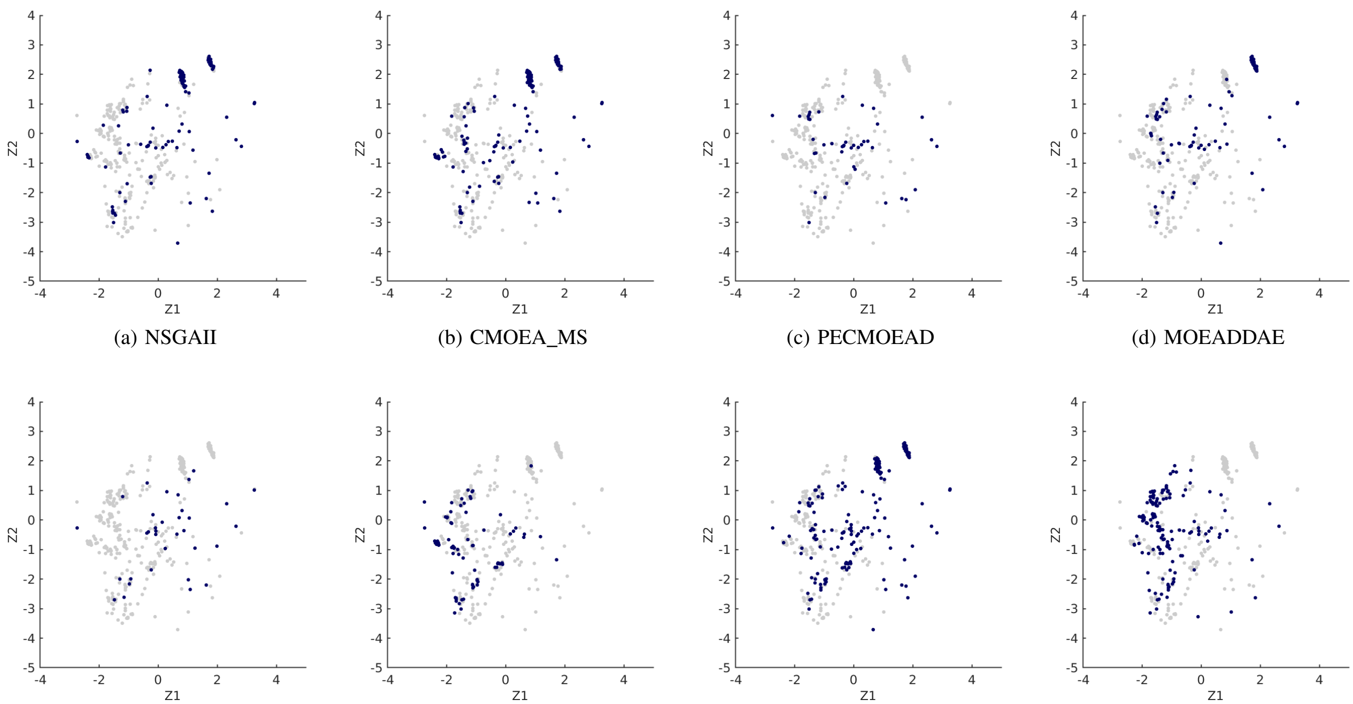
Within the instance space, we can gain insights into an algorithm's strength and weakness by examining the distribution of features across the space. Here, we present a subset of features that better explain how easy or difficult an instance is for at least one algorithm. Figures 6a and 4a show that instances that have high positive correlation between constraints and objectives are easier to solve. This suggests that the evolutionary trajectory of the search in those instances is not affected by the infeasible area; a search directed by objectives or constraints will probably lead directly to the optimal set of solutions. $cv\_range\_coeff$ is another feature that can identify instances that may be easy to solve as shown in Figure 6f, a large value indicates that there is at least one decision variable carries most of the violation weight. In addition, Figures 6b and 6e suggests that a smaller proportion of $cpo\_upo\_n$ or $piz\_ob\_min$, representing isolation of the non-dominate set or a narrow feasible area, causes difficulty for most algorithms.

