# Multimodal Scalarized Preferences in Multi-objective Optimization

Marlon Braun
Institute AIFB, KIT, Germany
marlon.braun@kit.edu

Lars Heling
Lars Heling
lars.heling@student.kit.edu

Pradyumn Shukla
Institute AIFB, KIT, Germany
pradyumn.shukla@kit.edu

Hartmut Schmeck
Institute AIFB, KIT, Germany
hartmut.schmeck@kit.edu

## ABSTRACT

Scalarization functions represent preferences in multi-objective optimization by mapping the vector of objectives to a single real value. Optimization techniques using scalarized preferences mainly focus on obtaining only a single global preference optimum. Instead, we propose considering all local and global scalarization optima on the global Pareto front. These points represent the best choice in their immediate neighborhood. Additionally, they are usually sufficiently far apart in the objective space to present themselves as true alternatives if the scalarization function cannot capture every detail of the decision maker's true preference. We propose an algorithmic framework for obtaining all scalarization optima of a multi-objective optimization problem. In said framework, an approximation of the global Pareto front is obtained, from which neighborhoods of local optima are identified. Local optimization algorithms are then applied to identify the optimum of every neighborhood. In this way, we have an *optima-based approximation* of the global Pareto front based on the underlying scalarization function. A computational study reveals that local optimization algorithms must be carefully configured for being able to find all optima.

## CCS CONCEPTS

**•Applied computing → Multi-criterion optimization and decision-making;**

## KEYWORDS

multi-objective optimization, evolutionary algorithm, scalarization, multimodal optimization, local optimum

**ACM Reference format:**
Marlon Braun, Lars Heling, Pradyumn Shukla, and Hartmut Schmeck. 2017. Multimodal Scalarized Preferences in Multi-objective Optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference 2017, Berlin, Germany, July 15–19, 2017 (GECCO '17),* 8 pages.
DOI: 10.1145/3071178.3079189


GECCO '17, July 15-19, 2017, Berlin, Germany

ACM ISBN 978-1-4503-4920-8/17/07…$15.00
http://dx.doi.org/10.1145/3071178.3079189

## 1 INTRODUCTION AND MOTIVATION

Multi-objective optimization (MOO) is the discipline of finding solutions to problems possessing several conflicting criteria. Such problems do not have a single solution that optimizes all objectives simultaneously. Instead, there exists a set of tradeoff solutions that can only be ameliorated in one criterion if another objective is deterioated at the same time. Such tradeoff solutions are denoted by Pareto optimal or efficient and their image in the objective space forms a so-called Pareto optimal front [7].

Choosing an appropriate Pareto optimal solution for implementation depends on the preferences of the decision maker (DM) [13]. *Scalarization* – aggregating the vector of objective values to a scalar – is a popular technique in MOO to identify preferred solutions. By picking the solution exhibiting the best scalarization value, the DM is not overburdened by being required to make a choice between multiple alternatives. The challenge lies only in finding a scalarization function (SF) that is a suitable representation of the DM's preference [18].

Scalarization, however, also suffers from several drawbacks. Since the search result is narrowed down to a single point, no additional information about the global Pareto front is generated. It is a well known fact in psychology and economics, however, that human preferences greatly depend on the available information [17]. Additionally, if an SF possesses multiple global scalarization optima (GSO), optimizing the SF usually yields only a single GSO, while other GSOs are neglected. If there exist multiple GSOs the DM often has different preferences regarding these points and bases his choice on the decision variables of the GSOs [21]. It is also conceivable that the SF is not an exact representation of the DM's true preference as any mathematical formulation is only a model and thereby an abstraction of the real-world [13]. This may result in counterintuitive or suboptimal results.

Tradeoffs, for example, are a popular method in MOO and multiple criteria decision analysis (MCDA) to describe preferences [13]. They allow a precise definition of minimum compensations a DM is willing to accept, if a deterioration in one or multiple objectives occur. Tradeoff-based preference models often identify boundary points on strictly concave Pareto fronts as optima [23]. The reason therefore is that any move towards the interior of the front results in trading in more units of some objectives compared to the number of units gained in return. Boundary points, however, do not provide any balance between the different objectives. Depending on the curvature of the front, there usually exist, however, interior points

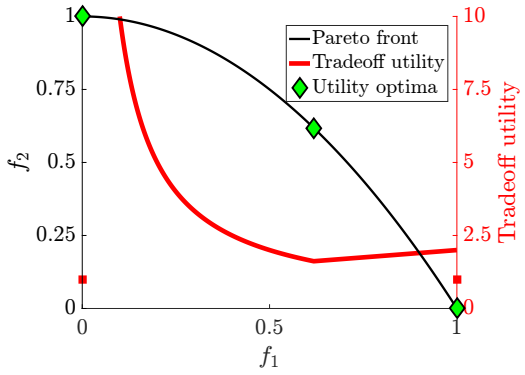


**Figure 1: Utility landscape of tradeoff utility [23] (see Equation 4), which is minimized, on the two objective minimization problem ZDT2 [26]. Global preference optima are located at the boundary of the front, which do not provide a balance between the two objectives (the utility values of global optima are marked by red squares in the plot). The utility of the interior LSO is close to the utility of the GSOs. A global search, however, neglects the LSO.**

that provide a balance between objectives and exhibit an acceptable tradeoff with respect to the global optima (see Figure 1).

Considering additional solutions is one possibility to address the issues outlined in the previous paragraphs. These solutions must be interesting to the DM by exhibiting a good scalarization value. On the other hand, they should be further apart in the objective space from each other and the GSO. Solutions that are close to the GSO possess similar objective values compared to the GSO. DMs, however, are usually indifferent between nondominated points that possess similar objective values. This is a known result in MCDA and, for example, explicitly considered in outranking techniques such as ELECTRE by using indifference thresholds [13]. Additionally, if the GSO does not meet the DM's expectations (see Figure 1), solutions close to the GSO are also likely to be unsatisfactory. Using different parametrizations of the SF, for example by adjusting weights, or subset selection mechanisms that target areas around the GSO usually results in obtaining solutions that have similar objective values compared to the GSO. We therefore propose considering all optima (local and global) of the scalarization function. These solutions have the best scalarization value in their immediate neighborhood on the global Pareto front and are usually further apart in the objective space (see Figures 1 to 3). The contributions of this work can be summarized in the following way:

- a definition of scalarization optima and how the focus on such points compares to previous approaches,
- proposition of a method for obtaining all scalarization optima of a multi-objective optimization problem (MOP) irrespective of the underlying SF,
- a quantitative analysis of the proposed method on a set of benchmark problems.

The subsequent work is structured as follows. The next section outlines basic concepts in MOO and a comparison between our and existing solving paradigms in MOO is drawn. Section 3 presents our proposed method for obtaining local scalarization optima (LSO). Subsequently, simulation results on benchmark problems are presented. The final section provides a summary and an outlook on future work.

