Optimization Methods and Software

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/goms20

Using first-order information in direct multisearch for multiobjective optimization

R. Andreani, A. L. Custódio & M. Raydan

To cite this article: R. Andreani, A. L. Custódio & M. Raydan (2022): Using first-order information in direct multisearch for multiobjective optimization, Optimization Methods and Software, DOI: 10.1080/10556788.2022.2060971

To link to this article: https://doi.org/10.1080/10556788.2022.2060971

Published online: 21 Apr 2022.

OPTIMIZATION METHODS & SOFTWARE
https://doi.org/10.1080/10556788.2022.2060971

Taylor & Francis
Taylor & Francis Group

Using first-order information in direct multisearch for multiobjective optimization

R. Andreani[a], A. L. Custódio[b] and M. Raydan[c]

[a]Department of Applied Mathematics, IMECC-UNICAMP, University of Campinas, Campinas SP, Brazil; [b]Department of Mathematics, FCT-UNL-CMA, Caparica, Portugal; [c]Centro de Matemática e Aplicações (CMA), FCT, UNL, Caparica, Portugal

ABSTRACT

Derivatives are an important tool for single-objective optimization. In fact, it is commonly accepted that derivative-based methods present a better performance than derivative-free optimization approaches. In this work, we will show that the same does not always apply to multiobjective derivative-based optimization, when the goal is to compute an approximation to the complete Pareto front of a given problem. The competitiveness of direct multisearch (DMS), a robust and efficient derivative-free optimization algorithm, will be stated for derivative-based multiobjective optimization (MOO) problems, by comparison with MOSQP, a state-of-art derivative-based MOO solver. We will then assess the potential enrichment of adding first-order information to the DMS framework. Derivatives will be used to prune the positive spanning sets considered at the poll step of the algorithm. The role of ascent directions, that conform to the geometry of the nearby feasible region, will then be highlighted.

ARTICLE HISTORY
Received 15 February 2021
Accepted 16 March 2022

KEYWORDS
Multiobjective optimization; Pareto front computation; derivative-based methods; derivative-free optimization; direct multisearch

1. Introduction

Multiobjective optimization (MOO) problems appear frequently in engineering and scientific applications, in such diverse areas as civil engineering, environment, medicine or aerospace engineering [2,34,38,39], just to cite a few. The major feature of a MOO problem is the presence of finitely many components in the objective function, associated to conflicting objectives, that have to be simultaneously optimized. Hardly a single point will optimize all of them at once, hence a nonstandard notion of optimality is required. The fundamental optimality concept is that of Pareto optimal point, which is a point such that no improvement in all the components of the objective function can be achieved by moving to another feasible point. The image set of all Pareto optimal points (also called the Pareto front) is usually a continuum that may have disjoint components. In general, for a problem with $p > 1$ objectives, the Pareto front is a manifold of dimension $p-1$. For example, if $p = 2$ the Pareto front will be a curve (or a set of curve segments), which provides in a

CONTACT A. L. Custódio alcustodio@fct.unl.pt Department of Mathematics, FCT-UNL-CMA, Campus de Caparica, Caparica 2829-516, Portugal

compact way all the information required for a user to choose an appropriate Pareto optimal point as a compromise solution between the usually conflicting components of the objective function. Like in classical single-objective optimization, finding global Pareto optimal points is difficult, unless additional information is available about the objective function. Thus, MOO algorithms typically try to find local Pareto optimal points for the problems, meaning that the definition of Pareto optimality is satisfied in a neighbourhood of the current point.

There are several classes of MOO algorithms, depending not only on the level of smoothness of the objective function but also on the time when the user establishes an order preference for the different components of the objective function [35]. In this work, we will focus on methods with *a posteriori articulation of preferences*, which attempt to capture the whole Pareto front of the problem, never establishing preferences among the several components of the objective function. Evolutionary algorithms, or other similar heuristics, belong to this class. However, these algorithms lack a well-established convergence analysis and are usually slow in reaching the Pareto front of a given problem, requiring a large number of iterations and function evaluations [24]. When derivatives of the different components of the objective function are available, typical approaches are based on generating a single sequence of iterates that converges to a point with corresponding image lying on the Pareto front (one at a time); see, e.g. Refs. [9,25,26,28,37]. Multistart approaches [36] or scalarization techniques [23] can help in finding approximations to the complete Pareto front of a given MOO problem. The first can be computational expensive and the latter generally fails in detecting nonconvex parts of it [19].

Recently, novel approaches have been developed to approximate the entire Pareto front using first- and second-order information [14,27]. The first is based on a steepest descent approach, not necessarily considering all components of the objective function when defining the search direction. The so-called MOSQP method was proposed in Ref. [27], keeping a list of nondominated points, which approximates the Pareto front of the MOO problem. This list is improved both for spread along the Pareto front and optimality by solving single-objective constrained optimization problems derived as sequential quadratic programming (SQP) problems. In Ref. [27], numerical results are reported indicating the superiority of the MOSQP algorithm when compared to other state-of-the-art multiobjective solvers.

In derivative-free optimization, direct multisearch (DMS) [18] is also able to compute approximations to the complete Pareto front of a given MOO problem. This is a well-established algorithm, with theoretical results regarding convergence, and consistently used with good results both for benchmark of new solvers [13,33] and in real applications [7,31].

The purpose of the current work is twofold. Our first objective is to compare the performance of the derivative-free DMS method and the derivative-based MOSQP algorithm. In single-objective optimization, it is common to say that if derivatives are available or can be obtained at a reasonable cost (e.g. using finite-differences), then derivative-based optimization is preferable to derivative-free optimization methods [4, p. 6]. We will provide numerical results on a large set of benchmark MOO problems that allow to assess the numerical performance of derivative-based and derivative-free optimization solvers, when computing approximations to the complete Pareto fronts of derivative-based MOO problems. Our second objective is to assess the potential enrichment of adding first-order information, when derivatives are available, to the DMS framework. We will describe and

analyze several different combined techniques that maintain the search/poll paradigm of DMS, while adding in a convenient way gradient information to the poll step. Again, the value of the proposed strategies will be assessed through numerical experiments.