Penalty-based algorithms do not have clear footprints in the represented instance space. The group represented by NSGAII,




(a) HV (b) IGD$^+$

Fig. 4. Number of algorithms performed well for each instance, where good performance means a normalized performance indicator within 1% of the best algorithm. The color scale corresponds to the total number of algorithms. A color closer to dark blue means fewer number of algorithms performed well.

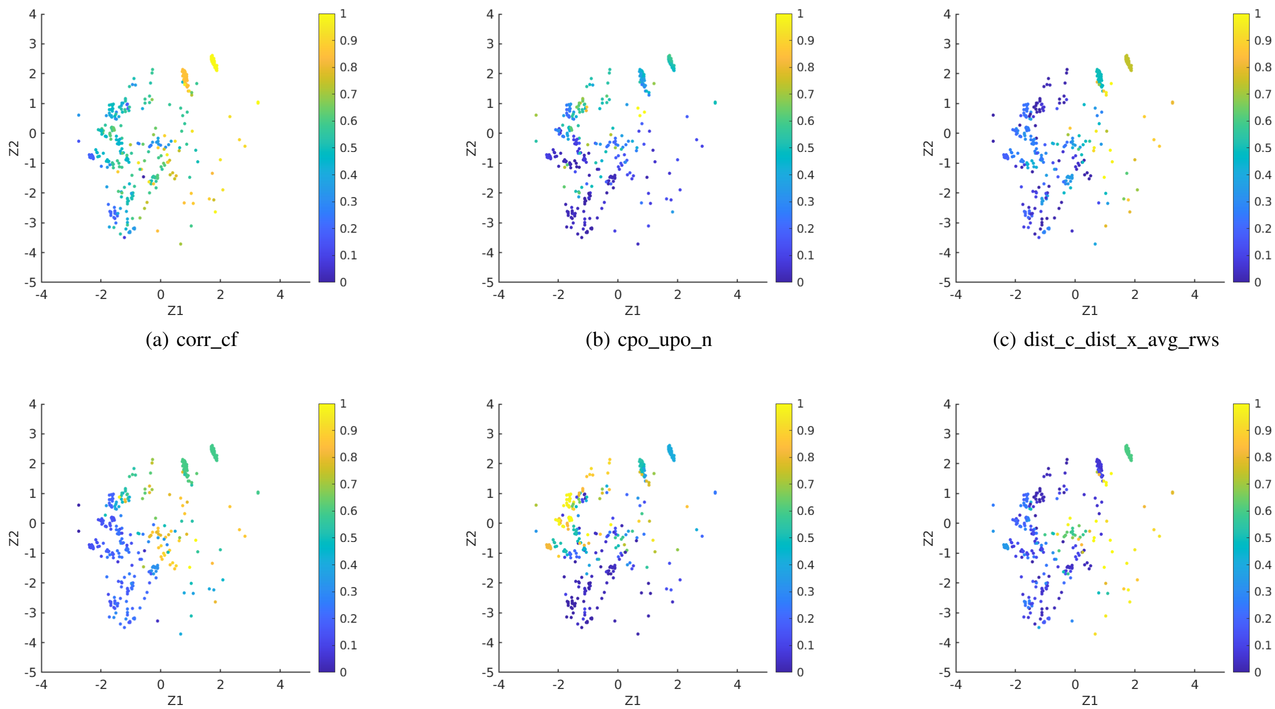



(a) NSGAII (b) CMOEA_MS (c) PECMOEAD (d) MOEADDAE

(e) CTAEA (f) ToP (g) CCMO (h) PPS

Fig. 5. Eight algorithms footprints in the projected instance space. Dark colored points represent good performance, where a good performance is defined as a normalized $HV$ within 1% of the best algorithm.