## 2 EXISTING SOLVING PARADIGMS AND PREFERENCE OPTIMA

We consider the minimization of $m$ objectives $f := (f_1, \ldots, f_m)$ using an $n$-dimensional decision variable $\mathbf{x} = (x_1, \ldots, x_n)$. The set of feasible elements is denoted by $X$, where $X \subseteq \mathbb{R}^n$ and $\mathbb{R}^n$ is the search space. The image of $X$ under $f$ is denoted by $\mathcal{Y}$, where $\mathcal{Y}$ is a subset of the objective space $\mathbb{R}^m$. Solutions are addressed as elements of $\mathcal{Y}$.

*Definition 2.1 (Pareto dominance).* Let $\mathbf{u}, \mathbf{v} \in \mathcal{Y}$. $\mathbf{u}$ *Pareto dominates* $\mathbf{v}$, denoted by $\mathbf{u} \prec_p \mathbf{v}$, if $u_i \leq v_i$ for all $i \in \{1, \ldots, m\}$ and $u_j < v_j$ for at least one $j \in \{1, \ldots, m\}$.

*Definition 2.2 (Pareto optimality).* A point $\mathbf{u}^* \in \mathcal{Y}$ is called *Pareto optimal* if there exists no $\mathbf{u} \in \mathcal{Y}$ such that $\mathbf{u} \prec_p \mathbf{u}^*$.

The set of Pareto optimal solutions is denoted by $\mathcal{Y}_p$. The incorporation of preferences in MOO can be regarded from an algorithmic perspective and on the problem level itself. We first describe scalarization on the problem level.

In case the DM possesses no preference, any Pareto optimal solution constitutes an equally good choice. Preferences, on the other hand, induce a ranking on the Pareto optimal front by favoring some solutions to others. In this sense, preferences restrict the Pareto optimal set to a preferred subset. Any MOP can be reformulated such that its solution – the set of its nondominated solutions – is the preferred subset identified by a given preference structure.

An MOP is defined by its objective function $f$ and the order $\prec$ imposed on $\mathcal{Y}$, which coincides with the Pareto order in the no preference case. $f$ is further composed of the triple $(X, Y, F)$, where $X$ is the domain (search space), $Y$ the codomain (objective space) and $F$ the graph, which is a collection of ordered pairs that describe the functional relationship between $X$ and $Y$. Any reformulation must occur by changing either of these components.

An SF is a map for expressing preferences by mapping an element of the objective space to a real number. We assume that smaller scalarization values imply higher desirability to conform with the notion of minimizing objectives.

*Definition 2.3 (Scalarization function).* A scalarization function is a map $\Psi : \mathcal{Y} \to \mathbb{R}$.

The codomain of $\Psi$ is denoted by scalarization space. Scalarization is a transformation of the objective space. By concatenating the map from $\mathbb{R}^n$ to $\mathbb{R}^m$ and $\mathbb{R}^m$ to $\mathbb{R}$ we obtain a new optimization problem, whose set of Pareto optimal solutions $\mathcal{Y}_p$ consists of a single element that is preferred to all other solutions – the GSO.

From an algorithmic perspective, MOO techniques are usually characterized by at which stage of the optimization process preferences are incorporated. A priori methods include DM preferences before the MOP is solved. Interactive techniques require the DM to interact with the optimization process to guide the search towards desirable solutions. Approaches that first obtain a uniform approximation to the Pareto front from which the DM selects his solution to implement are called posteriori techniques [7].

Our algorithmic categorization of MOO techniques focuses on the optimization outcome – the set of elements obtained as result of an optimization procedure. This approach is an abstraction from the DM's preference structure and characterizes the set of alternatives from which the DM chooses when making his final decision. To our understanding, irrespective of the chosen preference model, paradigms for solving MOPs can be broadly divided into four categories.

**Closed form description:** Under certain smoothness assumptions [15], the Pareto front of a real-valued MOP is a manifold of at most $m-1$ dimensions, e.g. a curve for two or a surface for three objectives, and can thereby be represented by a parametric equation or implicit function. Finding such an analytic description is mostly intractable for real-world applications and thereby restricted to artificial benchmark problems [7]. Full knowledge of the Pareto front is usually required to make a detailed analysis of optimization algorithm performances [16].

**Finite set of points:** Classic MOO approaches approximate the Pareto front by a finite set of uniformly distributed points [7]. This paradigm bears the advantage that the DM obtains a deep insight into the composition of the front as it is the case for the closed form description. At the same time, however, the finite set of points approach is tractable for real-world optimization. Information about the composition of the Pareto front can further be used for preference elicitation [13].

**Subset approximation:** The Pareto front approximation can be restricted to specific regions of interest on the Pareto front that are more likely to contain interesting solutions. These regions are often referred to as *knee regions* in the literature [2, 9, 23]. Focusing on knee regions, computational resources are directed at more promising solutions and not wasted for exploring options that are never even considered for implementation. Depending on the chosen threshold for deeming a region interesting, however, the resulting subset may be too small to garner sufficient information to make a qualified choice among those preferred solutions.

**Global optima:** The search can also be focused on a single point [18]. Such points are commonly known as *knee points* [2, 23, 24]. This approach is advisable if the DM has a clear picture of the type of solution he is interested in. Focusing the search on a single solution, however, bears the risk of missing viable alternatives that only become apparent if the Pareto front is further explored. Scalarized preferences belong to this category, however cone-based approaches can also be used to identify a single global optimum [9].

Recent approaches have tried to blend the different paradigms. Braun et al. [5], for example, have developed an algorithm that uses scalarized preference information to bias the Pareto front approximation towards regions of interest while retaining a scope of the entire front. Shukla et al. [4] have proposed a method for dividing a fixed number of solutions between approximating the knee regions and the remaining front.

Our approach can be classified as a blend between subset approximation and global optima, where regions of interest of the Pareto front are represented by a single point. If we restrict the domain of the SF $\Psi$ to the set of Pareto optimal solutions $\mathcal{Y}_p$, we can identify LSOs and GSOs in the domain of $\Psi$ (see Definition 2.4 an Figure 2). An LSO possesses the best scalarization value in its neighborhood. It is therefore the most preferable point among all alternatives in its immediate vicinity and thereby presents itself as an interesting candidate for implementation to the DM. We propose an algorithm for finding all LSOs on the Pareto front and presenting these options to the DM. Such an algorithm obtains the GSO as well. Furthermore, in case there exist multiple global optima, all of these would be found by the algorithm. Due to these characteristics, the algorithm obtains a *scalarization-optima-based* approximation of the Pareto front. Such an approximation would be of particular importance in decision making scenarios. The DM only needs to state an SF that matches his preferences. The algorithm then presents him with him all LSOs and GSOs among which he can make his choice.