The remaining of this document is organized as follows. In Section 2, we present the MOO problem and briefly revise the derivative-free optimization method DMS, since it will be later modified to incorporate first-order information. Section 3 is devoted to a full numerical comparison between DMS and MOSQP methods. Section 4 details the use of first-order information to eliminate directions in the poll step of DMS and assesses the corresponding numerical performance. In Section 5, the usefulness of ascent directions is motivated by illustrating their performance on one properly chosen biobjective problem. Results are then reported in the complete test set. Finally, in Section 6, we present some concluding remarks.

2. DMS at a glance

We consider multiobjective minimization problems of the form

$$\min_{x\in\Omega} F(x) \equiv (f_1(x),\ldots,f_p(x))^\top, \tag{1}$$

where $p \geq 2$, $\Omega \subseteq \mathbb{R}^n$ represents the feasible region, typically defined as a box $\Omega = \{x \in \mathbb{R}^n : l \leq x \leq u\}$, and for each i $(1 \leq i \leq p)$ $f_i : \Omega \to \mathbb{R} \cup \{+\infty\}$ denotes a component of the objective function, which we assume to be strictly differentiable in Ω (continuity of the partial derivatives is not required).

The DMS method was originally proposed in Ref. [18], generalizing directional direct search to multiobjective derivative-free optimization. For a review on single objective derivative-free optimization methods, we recommend Refs. [4,15]. The algorithm has also been successfully extended to global multiobjective derivative-free optimization [17], by coupling it with a multistart initialization technique, where not all the initialized searches are conducted until the end.

Being a directional direct search method, each iteration of DMS conforms to the search/poll paradigm. The search step is optional, since the convergence results derive from the poll step of the algorithm. In fact, in the original presentation of the method [18], it was left empty and this will be the approach followed in the present work. Recently, the minimization of quadratic polynomial models, which have always played a key role in derivative-free methods for single objective optimization, was used for successfully defining a search step for DMS [6]. First-order information can surely be used to define appropriate search steps, following the strategies proposed in Ref. [6], but that will not be the subject of the present work, which will focus on the poll step.

We present a simplified description of the DMS framework, where only the poll step is considered, and where the globalization strategy is based on the use of integer lattices, meaning that all the points generated by the algorithm lie on an implicit mesh. For a more general description, we refer to the original work [18].

The algorithm initializes with a list of feasible, nondominated points (possibly just one) and corresponding stepsize parameters. Making use of the strict partial order induced by the cone $\mathbb{R}^p_+$, we say that point x dominates point y when $F(x) \prec_F F(y)$, i.e. when $F(y) - F(x) \in \mathbb{R}^p_+ \setminus \{0\}$. Given a list of feasible points, pairwise comparisons, using the previous

definition of dominance, allow to define the set of nondominated points, comprising all points which are not dominated by any other point in the list. This list, representing the current approximation to the Pareto front of the MOO problem, will be updated at every iteration by generating new feasible points which are compared with the points already stored in it, only keeping the nondominated ones.

At each iteration, a feasible nondominated point stored in the list and the associated stepsize parameter will be selected. Different strategies can be considered in the selection of this poll center. Currently it is based on a spread metric [18], in an attempt of reducing the gaps between consecutive points lying in the current approximation to the Pareto front of the problem.

The poll step of the algorithm consists on a local search around the selected poll center, by testing a set of directions with an adequate geometry, scaled by the corresponding stepsize parameter. Typically, positive spanning sets are considered [20] that should conform to the geometry of the nearby active constraints of the current poll center [32].

For convergence purposes, the poll step can be performed either in a complete or an opportunistic way. In the latter, the polling procedure is stopped once a new feasible nondominated point is found. The complete approach tests all the poll directions, only adding to the list the new feasible nondominated points found (and removing from the list all the dominated ones). We will follow this last approach, which is the one corresponding to the original algorithmic implementation of DMS [18], in an attempt of maximizing the number of feasible nondominated points generated at each iteration.

The final step of each iteration is the update of the stepsize parameter, which is increased or kept constant for successful iterations and decreased for unsuccessful ones. An iteration is said to be successful if the list changes, meaning that at least one new feasible nondominated point was found. Unsuccessful iterations keep the list unchanged.

A simplified description of the DMS framework is provided in Algorithm 2.1. For a complete description , see Ref. [18].

Algorithm 2.1: A simplified description of DMS

Initialization

Choose a set of nondominated points $\{x^i_{ini} \in \Omega,\ i \in I\}$ with $f_j(x^i_{ini}) < +\infty, \forall j \in \{1,\ldots,p\}, \forall\, i \in I$, $\alpha^i_{ini} > 0$, $i \in I$ initial stepsizes, $0 < \beta_1 \leq \beta_2 < 1$ the coefficients for stepsize contraction and $\gamma \geq 1$ the coefficient for stepsize expansion. Let $\mathcal{D}$ be a set of positive spanning sets. Initialize the list of feasible nondominated points and corresponding stepsize parameters $L_0 = \{(x^i_{ini}; \alpha^i_{ini}), i \in I\}$.