illustrated in Figure 5a, and CMOEA_MS in Figure 5b find it easier to solve a problem if the average ratio of the distance between neighbors in the violation space to the distance in the decision space is not low, as shown in Figure 6c. This suggests the presence of large, neutral areas in the violation landscape. CCMO footprints seem to overlap with the distribution of *dist_f_dist_x_avg_rws*, illustrated in Figure 6d. The higher this feature is, the more likely it is that CCMO 5g will succeed. Moreover, CCMO seems to be capable of finding solutions in instances that have a low ratio of solutions in the ideal zone of one objective, as shown in Figure 6e, meaning that CCMO has the ability to find isolated optima. Although,




(a) corr_cf (b) cpo_upo_n (c) dist_c_dist_x_avg_rws

(d) dist_f_dist_x_avg_rws (e) piz_ob_min (f) cv_range_coeff

Fig. 6. Distribution of normalized subset of features in the projected instance space. The color scale corresponds to normalized feature values.

this feature has the opposite impact on PPS, as observed in Figure 5h, suggesting that PPS is better suited to find diverse solutions when there are more feasible solutions, which is also supported by the low values of *dist_c_dist_x_avg_rws* as illustrated in Figure 6c. Furthermore, when there is negative correlation between constraints and objectives, illustrated in Figure 6a, PPS is one of the best performing algorithms. We can observe that the instances that have the highest conflict between constraints and objectives, are the instances that ToP was capable to excel at, as shown in Figure 5f.

### E. A Step Towards Algorithm Selection

Algorithm selection is the process of selecting an algorithm from a set based on its expected performance to optimize a specific instance [52]. In order to map an algorithm to an instance, the selector must understand the algorithm's general behavior with similar instances. This is where informative landscape features come in handy, as they can distinguish instances from each other. In the previous sections, we have visualized the similarities and differences between instances by using informative features, and highlighted algorithms' strengths and weaknesses on the instance space. This information can then be used by a classifier to partition the instance space and determine which algorithm is best suited for each part. This suggests that it should be possible to generate automated algorithm recommendations for untested instances based on its location in the instance space. Here, we will examine whether a machine learning classifier, trained on the set of features in 13 and algorithms performances, might be able to provide insights into the mapping of particular algorithm to part of the instance space.

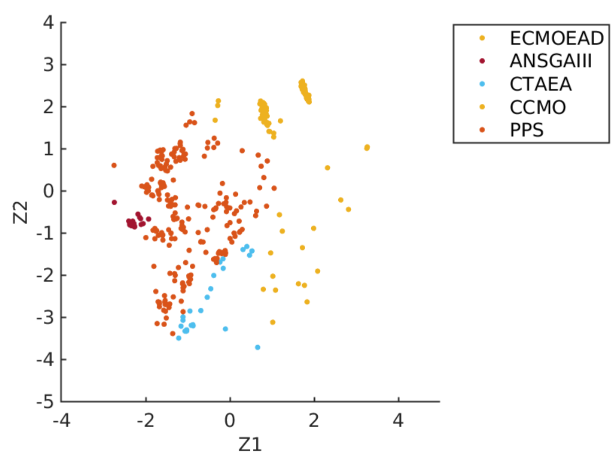



Fig. 7. Algorithm recommendations by SVM selection model for the projected instance space.

Figure 7 presents the SVM results generated by the MATILDA web tools [50] using default settings. The figure shows that hyper-strategies are more likely to be selected by the SVM model because they surpassed others in larger and clearer regions. The instance space is almost divided between CCMO and PPS, however, we noticed that PPS has been selected for the instances around origin even though CCMO succeed in this region. There are small area in the bottom-

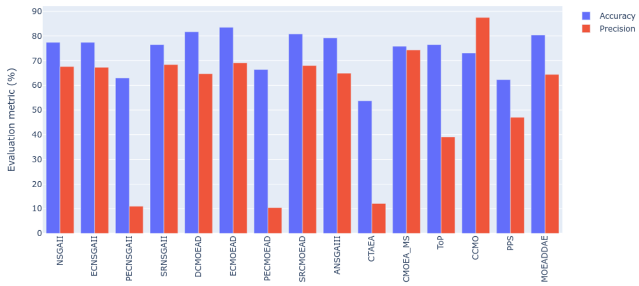



Fig. 8. SVM selection model accuracy and precision for each algorithm.