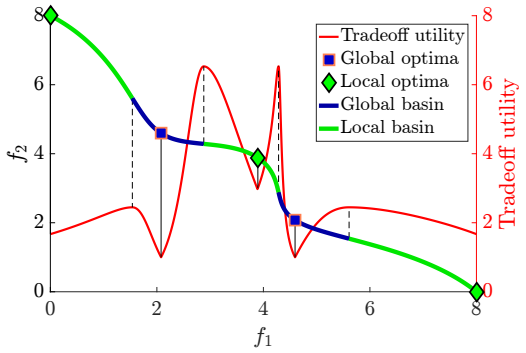


**Figure 2: Illustration of LSOs and their corresponding basins. Depiction of basins has been restricted to the Pareto front for clarity.**

*Definition 2.4 (Local scalarization optimum (LSO)).* Let $\Psi$ be an SF. A point $\mathbf{u}^L \in \mathcal{Y}_p$ is called a *local scalarization optimum (LSO)* of $\Psi$ if there exists a neighborhood $U(\mathbf{u}^L) \subseteq \mathcal{Y}_p$ of $\mathbf{u}^L$ such that $\forall \mathbf{u} \in U(\mathbf{u}^L)$ it holds that $\Psi(\mathbf{u}^L) \leq \Psi(\mathbf{u})$.

A concept that is of central importance in identifying LSOs is the notion of *basins of attraction* (see Definition 2.5). The basin of an LSO $\mathbf{u}^L$ consists of all points for which there exists a descending path with respect to the scalarization values in the objective space to $\mathbf{u}^L$. The scalarization values of the elements on this path monotonically decrease. All elements of the basin $B(\mathbf{u}^L)$ possess a larger scalarization value than $\mathbf{u}^L$. From a decision making perspective, $\mathbf{u}^L$ should be preferred to all members of $B(\mathbf{u}^L)$. A local search that is started in a given basin is expected to quickly converge to the LSO of said basin [21].

*Definition 2.5 (Basin c.f. [21, p. 5]).* The basin $B(\mathbf{u}^L)$ of an LSO $\mathbf{u}^L$ is the set of points $\mathbf{u} \in \mathcal{Y}$ for which there exists a continuous function $\alpha : [0,1] \to \mathbb{R}^m$ such that $\alpha(0) = \mathbf{u}$ and $\alpha(1) = \mathbf{u}^L$ and $\forall\, t_1 < t_2$ it holds that $\Psi(\alpha(t_1)) \geq \Psi(\alpha(t_2))$.

Finally, the definition of basins allows us to describe LSOs as preference on the problem level. In addition to changing the codomain, we change the order of the scalarization space. A solution $\mathbf{u}$ dominates a solution $\mathbf{v}$ if $\Psi(\mathbf{u}) < \Psi(\mathbf{v})$ and if additionally there exists an $\mathbf{u}^L$ such that $\mathbf{u}, \mathbf{v} \in B(\mathbf{u}^L)$. By requiring that $\mathbf{u}$ can only dominate $\mathbf{v}$ if they are located in the same basin, each local optimizer represents a non-dominated element of the scalarization space.

## 3 ALGORITHMIC APPROACH

The notion of LSOs is closely related to the field of multimodal optimization (MMO). MMO is the discipline of finding all optimizers (local or global) of a single-objective optimization problem (SOP) [21]. Our algorithmic approach is therefore inspired by established techniques that have been successfully applied in MMO. MMO methodology mainly consists of two steps: 1) identifying basins of attraction and 2) computation of local optimizers by executing local searches in the respective basins.

Basins are identified in MMO by sampling points in the decision space. These points are divided into subsets by applying a clustering algorithm. Ideally, all points of a single cluster lie in the same basin of attraction. Thereby, the cluster member exhibiting the smallest objective value (scalarization value in MOO) is an estimate of a local optimizer. Sufficient exploration of the search space is thereby imperative for identifying all basins of attraction. This is achieved in MMO by resampling in the decision space multiple times [21]. In MOO, we can exploit the fact that scalarization optima are located only on the Pareto front. In our approach, we therefore first obtain an approximation of the Pareto front.

We adopt nearest better clustering (NBC) [21, Chp. 4] from MMO to identify basins of attraction, however any other clustering method could be used instead. NBC creates a spanning tree from the Pareto front approximation obtained in the previous step by connecting each solution to its closest neighbor in the objective space that has a smaller scalarization value. Edge weights correspond to the Euclidean distances in the objective space between individual solutions. Next, two rules are applied to cut edges in the spanning tree – see Lines 7 and 10 in Algorithm 1. The first rule cuts edges of solutions that are presumably too far apart in the objective space for being located in the same basin. The second rule cuts the outgoing edge of solutions that possess multiple neighbors with a larger scalarization value. $\phi$ and $b$ (see Algorithm 1) are two parameters that control the number of clusters generated. Decreasing both parameters results in more edges being cut and thereby more clusters being generated (cf. [21, Chp. 4])

Cutting edges results in the spanning tree being divided into multiple disconnected subgraphs. The solutions representing each subgraph are now used as initial set of solutions in a single-objective local search, where the objective function is the SF $\Psi$ applied to the underlying MOP that is solved. The entire procedure is summarized in Algorithm 1.

In case the Pareto front is sufficiently approximated, NBC is expected to correctly identify all basins of attraction. It is imperative that the local search does not escape its dedicated basin and converges to the LSO of another basin. This is a general problem that is also recognized in MMO [21]. The metaheuristics applied in the local search need to be configured to narrowly search their dedicated basin. For this reason, we investigate multiple algorithms for their suitability with our approach in the computational study in the next section.

Decomposition-based algorithms such as MOEA/D [25] should also be discussed in light of the proposed algorithmic framework. These algorithms obtain GSOs to the same aggregated problem instance, using different parametrizations of the chosen scalarization function. In choosing different parametrizations, e.g. weights for the Chebyshev method in MOEA/D, these methods effectively obtain solutions for different DM preferences. LSOs could be obtained by choosing appropriate weights, however since the location of these LSO is not known, there is no straightforward way for setting these weights correctly.

**Algorithm 1:** LSOs Finding Procedure

**Input:** MOP $\mathcal{M}$ and SF $\Psi$
1 Generate Pareto front approximation $P$ of $\mathcal{M}$
2 Compute $\forall \mathbf{p} \in P : \Psi(\mathbf{p})$
3 Create empty graph $G = (V, E)$ `// spanning tree`
4 **foreach** $\mathbf{p} \in P$ **do**
5 $\quad \mathbf{q}^* := \arg\min_{\mathbf{q} \in P : \Psi(\mathbf{q}) > \Psi(\mathbf{p})} \text{distance}(\mathbf{q}, \mathbf{p})$
6 $\quad$ Add edge $(\mathbf{p}, \mathbf{q}^*)$ to $G$
7 **Rule 1:** cut edges of length $> \phi \cdot$mean(lengths of all edges in $G$)
8 **Rule 2: forall** vertices with at least 3 incoming and 1 outgoing edges **do**
9 $\quad$ **if** length(outgoing edge)/(median(length(incoming edges)) $> b$ **then**
10 $\quad\quad$ Cut outgoing edge
11 $U := \emptyset$ `// set of scalarization optima`
12 **forall** disconnected subgraphs $C$ of $G$ **do**
13 $\quad$ Use $C$ as start population in a local search to find an approximation $\mathbf{u}^L$ to an LSO of $\Psi$ on $\mathcal{M}$. Add $\mathbf{u}^L$ to $U$.
14 **return** $U$