For $k = 0, 1, 2, \ldots$

(1) **Selection of an iterate point:** Order the list L_k according to some criteria and select the first item $(x;\alpha) \in L_k$ as the current iterate and stepsize parameter (thus setting $(x_k;\alpha_k) = (x;\alpha)$).
(2) **Poll step:** Choose a positive spanning set D_k from the set $\mathcal{D}$. Evaluate F at the feasible poll points belonging to $\{x_k + \alpha_k d :\ d \in D_k\}$. Compute L_{trial} by removing all dominated points from $L_k \cup \{(x_k + \alpha_k d; \alpha_k) : d \in D_k \wedge x_k + \alpha_k d \in \Omega\}$. If $L_{trial} \neq L_k$, declare the iteration (and the poll step) successful and set $L_{k+1} =$

L_{trial}. Otherwise, declare the iteration (and the poll step) unsuccessful and set $L_{k+1} = L_k$.

(3) **Stepsize parameter update:** If the iteration was successful then maintain or increase the corresponding stepsize parameters, by considering $\alpha_{k,new} \in [\alpha_k, \gamma\alpha_k]$ and replacing all the new points $(x_k + \alpha_k d; \alpha_k)$ in L_{k+1} by $(x_k + \alpha_k d; \alpha_{k,new})$. Replace also $(x_k; \alpha_k)$, if in L_{k+1}, by $(x_k; \alpha_{k,new})$.
Otherwise, decrease the stepsize parameter, by choosing $\alpha_{k,new} \in [\beta_1\alpha_k, \beta_2\alpha_k]$, and replace the poll pair $(x_k; \alpha_k)$ in L_{k+1} by $(x_k; \alpha_{k,new})$.

The convergence of DMS has been established in Ref. [18], closely following the arguments used in the analysis of single-objective directional direct search methods. After stating the existence of a subsequence of stepsize parameters converging to zero, this property is used for establishing Pareto-Clarke Karush–Kuhn–Tucker(KKT) criticality. The result is formalized in Theorem 2.3 for limit points of convergent refining subsequences.

Definition 2.1: A subsequence $\{x_k\}_{k\in K}$ of iterates corresponding to unsuccessful poll steps is said to be a refining subsequence if $\{\alpha_k\}_{k\in K}$ converges to zero.

The concept of refining direction is associated with convergent refining subsequences and is formalized in Definition 2.2.

Definition 2.2: Let x_* be the limit point of a convergent refining subsequence $\{x_k\}_{k\in K}$. If the limit $\lim_{k\in K'} d_k/\|d_k\|$ exists, where $K' \subseteq K$ and $d_k \in D_k$, and if $x_k + \alpha_k d_k \in \Omega$, for sufficiently large $k \in K'$, then this limit is said to be a refining direction for x_*.

Assuming the density of the set of refining directions in the Clarke tangent cone to Ω computed at limit points of refining subsequences [12], the convergence of DMS is established.

Theorem 2.3 (see Ref. [18]): *Consider a refining subsequence $\{x_k\}_{k\in K}$ converging to $x_* \in \Omega$. Assume that F is strictly differentiable at x_* and that the interior of the tangent cone to Ω at x_* is nonempty. If the set of refining directions for x_* is dense in the Clarke tangent cone to Ω at x_*, then x_* is a Pareto-Clarke-KKT critical point, i.e.*

$$\forall d \in T_\Omega^{Cl}(x_*), \quad \exists j(d) \in \{1, 2, \ldots, p\} : \nabla f_{j(d)}(x_*)^\top d \geq 0.$$

Recently, worst-case complexity bounds were provided for DMS, but considering a globalization strategy that requires sufficient decrease for accepting new points [16]. For a particular algorithmic instance, which considers a stricter criterion for accepting new nondominated points, DMS presents a worst-case complexity bound of $\mathcal{O}(\epsilon^{-2})$, similar to the one of steepest descent.

3. Comparing DMS and MOSQP

Derivatives are a keystone for optimization. As previously mentioned, in single-objective optimization, when in the presence of smooth functions, derivative-based methods are

Table 1. The test set considered in the numerical experiments. Here, n represents the number of variables and p is the number of components of the objective function.

Problem	n	p	Problem	n	p	Problem	n	p	Problem	n	p
BK1	2	2	DTLZ4n2	2	2	lovison3	2	2	MOP7	2	3
CL1	4	2	DTLZ6	22	3	lovison4	2	2	SK1	1	2
Deb41	2	2	DTLZ6n2	2	2	lovison5	3	3	SK2	4	2
Deb513	2	2	ex005	2	2	lovison6	3	3	SP1	2	2
Deb521b	2	2	Far1	2	2	LRS1	2	2	SSFYY1	2	2
DG01	1	2	Fonseca	2	2	MHHM1	1	3	SSFYY2	1	2
DPAM1	10	2	IKK1	2	3	MHHM2	2	3	TKLY1	4	2
DTLZ1	7	3	IM1	2	2	MLF1	1	2	VFM1	2	3
DTLZ1n2	2	2	Jin1	2	2	MLF2	2	2	VU1	2	2
DTLZ2	12	3	Jin3	2	2	MOP1	1	2	VU2	2	2
DTLZ2n2	2	2	L2ZDT2	30	2	MOP2	4	2	ZDT2	30	2
DTLZ3	12	3	L3ZDT2	30	2	MOP3	2	2	ZLT1	10	3
DTLZ3n2	2	2	lovison1	2	2	MOP5	2	3			
DTLZ4	12	3	lovison2	2	2	MOP6	2	2			

preferable to derivative-free optimization algorithms, even if one has to spend some time and effort to obtain good quality derivatives (see Ref. [15, p. 7] or Ref. [4, p. 6]). In this section, we will assess the situation for multiobjective derivative-based optimization, when the goal is to compute approximations to complete Pareto fronts.