left that predicted to be solved mostly by CTAEA. However, this area was not easy to solve by almost all the algorithms, as observed in Figure 5. Our analysis of the instances in this area found that either there are no feasible solutions, or the HV of the set found by algorithms is approximately zero. Therefore, we conclude that the selection in this area is not accurate.

In addition, figure 8 describes the accuracy and precision of SVM model for each algorithm individually. The results validate the accuracy of the overall model. They illustrate all algorithms have high accuracy and precision, except PPS, CTAEA, and PECMOEAD. In addition, the metrics show that the quality of the algorithm selection model based on the selected features correlate with the clarity of the visualized footprints. For example, CTAEA and PECMOEAD have poor SVM metrics values, and they do not have clear footprints on the projected instance space. Nevertheless, our method does have limitations, as it relies on a large sample size, which may be more expensive than what one would be willing to invest in an application.

In a recent survey paper, Kerschke *et al.*, [52] stated that the cost of calculating features should not exceed the benefits of algorithm selection. Multiple works in single objective [53], multi-objective [12] and constrained [54] optimisation have shown that samples between $n \times 50$ to $n \times 200$ could be used. However, for CMOPs, determining a sample size that guarantees accuracy and reliability of the features will require further investigation.

## VI. Conclusion

We have presented a detailed instance space analysis of CMOPs. Our primary motivation was to systematically evaluate and characterize the conditions where a selected CMOEA was expected to perform well based on the landscape analysis features of CMOP instances. Firstly, we have identified CMOP features in terms of three landscapes: the multi-objective landscape; the violation landscape and the multi-objective-violation landscape. Secondly, we have collected a large volume of meta-data encapsulating multiple benchmark problem instances and algorithms (including alternative constraint handling techniques). Finally, footprints corresponding to regions of varying algorithm performance were identified. This visual representation provides useful insights, helping to explain CMOPs characteristics and the strengths and weakness of a particular algorithm. In addition, a SVM classifier was used to provide a preliminary mapping between the 'strength region' of an algorithm and particular problem characteristics.

Our results show that some CMOEAs, CCMO and PPS in particular, have distinct footprints. CCMO and PPS employ hyper constraint handling techniques, where they use two strategies in two populations/stages. CCMO can effectively converge on isolated optima, whereas PPS generates more diversity when there is a large optimal set. Significantly, the analysis shows that most CMOEAs fail to evolve high quality solutions when there is negative correlation between constraints and objectives. Moreover, CTAEA and other penalty-based algorithms have no clear area of strength, which indicates that the available benchmarks lack examples on which these algorithms would outperform.

It is widely acknowledged that any benchmark suite of problems should ideally test the efficacy of the optimizer. However, our analysis reveals a lack of diversity in the benchmark suites examined, with many instances sharing similar objective and/or constraint functions. Only a few instances provide a high proportion of constrained Pareto front to unconstrained front, and fewer instances have a highly negative correlation between constraints and objectives, despite the fact that those two characteristics are challenging for most algorithms. Our investigation of where real-world problems fall within the instance space revels that the current benchmark suites do not have enough characteristics to represent the real-world problems.

A wide range of existing and new Landscape Analysis features have been used in this work, however, they do not result in clear footprints for all algorithms. This, in turn, suggests that there is scope to further explore new features tailored specifically for CMOPs.

## References

[1] A. Kumar, G. Wu, M. Z. Ali, Q. Luo, R. Mallipeddi, P. N. Suganthan, and S. Das, "A Benchmark-Suite of real-World constrained multi-objective optimization problems and some baseline results," *Swarm Evol. Comput.*, vol. 67, p. 100961, 2021.