## 4 COMPUTATIONAL STUDY

### 4.1 Simulation Setup

For benchmarking our algorithmic approach, we have chosen four SFs that all use different concepts of measuring preferences. They especially differ in their mathematical complexity to reflect a varying degree of difficulty. The sum of objectives is a special case of the weighted sum method, where all objectives are equally important [18]. The weighted sum method is the centerpiece of many multi-attribute utility and value theory (MAUT/MAVT) techniques in MCDA to identify preferred solutions. The Nash bargaining solution [19] is a popular SF in economics and engineering applications [18]. Angle utility [6] is a recent technique that utilizes the Pareto front's geometry. Preferred solutions are located in convex bulges, however interior points are preferred on strictly concave fronts. Tradeoff utility [23], also known as proper utility, identifies preferred solution by assessing how many units have to be given up in one objective by obtaining a certain improvement in another objective. The popular Chebyshev method [18] was omitted, because it only exhibits more than one LSO on disconnected Pareto fronts of problems in this study.

**Sum of objectives**

$$\Psi^S(\mathbf{u}) = \sum_{i=1}^{m} u_i. \tag{1}$$

**Nash bargaining solution**

$$\Psi^N(\mathbf{u}) = \prod_{i=1}^{m} \left(u_i^{ndr} - u_i\right), \tag{2}$$

where $\mathbf{u}^{ndr}$ is the Nadir point.

**Angle utility**

$$\Psi^A(\mathbf{u}) = \max_{k\in\{1,\ldots,m\}} \arctan\left(\frac{\sqrt{\sum_{i=1,\, i\neq k}^{m}\left(u_i - e_i^k\right)^2}}{|u_k - e_k^k|}\right), \tag{3}$$

where $\mathbf{e}^k$ is the extreme point of objective $k$ with $\mathbf{e}^k = \arg\min_{\mathbf{v}\in\mathcal{Y}} \max_{i\in\{1,\ldots,m\}}(w_i^k v_i)$, where $\mathbf{w}$ is a vector of length $m$ where the $k$-th entry equals to 1 and all other entries equal to $\varepsilon$ with $\varepsilon > 0$ but close to 0.

**Tradeoff utility**

$$\Psi^T(\mathbf{u}) = \max_{\mathbf{v}\in\mathcal{Y}} \frac{\max_{i\in\{1,\ldots,m\}} u_i - v_i}{\max_{i\in\{1,\ldots,m\}} v_i - u_i} \tag{4}$$

The sum of objectives can be computed without any further knowledge of the Pareto front. The other three functions require reference points. The Nadir point and extreme points can be determined from a Pareto front approximation that covers the boundary of the front. Tradeoff utility requires a uniform sample of the Pareto front as reference set for computing the maximum tradeoff a solution exhibits. All this information must be obtained before any scalarization values can be computed. Angle and tradeoff utility utilize the maximum function, which can render the utility landscape of the Pareto front non-smooth as depicted in Figure 3. This might deter local searches from correctly identifying LSOs.

The benchmark problems chosen for this study exhibit multiple LSOs. The DEB2DK, DEB3DK and DO2DK problems [2] possess a parameter $k$ for controlling the number of convex bulges on the Pareto front. These bulges introduce hills and valleys in the utility landscapes making them ideal candidates for this study. DO2DK additionally possesses a parameter $s$ that adds skewing to the front. WFG1's [16] Pareto front also exhibits multiple bulges. The Pareto front of DTLZ7 [11], WFG2 [16] and ZDT3 [26], on the other hand, consist of multiple disconnected patches. Consequently, each patch contains at least one LSO. The number of LSOs of each problem per SF is listed in Table 1. Minima were computed using either mathematical programming techniques or by uniformly sampling points on the Pareto front and identifying those solutions having the smallest scalarization value in their immediate neighborhood.

NSGA-II [10] was chosen to obtain the Pareto front approximation in the first step of Algorithm 1, because it has been successfully applied to solve artificial benchmark and real-world problems. However, any algorithm capable of generating a uniform approximation to the Pareto front would constitute a viable alternative. NSGA-II was configured according to the parameters used in [10]. A population size of 200 was chosen for all problems. 25 000 functions evaluations were used to approximate the Pareto fronts of all problems besides the WFG instances, for which 50 000 function evaluations were used. Since many basins in our study are very small and thus easily missable, we set parameter $\phi$ to 1.1 in cutting rule 1 instead of 2 as it is suggested in [21, p. 91]. Parameter $b$ was set as proposed in [21, Eq. 4.18].

**Table 1: Number of LSOs of each test problem for every SF and the number of objectives $m$ of each problem.**

| | Sum | Nash | Angle | Tradeoff | $m$ |
|---|---|---|---|---|---|
| DEB2DK $k1$ | 3 | 1 | 1 | 3 | 2 |
| DEB2DK $k2$ | 4 | 2 | 2 | 5 | 2 |
| DEB2DK $k3$ | 5 | 3 | 3 | 7 | 2 |
| DEB2DK $k4$ | 6 | 4 | 4 | 9 | 2 |
| DEB3DK $k1$ | 5 | 1 | 1 | 6 | 3 |
| DEB3DK $k2$ | 7 | 4 | 4 | 17 | 3 |
| DO2DK $k1s1$ | 1 | 1 | 1 | 2 | 2 |
| DO2DK $k2s1$ | 2 | 1 | 1 | 3 | 2 |
| DO2DK $k3s1$ | 2 | 1 | 1 | 5 | 2 |
| DO2DK $k4s1$ | 3 | 2 | 1 | 5 | 2 |
| DTLZ7 | 4 | 4 | 4 | 7 | 3 |
| WFG1 | 5 | 3 | 5 | 10 | 2 |
| WFG2 | 6 | 6 | 6 | 6 | 2 |
| ZDT3 | 5 | 5 | 5 | 6 | 2 |

The performance of Algorithm 1 is not only influenced by the problem to solve, but also by the chosen SF. We therefore opted to focus our computational study on comparing different local search algorithms. Since our algorithmic approach is intended to not make any assumptions about the mathematical structure of the SF, black box optimization techniques present themselves as suitable candidates for benchmarking.

We have chosen four metaheuristics for our study: CMA-ES (CMA) [14] uses a covariance matrix to draw decision variable values from a normal distribution. The matrix is updated in each iteration to steer the search towards the solutions that have achieved the smallest scalarization value in the previous iteration. CMA-ES is credited as one of best performing black box optimization techniques [21]. An elitist genetic algorithm (GA) using simulated binary crossover (SBX) [1] and polynomial mutation [8] was chosen as well. Furthermore, a particle-swarm optimization algorithm (PSA) was selected for the study. We complemented the study by a hill climber algorithm (HC) that used polynomial mutation as search operator and an archive to store the best solution yet found. Each local search algorithm was run for 5000 (10 000 for WFG1 and WFG2) function evaluations on each cluster obtained by NBC. Each configuration was run 100 times on every test problem. A baseline (B) serves as benchmark for comparing results after clustering but before local searches are executed.