For that, DMS algorithm [18] will be numerically tested against MOSQP [27]. The latter is a recent solver proposed for multiobjective derivative-based optimization, making use of a SQP approach and for which a Matlab implementation is publicly available. MOSQP keeps a list of nondominated points that is improved both for spread along the Pareto front and optimality by solving single-objective constrained optimization problems. Thus, MOSQP is able to generate approximations to complete Pareto fronts, an advantage over classical derivative-based MOO solvers, which compute a single Pareto point. At the time of the release, extensive numerical results were provided for MOSQP, including a comparison with a classical scalarization approach for biobjective problems [27]. The good results obtained allowed the authors to conclude that MOSQP should be 'the preferred solution framework for MOO problems when derivatives of objective and constraint functions are available' [27], which justifies our algorithmic choice as baseline against DMS. Default parameters were considered for both solvers, with exception to the maximum number of function evaluations allowed, which was set to 20,000. In some cases, a small budget of 500 function evaluations was additionally considered to ensure that the conclusions drawn were not dependent on the large number of function evaluations allowed.

As test set, we considered the collection of 100 bound constrained MOO problems available at http://www.mat.uc.pt/dms. This collection was previously used to test DMS and MOSQP, at the time of their first release [18,27]. From this collection, we selected a total of 54 problems, for which we were able to guarantee the existence of derivatives. Table 1 reports the resulting test set, which comprises problems with 2 or 3 components in the objective function and a number of variables, n, between 1 and 30. We notice that MOO problems with more than three components in the objective function are commonly denoted by 'many-objective optimization' problems [10] and are not the focus of any of the solvers considered in the present work, requiring specific techniques to be efficiently addressed [8,30,41].

For performance assessment, we considered typical metrics from the MOO literature, like is the case of purity and spread metrics, as defined in Ref. [18], and also the hypervolume indicator [42,44]. While other metrics could have been considered [5,11,40], these are typical choices in recent works [3,6,13].

In a simplified view, purity measures the percentage of nondominated points generated by a given solver. For problem $\hat{p} \in \mathcal{P}$ and solver $s \in \mathcal{S}$, purity is defined by the ratio

$$\bar{t}_{\hat{p},s} = \frac{|F_{\hat{p},s} \cap F_{\hat{p}}|}{|F_{\hat{p},s}|},$$

where $F_{\hat{p},s}$ denotes the approximation to the Pareto front computed for problem $\hat{p} \in \mathcal{P}$ by solver $s \in \mathcal{S}$ and $F_{\hat{p}}$ is a reference Pareto front for problem $\hat{p} \in P$. This reference Pareto front is computed by joining the final approximations computed by any of the solvers tested and removing from it all the dominated points. A value of purity near one indicates that the majority of the points generated by the corresponding solver is nondominated. However, these could be concentrated in a single part of the true Pareto front. Spread metrics are required to have a fair assessment of the solver's performance.

Since the goal is to build an approximation to the complete Pareto front of each problem, the computation of spread metrics initiates with the computation of the so-called 'extreme points' of the Pareto front (see Ref. [18]). The spread Γ measures the maximum gap between consecutive points lying in the approximated Pareto front. The metric $\Gamma_{\hat{p},s} > 0$ for problem $\hat{p} \in \mathcal{P}$ and solver $s \in \mathcal{S}$ is given by

$$\Gamma_{\hat{p},s} = \max_{j \in \{1,\ldots,p\}} \left(\max_{i \in \{0,\ldots,N\}} \{\delta_{i,j}\} \right), \tag{2}$$

where $\delta_{i,j} = (f_j(x_{i+1}) - f_j(x_i))$, $x_1, x_2, \ldots, x_N$ represent the points generated by solver $s \in \mathcal{S}$ for problem $\hat{p} \in \mathcal{P}$, and x_0, x_{N+1} correspond to the 'extreme points'. Implicitly, we are assuming that the objective function values have been sorted by increasing order for each objective $j \in \{1, \ldots, p\}$.

The spread metric Δ [21] measures the uniformity of the gaps across the approximation to the Pareto front:

$$\Delta_{\hat{p},s} = \max_{j \in \{1,\ldots,p\}} \left(\frac{\delta_{0,j} + \delta_{N,j} + \sum_{i=1}^{N-1} |\delta_{i,j} - \bar{\delta}_j|}{\delta_{0,j} + \delta_{N,j} + (N-1)\bar{\delta}_j} \right), \tag{3}$$

where $\bar{\delta}_j$, for $j = 1, \ldots, p$, represents the average of the distances $\delta_{i,j}$, $i = 1, \ldots, N-1$.

The fourth metric considered is the hypervolume indicator [44], which measures the volume of the portion of the objective function space that is dominated by the computed approximation to the Pareto front of the problem, and upper bounded by a given reference point $U_{\hat{p}} \in \mathbb{R}^p$. This reference point should be dominated by all points belonging to the approximations computed for the Pareto front of a given problem $\hat{p} \in \mathcal{P}$. Formally, it can be defined as

$$HV_{\hat{p},s} = Vol\{y \in \mathbb{R}^p \,|\, y \leq U_{\hat{p}} \wedge \exists x \in F_{\hat{p},s} : x \leq y\} = Vol\left(\bigcup_{x \in F_{\hat{p},s}} [x, U_{\hat{p}}] \right),$$

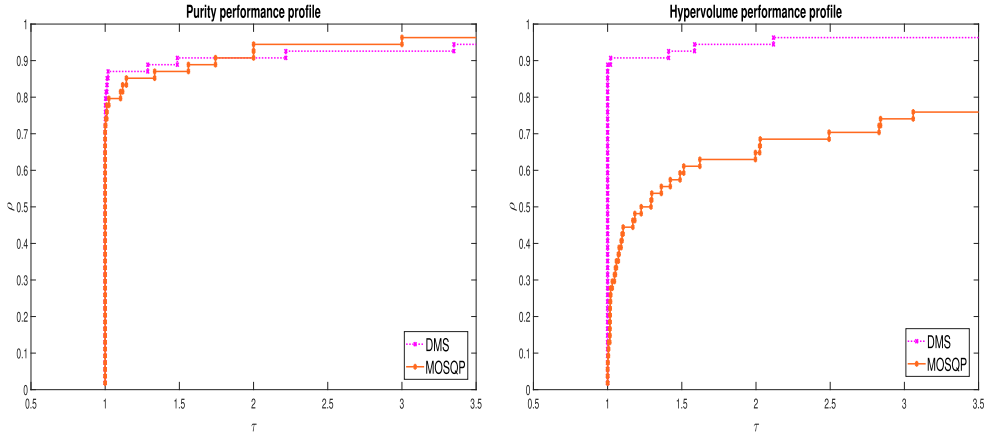

Figure 1. Performance profiles for purity and hypervolume metrics, comparing the original DMS implementation and MOSQP (maximum budget of 20,000 function evaluations).