[2] Z. Fan, W. Li, X. Cai, H. Li, C. Wei, Q. Zhang, K. Deb, and E. Goodman, "Difficulty Adjustable and Scalable Constrained Multiobjective Test Problem Toolkit," *Evol. Comput.*, vol. 28, no. 3, pp. 339–378, 2020.

[3] F. Bogdan and V. Aljoša, "Multiobjective Optimization in the Presence of Constraints," 2021.

[4] Z. Fan, F. Yi, W. Li, J. Lu, X. Cai, and C. Wei, "A comparative study of constrained multi-objective evolutionary algorithms on constrained multi-objective optimization problems," in *2017 IEEE Congr. Evol. Comput. (CEC 2017) - Proc.*, 2017, pp. 209–216.

[5] O. Mersmann, B. Bischl, H. Trautmann, M. Preuss, C. Weihs, and G. Rudolph, "Exploratory landscape analysis," in *GECCO 2011 - Proc. 2011 Genet. Evol. Comput. Conf.*, 2011, pp. 829–836.

[6] K. Smith-Miles, D. Baatar, B. Wreford, and R. Lewis, "Towards objective measures of algorithm performance across instance space," *Comput. Oper. Res.*, vol. 45, pp. 12–24, 2014.

[7] E. Yap, M. A. Muñoz, K. Smith-Miles, and A. Liefooghe, "Instance Space Analysis of Combinatorial Multi-objective Optimization Problems," in *2020 IEEE Congr. Evol. Comput. (CEC 2020) - Proc.*, 2020.

[8] M. A. Muñoz and K. Smith-Miles, "Performance analysis of continuous black-box optimization algorithms via footprints in instance space," *Evol. Comput.*, vol. 25, pp. 529–554, 2017.

[9] K. M. Malan, "A survey of advances in landscape analysis for optimisation," *Algorithms*, vol. 14, no. 2, 2021.

[10] K. Smith-Miles and M. A. Muñoz, "Instance space analysis for algorithm testing: Methodology and software tools," 2021, http://dx.doi.org/10.13140/RG.2.2.33951.48805.

[11] P. Kerschke, H. Wang, M. Preuss, C. Grimme, A. H. Deutz, H. Trautmann, and M. T. Emmerich, "Search dynamics on multimodal multiobjective problems," *Evol. Comput.*, vol. 27, no. 4, pp. 577–609, 2018.

[12] A. Liefooghe, S. Verel, B. Lacroix, A. C. Zăvoianu, and J. McCall, "Landscape features and automated algorithm selection for multiobjective interpolated continuous optimisation problems," in *GECCO 2021 - Proc. 2021 Genet. Evol. Comput. Conf.*, 2021, pp. 421–429.

[13] K. M. Malan, J. F. Oberholzer, and A. P. Engelbrecht, "Characterising constrained continuous optimisation problems," in *2015 IEEE Congr. Evol. Comput. (CEC 2057) - Proc.*, 2015, pp. 1351–1358.

[14] S. Poursoltan and F. Neumann, *Ruggedness Quantifying for Constrained Continuous Fitness Landscapes*. New Delhi: Springer India, 2015, pp. 29–50.

[15] C. Picard and J. Schiffmann, "Realistic constrained multiobjective optimization benchmark problems from design," *IEEE Trans. Evol. Comput.*, vol. 25, no. 2, pp. 234–246, 2021.

[16] A. Vodopija, T. Tušar, and B. Filipič, "Characterization of constrained continuous multiobjective optimization problems: A feature space perspective," *Info. Sci.*, vol. 607, pp. 244–262, 2022.

[17] J. R. Rice, "The algorithm selection problem," *Adv. Comput.*, vol. 15, pp. 65–118, 1976.

[18] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, 1997.

[19] Z. Ma and Y. Wang, "Evolutionary constrained multiobjective optimization: Test suite construction and performance comparisons," *IEEE Trans. Evol. Comput.*, vol. 23, no. 6, pp. 972–986, 2019.