Parametrization plays an important role in improving the performance of local search. Preliminary tests were conducted to identify the best parametrization for each algorithm. A detailed analysis is omitted to conserve space. Mutation probabilities were all set to one divided by the number of decision variables. To focus the search narrowly on the dedicated basin, distribution indices of mutation and crossover operators were set to high values to generate offspring solutions that are located closer to their parents. After analyzing our preliminary results, the following values were chosen: 140 (HC), 140 (GA – both mutation and crossover). Our code is implemented in the jMetal framework version 4.5 [12] and available on *Sourceforge*[1]. jMetal implementations of CMA-ES, the GA and the PSA were used in this study. All distances were computed using the Euclidean norm. Objective values were normalized based on

[1] http://sourceforge.net/projects/jmetalbymarlonso/

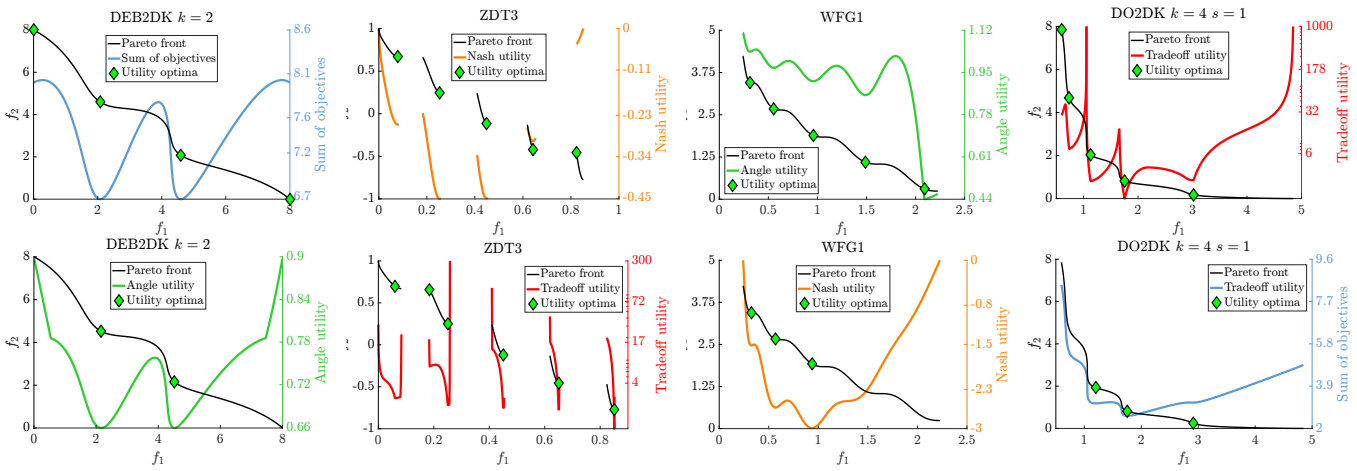


**Figure 3: Example of utility landscapes of problems used in this study. Steep valleys might lead local optimization algorithms to accidentally jump to different basins.**

the size of the Pareto front to guarantee the comparability of the results.

## 4.2 Simulation Results and Discussion

Table 2 shows the results for the peak distance (PD) indicator [21, p. 122]. PD computes the mean of the distances between every LSO and its closest approximation in the objective space that was obtained in a single run of Algorithn 1. Thereby, PD measures how well individual minima are approximated on average. The HC achieves the overall best performance, closely followed by the GA. CMA-ES and the PSA show good results on some problem instances but exhibit at the same time bad performance for others, sometimes even performing worse than the baseline.

Although there exist performance differences between problem classes and SFs, there are no clear trends apparent. Even though tradeoff utility introduces non-smoothness and high peaks in the utility landscape, possibly deterring local search, results do not deteriorate overall in comparison to the other SFs. For example, while the performance of the PSA is worse for tradeoff utility with respect to the sum of objectives, PD improves for the HC on some instances. Performances on three objective problems are often only slightly worse with respect to two objective problems. These observations suggest that Algorithm 1 delivers reliable results irrespective of the chosen SF.

Results for the peak ratio (PR) indicator [21, p. 122] are displayed in Table 3. PR counts for each run the number of minima found by the optimization algorithm and divides that number by the number of overall minima of the given problem. We deem a minimum as found if the distance to its closest approximation in the objective space is less than $10^{-2}$. A value of 1 implies that all minima were identified and a value of zero 0 that no minimum was identified with a precision smaller than $10^{-2}$.

Since PD and PR are closely related, performance rankings for both indicators are similar. The GA and the HC are again the two top performing algorithms. The PSA, however, can be identified as third best performing algorithm in contrast to the results for the PD. The results achieved by CMA-ES, on the other hand, can be considered inferior with respect to the other three algorithms.

The performance differences between PR and PD can be attributed to the PD being stronger influenced by outliers. The PD results for the PSA suggest that it entirely misses some minima. The minima the algorithm finds, however, are well approximated, which is supported by high peak ratios. CMA-ES, on the other hand, appears to correctly identify all minima on nearly every problem instance, but the approximations obtained by the algorithm are not close enough to fall below the $10^{-2}$ threshold.

It may look surprising that a simple HC outperforms more sophisticated optimization techniques such as CMA-ES, GAs, or particle swarm optimization (PSO). The challenge of obtaining all LSOs might offer an explanation for this observation. Basin identification is started at or near the Pareto front. If the Pareto front approximation is nearly uniform, then it is likely that the solution having the best scalarization value of a given cluster is already a good estimate of the corresponding minimum. Search methods should therefore benefit from a stronger focus on exploitation instead of exploration. The HC does exactly that by generating only a single new solution in the vicinity of the old solution in each iteration.

The PD and PR results of the GA and the HC differ only marginally, which can be explained by both algorithms employing a similar search strategy. The GA employs a larger population and, in addition, a crossover operator. The main force that drives the search in PSO is the speed of each particle that determines its position in the next iteration, and which is influenced by the globally and locally best solution. PSAs are known to make large jumps in the search space if particle speed is not controlled properly [20]. We could therefore observe that, often, individual swarms left their original basins if single particles got attracted by other basins that feature stronger minima. Since basins are often quite narrow, individual particles may easily escape their basin. As can be observed in Figure 3, a small move in the search space might translate to a large move in the scalarization space. This might as well lead to individual particles escaping their dedicated basin. This observation also