where $Vol(.)$ denotes the Lebesgue measure of a p-dimensional set of points and $[x, U_{\hat{p}}]$ denotes the interval box with lower corner x and upper corner $U_{\hat{p}}$. The approach proposed in Ref. [29] was used for its practical computation and the resulting hypervolume values were scaled to the interval $[0, 1]$, following the procedure described in Ref. [6].

Additionally, computational time is also used as a performance metric. In this case, since it is not deterministic, results respect to average computational time, obtained from a sample of 30 runs for each solver on each problem.

Performance profiles [22] will be depicted for the five metrics considered. Let $t_{\hat{p},s}$ denote the performance of solver $s \in \mathcal{S}$ on problem $\hat{p} \in \mathcal{P}$, assuming that lower values of $t_{\hat{p},s}$ indicate a better performance. Each performance profile represents the curve

$$\rho_s(\tau) = \frac{1}{|\mathcal{P}|} |\{\hat{p} \in \mathcal{P} : r_{\hat{p},s} \leq \tau\}|,$$

with $r_{\hat{p},s} = t_{\hat{p},s} / \min\{t_{\hat{p},\bar{s}} : \bar{s} \in \mathcal{S}\}$. In the case of purity and hypervolume metrics, larger values indicate better performance. Thus, when computing performance profiles for these two metrics, we set $t_{\hat{p},s} = 1/t_{\hat{p},s}$ as proposed in Ref. [18].

Figures 1–3 compare DMS against MOSQP, when a maximum budget of 20,000 function evaluations is considered.

The two solvers present a similar performance in terms of robustness for purity and Δ metrics, with DMS being more efficient in terms of purity and MOSQP with respect to the Δ metric. However, there is a huge gain in performance with DMS when hypervolume or the Γ metrics are considered. In terms of average computational time, there are clear gains of MOSQP. The sorting procedure performed by DMS at each iteration, associated to the selection of the poll center, is mainly responsible for this bad performance. In fact, if the total budget of function evaluations is reduced to only 500, the good performance of DMS when compared against MOSQP appears not only associated to hypervolume and spread Γ but also for purity and computational time (see Figures 4–6).

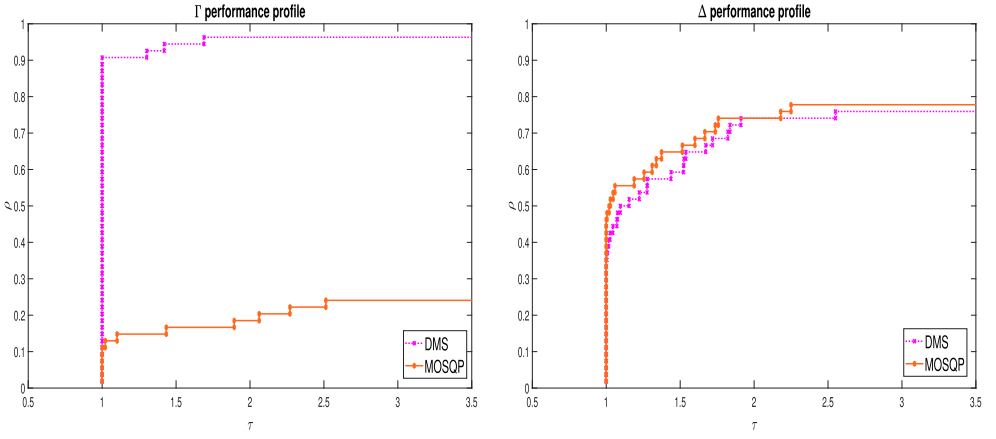

Figure 2. Performance profiles for Γ and Δ metrics, comparing the original DMS implementation and MOSQP (maximum budget of 20,000 function evaluations).

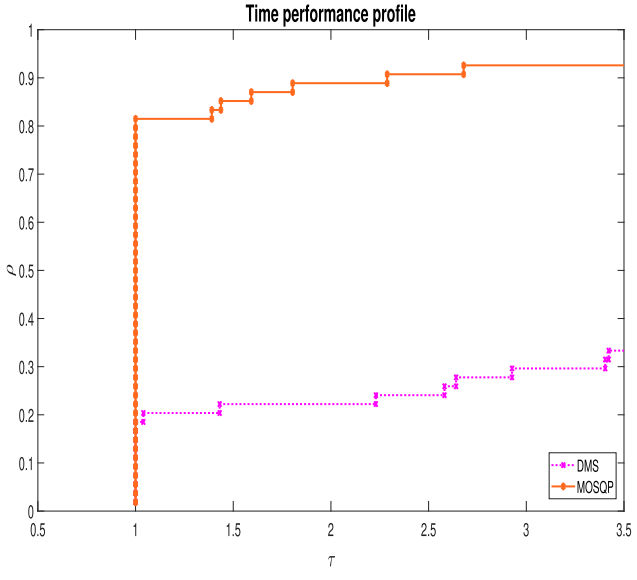

Figure 3. Performance profile for computational time (average of 30 runs), comparing the original DMS implementation and MOSQP (maximum budget of 20,000 function evaluations).