[20] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari, "Multiobjective optimization Test Instances for the CEC 2009 Special Session and Competition," The School of Computer Science and Electronic Engieering, University of Essex, Tech. Rep. CES-487, 2009.

[21] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part ii: Handling constraints and extending to an adaptive approach," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 602–622, 2014.

[22] K. Li, R. Chen, G. Fu, and X. Yao, "Two-archive evolutionary algorithm for constrained multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 2, pp. 303–315, 2019.

[23] Z. Fan, W. Li, X. Cai, H. Huang, Y. Fang, Y. You, J. Mo, C. Wei, and E. Goodman, "An improved epsilon constraint-handling method in MOEA/D for CMOPs with large infeasible regions," *Soft Comput.*, vol. 23, pp. 12 491–12 510, 2019.

[24] O. Cuate, L. Uribe, A. Lara, and O. Schütze, "A benchmark for equality constrained multi-objective optimization," *Swarm Evol. Comput.*, vol. 29, p. 105130, 2020.

[25] P. Stadler, "Fitness landscapes," in *Biological Evolution and Statistical Physics, Lecture Notes in Physics*. Berlin, Heidelberg: Springer-Verlag, 2002, pp. 183–204.

[26] S. Vérel, A. Liefooghe, L. Vermeulen-Jourdan, and C. Dhaenens, "Analyzing the effect of objective correlation on the efficient set of mnk-landscapes," in *LION*, 2011.

[27] J. Knowles and D. Corne, "Instance generators and test suites for the multiobjective quadratic assignment problem," in *Evolutionary Multi-Criterion Optimization*, C. M. Fonseca, P. J. Fleming, E. Zitzler, L. Thiele, and K. Deb, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 295–310.

[28] H. E. Aguirre and K. Tanaka, "Working principles, behavior, and performance of moeas on mnk-landscapes," *Eur. J. Oper. Res.*, vol. 181, pp. 1670–1690, 2007.

[29] A. Liefooghe, F. Daolio, S. Verel, B. Derbel, H. Aguirre, and K. Tanaka, "Landscape-Aware Performance Prediction for Evolutionary Multiobjective Optimization," *IEEE Trans. Evol. Comput.*, vol. 24, no. 6, pp. 1063–1077, 2019.

[30] T. Jones and S. Forrest, "Fitness Distance Correlation as a Measure of Problem Difficulty for Genetic Algorithms," in *the 6th Int. Conf. Genet. Algorithms*, 1995, pp. 184–192.

[31] A. Liefooghe, S. Vérel, H. E. Aguirre, and K. Tanaka, "What makes an instance difficult for black-box 0-1 evolutionary multiobjective optimizers?" in *Artif. Evol.*, 2013.

[32] E. Yap, M. A. Muñoz, and K. Smith-Miles, "Informing multi-objective optimisation benchmark construction through instance space analysis," *IEEE Trans. Evol. Comput.*, 2022.

[33] Y. Tian, Y. Zhang, Y. Su, X. Zhang, K. C. Tan, and Y. Jin, "Balancing Objective Optimization and Constraint Satisfaction in Constrained Evolutionary Multiobjective Optimization," *IEEE Trans. Cybern.*, pp. 1–14, 2021.

[34] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, 2002.

[35] Z. Fan, W. Li, X. Cai, H. Li, C. Wei, Q. Zhang, K. Deb, and E. D. Goodman, "Push and pull search for solving constrained multi-objective optimization problems," *Swarm Evol. Comput.*, vol. 44, pp. 665–679, 2019.

[36] A. Angantyr, J. Andersson, and J.-O. Aidanpaa, "Constrained optimization based on a multiobjective evolutionary algorithm," in *2003 IEEE Congr. Evol. Comput. (CEC 2003) - Proc.*, vol. 3, 2003, pp. 1560–1567 Vol.3.