**Table 2: PD. Median results across 100 runs. Best performances are highlighted in dark gray, second-best performances in light gray. Results are grouped by scalarization function.**

| | Sum | | | | | Nash | | | | | Angle | | | | | Tradeoff | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CMA | GA | HC | PSA | B | CMA | GA | HC | PSA | B | CMA | GA | HC | PSA | B | CMA | GA | HC | PSA | B |
| DEB2DK $k1$ | 4.8e−1 | 1.9e−4 | 1.1e−5 | 1.9e−5 | 6.0e−3 | 3.6e−3 | 4.4e−4 | 8.9e−5 | 1.4e−4 | 5.7e−3 | 2.2e−3 | 9.8e−4 | 8.7e−4 | 1.0e−3 | 4.2e−3 | 2.5e−1 | 4.3e−4 | 2.8e−4 | 2.4e−4 | 6.4e−3 |
| DEB2DK $k2$ | 4.6e−1 | 1.3e−1 | 1.3e−1 | 1.3e−1 | 1.3e−1 | 2.0e−1 | 7.7e−4 | 2.6e−4 | 2.0e−1 | 7.6e−3 | 2.1e−1 | 8.3e−4 | 4.5e−4 | 2.1e−1 | 7.3e−3 | 1.9e−1 | 5.7e−4 | 5.2e−4 | 4.9e−2 | 6.9e−3 |
| DEB2DK $k3$ | 4.3e−1 | 1.5e−1 | 1.5e−1 | 4.3e−1 | 1.5e−1 | 2.1e−1 | 7.9e−4 | 3.7e−4 | 2.1e−1 | 7.4e−3 | 2.3e−1 | 1.5e−3 | 1.1e−3 | 2.3e−1 | 7.1e−3 | 2.1e−1 | 4.7e−2 | 4.7e−2 | 1.6e−1 | 5.0e−2 |
| DEB2DK $k4$ | 4.2e−1 | 9.4e−2 | 9.4e−2 | 3.0e−1 | 9.7e−2 | 2.4e−1 | 1.3e−1 | 1.3e−1 | 2.4e−1 | 1.4e−1 | 2.7e−1 | 7.7e−4 | 3.6e−4 | 2.6e−1 | 7.2e−3 | 1.6e−1 | 4.6e−2 | 4.6e−2 | 1.2e−1 | 4.9e−2 |
| DEB3DK $k1$ | 4.9e−1 | 2.5e−1 | 2.5e−1 | 2.5e−1 | 3.0e−1 | 2.5e−2 | 3.8e−3 | 2.2e−3 | 4.8e−3 | 8.3e−2 | 2.2e−2 | 1.3e−2 | 1.0e−2 | 1.1e−2 | 7.8e−2 | 3.3e−1 | 2.7e−1 | 2.8e−1 | 2.7e−1 | 2.2e−1 |
| DEB3DK $k2$ | 5.8e−1 | 2.4e−1 | 2.5e−1 | 2.5e−1 | 3.0e−1 | 2.7e−2 | 4.4e−3 | 3.3e−3 | 5.4e−3 | 7.8e−2 | 2.6e−2 | 1.3e−2 | 9.4e−3 | 1.4e−2 | 8.0e−2 | 3.8e−1 | 2.6e−1 | 2.7e−1 | 2.7e−1 | 2.1e−1 |
| DO2DK $k1s1$ | 4.1e−3 | 7.2e−4 | 7.2e−5 | 5.5e−5 | 3.6e−3 | 4.9e−3 | 4.2e−4 | 1.7e−4 | 2.0e−4 | 3.1e−3 | 2.9e−3 | 7.5e−4 | 1.3e−3 | 1.3e−3 | 3.1e−3 | 2.5e−3 | 1.4e−3 | 2.7e−3 | 2.1e−3 | 4.8e−3 |
| DO2DK $k2s1$ | 1.5e−1 | 2.4e−3 | 5.3e−4 | 3.0e−1 | 1.3e−2 | 1.7e−3 | 5.3e−4 | 2.5e−4 | 1.7e−4 | 5.7e−3 | 1.5e−3 | 3.0e−4 | 3.4e−4 | 5.6e−4 | 6.0e−3 | 1.8e−1 | 2.0e−3 | 3.5e−3 | 3.7e−1 | 5.5e−3 |
| DO2DK $k3s1$ | 1.5e−1 | 7.6e−4 | 9.0e−5 | 2.9e−1 | 5.2e−3 | 2.6e−3 | 6.3e−4 | 1.5e−4 | 1.7e−4 | 4.2e−3 | 1.4e−3 | 4.0e−4 | 7.9e−4 | 8.0e−4 | 5.9e−3 | 1.1e−1 | 3.0e−2 | 4.0e−2 | 3.7e−1 | 1.9e−2 |
| DO2DK $k4s1$ | 1.7e−1 | 2.5e−1 | 2.5e−1 | 3.5e−1 | 1.6e−1 | 9.9e−2 | 6.9e−4 | 1.3e−4 | 1.8e−1 | 4.9e−3 | 2.5e−3 | 3.1e−3 | 3.2e−3 | 4.3e−3 | 3.4e−3 | 2.0e−1 | 1.7e−3 | 1.9e−3 | 4.0e−1 | 5.8e−3 |
| DTLZ7 | 4.1e−1 | 4.1e−1 | 4.1e−1 | 4.1e−1 | 4.2e−1 | 4.8e−1 | 1.2e−1 | 3.7e−2 | 4.7e−1 | 1.7e−1 | 2.3e−1 | 2.3e−1 | 2.3e−1 | 2.4e−1 | 1.6e−1 | 7.3e−2 | 8.4e−2 | 8.4e−2 | 2.7e−1 | 1.1e−1 |
| WFG1 | 5.5e−1 | 8.9e−1 | 9.1e−1 | 3.0e−1 | 2.0e−1 | 4.5e−1 | 4.5e−1 | 3.8e−1 | 3.1e−1 | 2.3e−1 | 5.5e−1 | 8.7e−1 | 8.9e−1 | 3.0e−1 | 2.1e−1 | 2.2e−1 | 3.4e−1 | 2.9e−1 | 2.7e−1 | 1.9e−1 |
| WFG2 | 2.6e−1 | 8.4e−2 | 3.4e−2 | 8.0e−1 | 3.5e−2 | 1.6e−1 | 2.0e−1 | 1.5e−1 | 3.9e−1 | 1.1e−1 | 7.4e−1 | 7.5e−2 | 3.4e−2 | 3.9e−1 | 3.4e−2 | 4.7e−3 | 2.2e−3 | 2.1e−3 | 1.6e−1 | 3.7e−3 |
| ZDT3 | 2.5e−1 | 4.3e−4 | 9.7e−5 | 6.0e−1 | 4.6e−3 | 3.9e−1 | 4.2e−1 | 4.2e−1 | 4.3e−1 | 1.1e−1 | 2.5e−1 | 2.3e−3 | 2.1e−3 | 3.7e−1 | 5.9e−3 | 3.7e−3 | 5.2e−4 | 4.9e−4 | 6.5e−2 | 5.0e−3 |