The benefits of an enriched set of directions are clear (and will be even clearer in the following sections). If derivative-based solvers are preferable to derivative-free optimization methods for single-objective optimization, the same does not necessarily apply to MOO.

4. Pruning the poll set

At each iteration of DMS, a positive spanning set is selected as poll set. The poll points correspond to directions in the poll set scaled by the stepsize parameter. The objective function

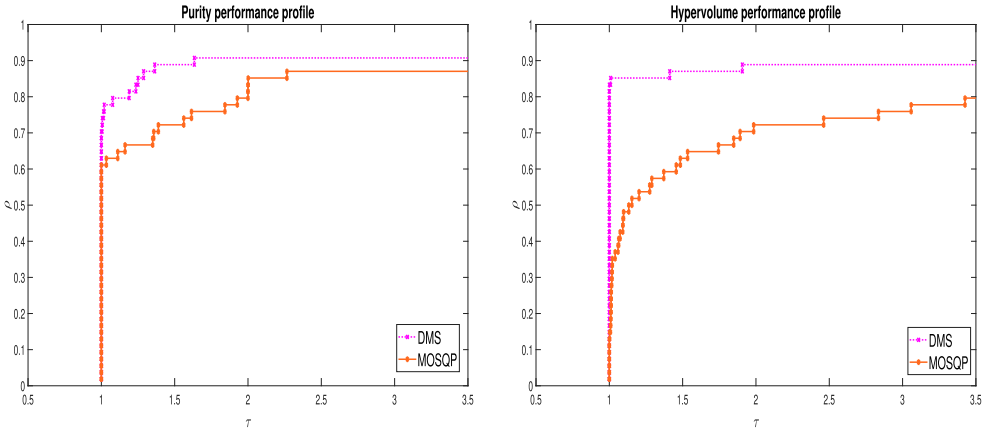

Figure 4. Performance profiles for purity and hypervolume metrics, comparing the original DMS implementation and MOSQP (maximum budget of 500 function evaluations).

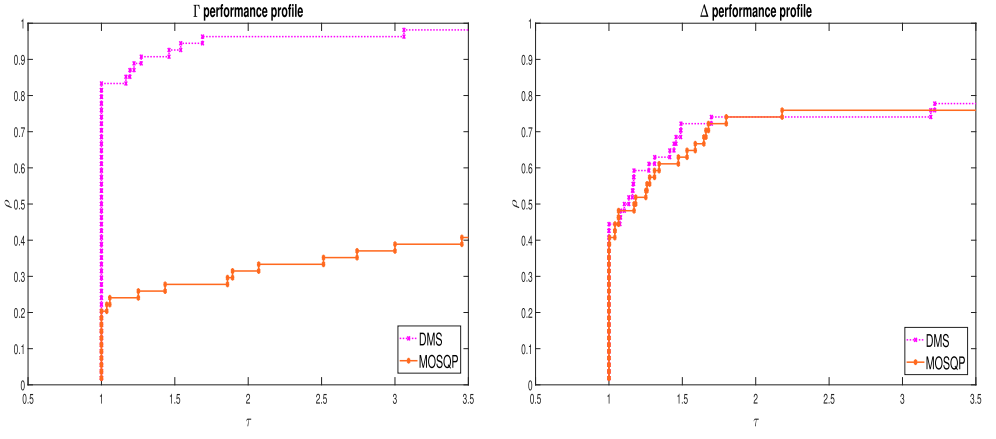

Figure 5. Performance profiles for Γ and Δ metrics, comparing the original DMS implementation and MOSQP (maximum budget of 500 function evaluations).

will then be evaluated at all the feasible poll points, independently of corresponding or not to descent directions.

The following result is well known for positive spanning sets (see Theorem 2.3 in Ref. [15]).

Theorem 4.1: *If* $\{v_1, \ldots, v_r\}$, *with* $v_j \neq 0$ *for all* $j \in \{1, \ldots, r\}$, *positively spans* $\mathbb{R}^n$ *then for every nonzero vector* $d \in \mathbb{R}^n$ *there is an index* $j \in \{1, \ldots, r\}$ *such that* $d^\top v_j > 0$.

Considering strict differentiability of each component of the objective function F, and setting $d = \nabla f_i(x)$ or $d = -\nabla f_i(x)$, for $i \in \{1, \ldots, p\}$, Theorem 4.1 allows us to conclude that in every positive spanning set, for each component of the objective function, we can find at least one ascent and one descent direction.

Thus, at each iteration, for $i \in \{1, \ldots, p\}$, if $\nabla f_i(x_k) \neq 0$, $d_k = -\nabla f_i(x_k)$ can be used to prune the positive spanning set, only keeping directions that are descent according to at

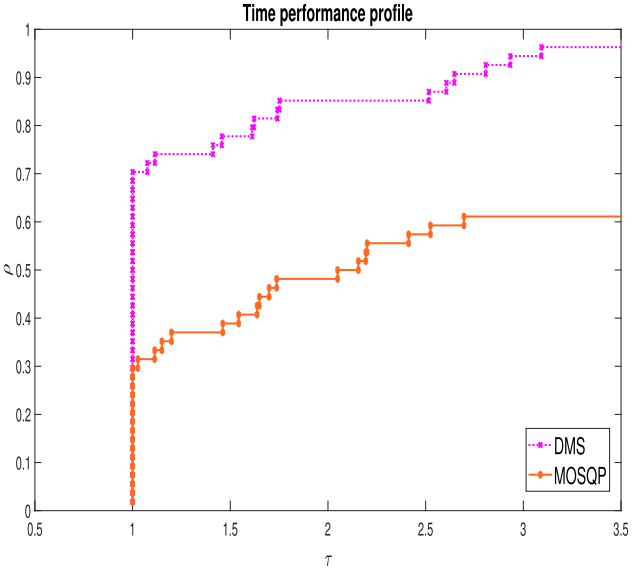

Figure 6. Performance profile for computational time (average of 30 runs), comparing the original DMS implementation and MOSQP (maximum budget of 500 function evaluations).

least one component of the objective function. Since we are only discarding directions that are ascent according to all components of the objective function, the convergence results of Section 2 still hold. The pruned set of directions, D_k^P, to be considered as poll directions for DMS at Step 2 of Algorithm 1 will then be

$$D_k^P = \bigcup_{i\in\{1,\ldots,p\}} \{d \in D_k : -\nabla f_i(x_k)^\top d > 0\}.$$

This strategy can be ineffective when the number of components of the objective function is high ($p > 3$), what is commonly known as many-objective optimization [10]. However, for problems comprising two or three components in the objective function, it could lead to considerable savings in function evaluations at each poll step.