[37] M. A. Jan, N. Tairan, and R. A. Khanum, "Threshold based dynamic and adaptive penalty functions for constrained multiobjective optimization," in *1st Int. Conf. Artif. Intell., Model. Simul. (AIMS 2013)*, Kota Kinabalu, Malaysia, 2013, pp. 49–54.

[38] Q. Zhu, Q. Zhang, and Q. Lin, "A constrained multiobjective evolutionary algorithm with detect-and-escape strategy," *IEEE Trans. Evol. Comput.*, vol. 24, no. 5, pp. 938–947, 2020.

[39] W. Ning, B. Guo, Y. Yan, X. Wu, J. Wu, and D. Zhao, "Constrained multi-objective optimization using constrained non-dominated sorting combined with an improved hybrid multi-objective evolutionary algorithm," *Eng. Optim.*, vol. 49, no. 10, pp. 1645–1664, 2017.

[40] T. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 4, no. 3, pp. 284–294, 2000.

[41] Y. Tian, T. Zhang, J. Xiao, X. Zhang, and Y. Jin, "A coevolutionary framework for constrained multiobjective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 25, no. 1, pp. 102–116, 2021.

[42] Z.-Z. Liu and Y. Wang, "Handling constrained multiobjective optimization problems with constraints in both the decision and objective spaces," *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 870–884, 2019.

[43] E. Zitzler, D. Brockhoff, and L. Thiele, "The Hypervolume Indicator Revisited: On the Design of Pareto-compliant Indicators Via Weighted Integration," in *Evolutionary Multi-Criterion Optimization. EMO 2007. Lect. Notes Comput. Sci.* Springer, Berlin, Heidelberg, 2007, vol. 4403.

[44] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima, "Modified distance calculation in generational distance and inverted generational distance," in *EMO*, A. Gaspar-Cunha, C. Henggeler Antunes, and C. C. Coello, Eds., 2015, pp. 110–125.

[45] C. Audet, S. L. Digabel, D. Cartier, J. Bigeon, and L. Salomon, "Performance indicators in multiobjective optimization," *Eur. J. Oper. Res.*, vol. 292, pp. 397–422, 2021.

[46] M. A. Muñoz, Y. Sun, M. Kirley, and S. K. Halgamuge, "Algorithm selection for black-box continuous optimization problems: A survey on methods and challenges," *Inf. Sci.*, vol. 317, pp. 224–245, 2015.

[47] D. A. van Veldhuizen, "Multiobjective evolutionary algorithms: classifications, analyses, and new innovations," Ph.D. dissertation, Air Force Institute of Technology, 1999.

[48] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, pp. 257–271, 1999.

[49] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "PlatEMO: A MATLAB platform for evolutionary multi-objective optimization," *IEEE Comput. Intell. Mag.*, vol. 12, no. 4, pp. 73–87, 2017.

[50] K. Smith-Miles, M. A. Munoz, and Neelofar, "Melbourne Algorithm Test Instance Library with Data Analytics (MATILDA)," 2020. [Online]. Available: https://matilda.unimelb.edu.au

[51] Y. Zhou, Y. Xiang, and X. He, "Constrained multiobjective optimization: Test problem construction and performance evaluations," *IEEE Trans. Evol. Comput.*, vol. 25, no. 1, pp. 172–186, 2021.

[52] P. Kerschke, H. H. Hoos, F. Neumann, and H. Trautmann, "Automated algorithm selection: Survey and perspectives," *Evol. Comput.*, vol. 27, pp. 3–45, 2019.

[53] P. Kerschke and H. Trautmann, "Automated algorithm selection on continuous black-box problems by combining exploratory landscape analysis and machine learning," *Evol. Comput.*, vol. 27, pp. 99–127, 2019.

[54] K. M. Malan, *Online Landscape Analysis for Guiding Constraint Handling in Particle Swarm Optimisation*. Singapore: Springer Singapore, 2021, pp. 101–118.