**Table 3: PR. Median results across 100 runs. Best performances are highlighted in dark gray. Results are grouped by scalarization function. The last two columns show median hypervolume (HV) and IGD of the Pareto front approximations computed in step 1 of Algorithm 1 across all runs.**

| | Sum | | | | | Nash | | | | | Angle | | | | | Tradeoff | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CMA | GA | HC | PSA | B | CMA | GA | HC | PSA | B | CMA | GA | HC | PSA | B | CMA | GA | HC | PSA | B | HV | IGD |
| DEB2DK $k1$ | 0.33 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 0.67 | 1.00 | 1.00 | 1.00 | 1.00 | 6.0e−1 | 4.9e−5 |
| DEB2DK $k2$ | 0.25 | 0.75 | 0.75 | 0.75 | 0.50 | 0.50 | 1.00 | 1.00 | 0.50 | 1.00 | 0.50 | 1.00 | 1.00 | 0.50 | 1.00 | 0.40 | 1.00 | 1.00 | 0.80 | 0.80 | 5.6e−1 | 5.2e−5 |
| DEB2DK $k3$ | 0.20 | 0.60 | 0.60 | 0.20 | 0.40 | 0.33 | 1.00 | 1.00 | 0.33 | 0.67 | 0.33 | 1.00 | 1.00 | 0.33 | 0.67 | 0.43 | 0.71 | 0.71 | 0.43 | 0.71 | 5.5e−1 | 5.2e−5 |
| DEB2DK $k4$ | 0.17 | 0.67 | 0.67 | 0.33 | 0.50 | 0.25 | 0.50 | 0.50 | 0.25 | 0.50 | 0.25 | 1.00 | 1.00 | 0.25 | 0.75 | 0.44 | 0.67 | 0.67 | 0.44 | 0.56 | 5.4e−1 | 5.2e−5 |
| DEB3DK $k1$ | 0.00 | 0.60 | 0.60 | 0.60 | 0.00 | 0.00 | 1.00 | 1.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.14 | 0.14 | 0.00 | 6.8e−1 | 5.8e−4 |
| DEB3DK $k2$ | 0.00 | 0.60 | 0.60 | 0.60 | 0.00 | 0.00 | 1.00 | 1.00 | 1.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.14 | 0.14 | 0.00 | 6.8e−1 | 5.7e−4 |
| DO2DK $k1s1$ | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 9.2e−1 | 3.9e−5 |
| DO2DK $k2s1$ | 0.50 | 1.00 | 1.00 | 0.50 | 0.50 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 0.50 | 1.00 | 1.00 | 0.50 | 1.00 | 8.9e−1 | 4.0e−5 |
| DO2DK $k3s1$ | 0.50 | 1.00 | 1.00 | 0.50 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 0.50 | 0.50 | 0.25 | 0.25 | 0.80 | 8.8e−1 | 4.5e−5 |
| DO2DK $k4s1$ | 0.33 | 0.33 | 0.33 | 0.33 | 0.33 | 0.50 | 1.00 | 1.00 | 0.50 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 0.50 | 1.00 | 1.00 | 0.25 | 1.00 | 8.7e−1 | 4.5e−5 |
| DTLZ7 | 0.25 | 0.50 | 0.50 | 0.25 | 0.00 | 0.25 | 0.25 | 0.00 | 0.25 | 0.00 | 0.25 | 0.25 | 0.00 | 0.25 | 0.00 | 0.00 | 0.14 | 0.14 | 0.14 | 0.00 | 2.8e−1 | 1.7e−3 |
| WFG1 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 3.8e−1 | 1.8e−3 |
| WFG2 | 0.33 | 0.50 | 0.67 | 0.00 | 0.67 | 0.50 | 0.33 | 0.50 | 0.17 | 0.67 | 0.00 | 0.58 | 0.67 | 0.00 | 0.83 | 0.83 | 0.83 | 0.83 | 0.00 | 0.83 | 5.6e−1 | 2.2e−5 |
| ZDT3 | 0.20 | 1.00 | 1.00 | 0.20 | 1.00 | 0.10 | 0.20 | 0.20 | 0.00 | 0.20 | 0.40 | 1.00 | 1.00 | 0.20 | 1.00 | 1.00 | 1.00 | 1.00 | 0.60 | 1.00 | 5.1e−1 | 6.4e−5 |

yields an explanation to why the baseline sometimes outperforms the local searches.

We have noticed that the covariance matrix of CMA-ES often became close to singular even before the budget of function evaluations had been fully exhausted. In this case, the covariance matrix can no longer be reasonably updated to steer the search in the next iteration and the algorithm terminates. This is mainly a numerical problem that can be attributed to the mathematical structure of the test problems and the limited numerical precision of the CMA-ES implementation in jMetal. The Pareto fronts of nearly all artificial benchmark problems in MOO are constructed in a similar way. Out of the $n$ decision variables of the problem, only a small number $k$ with usually $k = m - 1$ influence the position of a solution on the front. The remaining $n - k$ variables only increase all objective values at the same time and thereby move the solution away from the front. Those $n - k$ variables, consequently, need to attain a specific value for a solution to be Pareto optimal. In case the Pareto front approximation is close to the true front, the solutions of a cluster exhibit only limited variability in a few decision variables. This may result in the initial covariance matrix being singular to working precision.

Finally, we take a look at false positives (FP). A false positive occurs if an algorithm identifies a point as LSO that is too far away from a real local optimum to qualify as such. We chose a threshold of 0.2 for deeming a point too far away to qualify as approximation to a local minimum. This threshold was deliberately chosen to be much larger than the detection threshold for the PR to eliminate minima that have been correctly identified but not sufficiently well approximated. FPs are undesirable, since they are potentially misleading for a DM. We would expect to find other solutions in the immediate vicinity of an FP that exhibit a smaller scalarization value.

Results for FPs are displayed in Table 4. We observe that the results are similar across all algorithms. FPs only occur on a few benchmark problems. They increasingly appear for the more difficult SFs. Their utility landscapes feature an increasing multimodality including shallow basins, in which local search algorithms can get trapped. The results in Table 4 also support the conclusion that large PDs can be explained by local search algorithms escaping their dedicated cluster and moving to a stronger minimum instead. For the WFG1 problem, on the other hand, we could observe that the Pareto front approximation was often far way from the true front. Therefore, local search mostly identified the GSO. This observation supports our approach that basin identification should be done using a sufficient approximation of the Pareto front.