The idea of pruning positive spanning sets was already proposed in single objective derivative-free optimization [1]. In this setting, it is easy to see that the cardinality of the pruned set will be $1 \leq |D_k^P| \leq |D_k| - 1$. The authors were even able to provide a particular enriched positive spanning set, that always reduces to a singleton after pruning.

If the goal is to generate an approximation to the complete Pareto front of a given problem, we do not wish to reduce the poll directions to a singleton, as we do not wish to use opportunistic approaches, which would generate at most a new feasible nondominated point at each iteration. Moreover, in MOO, due to the presence of conflicting objectives, we cannot ensure the presence of a descent direction, according to all the components of the objective function [18]. In fact, it is possible to build examples where the cone of descent directions, considering all components of the objective function, can be as narrow as one would desire.

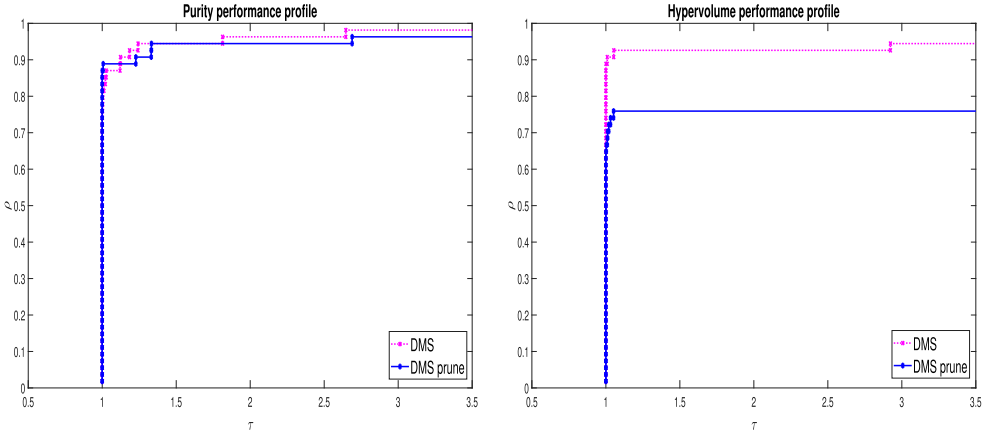

Figure 7. Performance profiles for purity and hypervolume metrics, comparing the original DMS implementation and a new version, where poll directions are pruned using first-order information (maximum budget of 20,000 function evaluations).

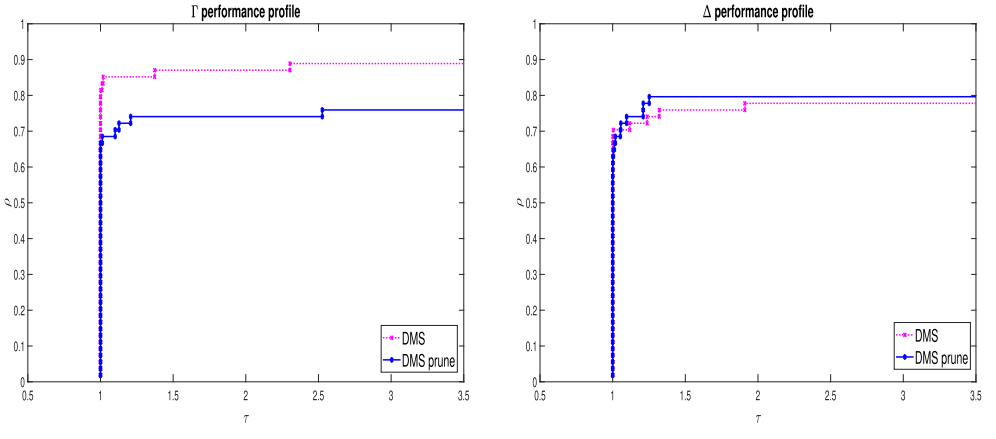

Figure 8. Performance profiles for Γ and Δ metrics, comparing the original DMS implementation and a new version, where poll directions are pruned using first-order information (maximum budget of 20,000 function evaluations).

The proposed strategy was implemented and numerically tested against the original DMS algorithm [18]. Figures 7–9 report the corresponding comparison, for a budget of 20,000 function evaluations.

In its current form, it is clear that the pruning strategy is not successful. In fact, there is a considerable decrease in performance regarding the hypervolume and Γ metrics. While it could seem surprising, as we will see in Section 5, ascent directions play an important role when the goal is to compute an approximation to the complete Pareto front of a given MOO problem. The computation of derivatives also causes a decrease in the efficiency associated to the version of DMS that uses the pruning strategy.