**Table 4: FP. Median results across 100 runs. Algorithms are abbreviated in the following way: CMA-ES (C), GA (G), HC (H) and PSA (P).**

| | Sum | | | | Nash | | | | Angle | | | | Tradeoff | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | C | G | H | P | C | G | H | P | C | G | H | P | C | G | H | P |
| DEB2DK $k1$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| DEB2DK $k2$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| DEB2DK $k3$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| DEB2DK $k4$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| DEB3DK $k1$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 2 | 1 | 0 |
| DEB3DK $k2$ | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 3 | 2 | 1 | 0 |
| DO2DK $k1s1$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| DO2DK $k2s1$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 |
| DO2DK $k3s1$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| DO2DK $k4s1$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| DTLZ7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 4 | 4 | 3 | 0 | 0 | 0 | 0 |
| WFG1 | 4 | 4 | 4 | 3 | 3 | 3 | 3 | 3 | 4 | 4 | 4 | 4 | 4 | 4 | 1 | 6 |
| WFG2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| ZDT3 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

## 5 SUMMARY AND OUTLOOK

We have proposed a new optimization paradigm in MOO in searching for points that are locally preferred on the Pareto front. Locally preferred points are interesting candidates for implementation, since they are the best option in their immediate neighborhood and further apart in the objective space to constitute true alternatives. An algorithmic framework has been presented for obtaining all LSOs of an MOP. A computational study has been performed to analyze the challenges that occur in finding LSOs. It has been concluded that the proposed framework works, however, great care needs to be given to selecting and parametrizing local search algorithms to guarantee fast convergence and prevent the escape of dedicated search basins.

There exist many research questions that should be addressed in future work. A detailed study for assessing the effect of parameter configuration on local search algorithms performance is imperative in evaluating and improving performances. At the same time, our approach should be tested on real-world applications such as [3] to also avoid the risk of overfitting algorithm configurations to artificial benchmark problems. Many-objective optimization problems are also a promising direction for our approach, because the Pareto fronts of these problems are difficult to visualize. Presenting the DM with a candidate set of LSOs could greatly simplify the selection process. Furthermore, our approach can be combined with other dominance concepts [23] or preference structures such as variable orderings [22] to obtain solutions that are locally non-dominated.

## REFERENCES


[1] Ram Bhusan Agrawal and Kalyanmoy Deb. 1994. Simulated binary crossover for continuous search space. *Complex Systems* 9, 3 (1994), 1–15.

[2] Jürgen Branke, Kalyanmoy Deb, Henning Dierolf, and Matthias Osswald. 2004. Finding Knees in Multi-objective Optimization. In *Parallel Problem Solving from Nature - PPSN VIII*. Springer, 722–731.

[3] Marlon Braun, Thomas Dengiz, Ingo Mauser, and Hartmut Schmeck. 2016. *Applications of Evolutionary Computation: 19th European Conference, EvoApplications 2016, Proceedings, Part I.* Springer, Chapter Comparison of Multi-objective Evolutionary Optimization in Smart Building Scenarios, 443–458.

[4] Marlon Alexander Braun, Pradyumn Kumar Shukla, and Hartmut Schmeck. 2011. Preference Ranking Schemes in Multi-Objective Evolutionary Algorithms. In *EMO (LNCS)*, Ricardo H.C. Takahashi and others (Eds.), Vol. 6576. Springer, 226–240.

[5] Marlon Alexander Braun, Pradyumn Kumar Shukla, and Hartmut Schmeck. 2015. Obtaining Optimal Pareto Front Approximations Using Scalarized Preference Information. In *GECCO*. ACM, New York, NY, USA, 631–638.

[6] Marlon Alexander Braun, Pradyumn Kumar Shukla, and Hartmut Schmeck. 2016. Angle-based Preference Models in Multi-objective Optimization. In *EMO*. 88–102.

[7] Carlos Coello Coello, Gary Lamont, and David Van Veldhuizen. 2007. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer.

[8] Kalyanmoy Deb and Mayank Goyal. 1996. A Combined Genetic Adaptive Search (GeneAS) for Engineering Design. *Computer Science and Informatics* 26 (1996), 30–45.

[9] Kalyanmoy Deb and Shivam Gupta. 2011. Understanding knee points in bicriteria problems and their implications as preferred solution principles. *Engineering optimization* 43, 11 (2011), 1175–1204.

[10] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. 2002. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on* 6, 2 (2002), 182–197.

[11] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. 2005. Scalable Test Problems for Evolutionary Multiobjective Optimization. In *Evolutionary Multiobjective Optimization*, Lakhmi Jain, Xindong Wu, Ajith Abraham, Lakhmi Jain, and Robert Goldberg (Eds.). Springer, 105–145.

[12] Juan J. Durillo and Antonio J. Nebro. 2011. jMetal: A Java framework for multi-objective optimization. *Advances in Engineering Software* 42, 10 (2011), 760–771.

[13] Salvatore Greco, Matthias Ehrgott, and José Figueira (Eds.). 2016. *Multiple criteria decision analysis: state of the art surveys*. Springer Science & Business Media.

[14] Nikolaus Hansen and Andreas Ostermeier. 2001. Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation* 9, 2 (2001), 159–195.

[15] Claus Hillermeier. 2001. *Nonlinear multiobjective optimization*. International Series of Numerical Mathematics, Vol. 135. Birkhäuser Verlag, Basel. vi+135 pages.

[16] Simon Huband, Philip Hingston, Luigi Barone, and Lyndon While. 2006. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation* 10, 5 (2006), 477–506.

[17] Daniel Kahneman and Amos Tversky. 1979. Prospect Theory: An Analysis of Decision Under Risk. *Econometrica: Journal of the Econometric Society* (1979), 263–291.

[18] R Timothy Marler and Jasbir S Arora. 2004. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization* 26, 6 (2004), 369–395.

[19] John Forbes Nash. 1950. The bargaining problem. *Econometrica: Journal of the Econometric Society* (1950), 155–162.

[20] A.J. Nebro, J.J. Durillo, J. Garcia-Nieto, C.A. Coello Coello, F. Luna, and E. Alba. 2009. SMPSO: A new PSO-based metaheuristic for multi-objective optimization. In *IEEE symposium on computational intelligence in multi-criteria decision-making, 2009. MCDM '09*. 66–73. DOI: http://dx.doi.org/10.1109/MCDM.2009.4938830

[21] Mike Preuss. 2015. *Multimodal Optimization by Means of Evolutionary Algorithms*. Springer.

[22] Pradyumn Kumar Shukla and Marlon Alexander Braun. 2013. Indicator Based Search in Variable Orderings: Theory and Algorithms. In *Evolutionary Multi-Criterion Optimization (Lecture Notes in Computer Science)*, Robin C. Purshouse, Peter J. Fleming, Carlos M. Fonseca, Salvatore Greco, and Jane Shaw (Eds.), Vol. 7811. Springer, 66–80.

[23] Pradyumn Kumar Shukla, Marlon Alexander Braun, and Hartmut Schmeck. 2013. Theory and Algorithms for Finding Knees. In *EMO (LNCS)*, Robin C. Purshouse, Peter J. Fleming, Carlos M. Fonseca, Salvatore Greco, and Jane Shaw (Eds.). LNCS, Vol. 7811. Springer, 156–170.

[24] Pradyumn Kumar Shukla, Marlon A. Braun, and Hartmut Schmeck. 2014. On the Interrelationships Between Knees and Aggregate Objective Functions. In *GECCO*. ACM, New York, NY, USA, 95–96.

[25] Qingfu Zhang and Hui Li. 2007. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *Evolutionary Computation, IEEE Transactions on* 11, 6 (Dec. 2007), 712–731.

[26] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. 2000. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary computation* 8, 2 (2000), 173–195.