Comparisons were also performed between DMS with the pruning strategy and MOSQP, for the same maximum budget of function evaluations. Since the results are very similar to the ones reported in Section 3, we are not presenting the corresponding

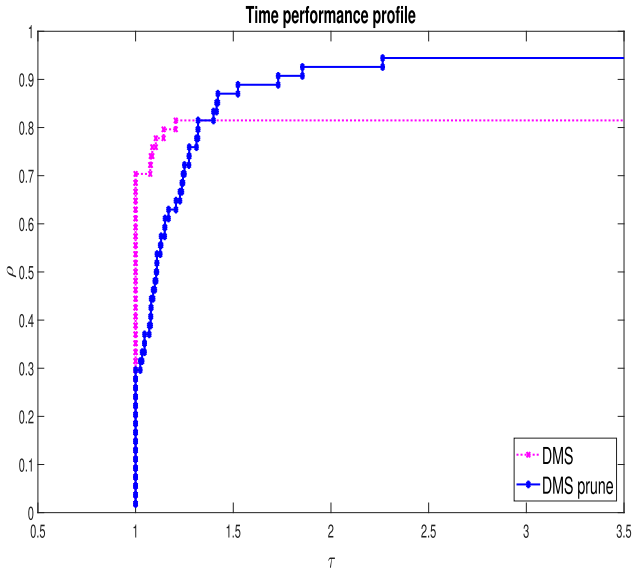

Figure 9. Performance profile for computational time (average of 30 runs), comparing the original DMS implementation and a new version, where poll directions are pruned using first-order information (maximum budget of 20,000 function evaluations).

profiles. The clear advantage of DMS, now using the pruning strategy, over MOSQP still prevails, even if in terms of hypervolume the two solvers are closer than when comparing to the original version of DMS. Regarding computational time, the pruning strategy allows a slight improvement in the performance of DMS, but it is still far from the one of MOSQP.

5. The role of ascent directions

In the presence of constraints, pruning the ascent directions is not always a good strategy. Consider the biobjective minimization problem ZDT2, with $n = 30$ [43]. In this case, if we provide as initialization one Pareto critical point, the algorithm stalls, being unable to generate other Pareto critical points in the Pareto front. This behaviour is accordingly to the convergence results derived for DMS, which only guarantee convergence to a single Pareto critical point. By providing ascent directions, that conform to the geometry of the nearby feasible region, the algorithm is able to proceed and generate a large number of Pareto critical points. Figure 10 illustrates the situation.

Thus, the approach taken was to return to the original positive spanning set D_k (without pruning) at some iterations. Assume that at a given iteration, the original positive spanning set was pruned and D_k^P was used as poll set, but the algorithm was unable to proceed because every poll point was infeasible. At the next iteration, pruning will not be applied, and the original positive spanning set D_k will be considered as the set of poll directions. Again, since we are only disregarding directions that are ascent according to all components of the objective function, and only at some iterations, the convergence results of Section 2

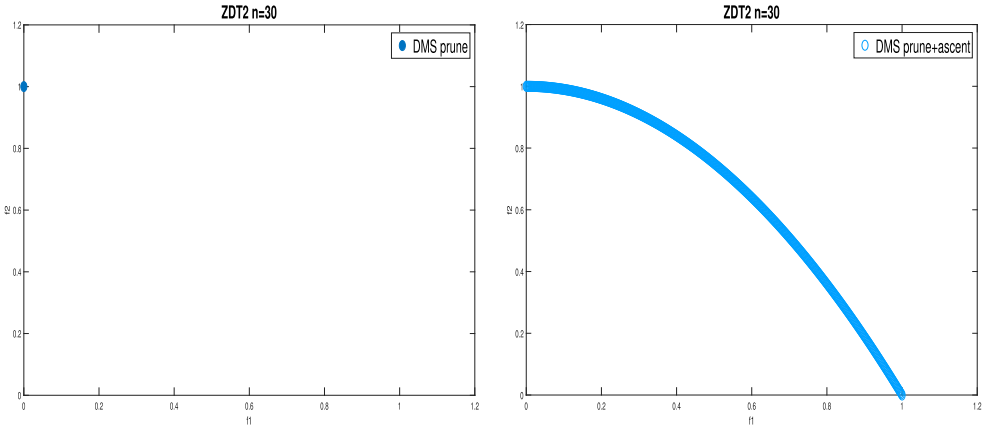

Figure 10. Final approximations to the Pareto front of problem ZDT2, generated by two different algorithmic variants of DMS. On the left, positive spanning sets are pruned to sets only comprising descent directions. On the right, ascent directions are considered at some iterations.

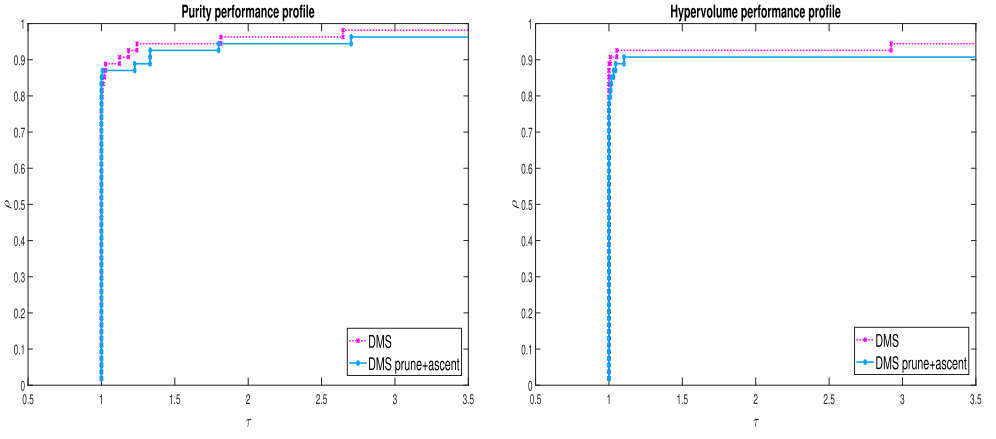

Figure 11. Performance profiles for purity and hypervolume metrics, comparing the original DMS implementation and a new version, where poll directions are pruned using first-order information, but not at all the iterations (maximum budget of 20,000 function evaluations).

continue to hold. Figures 11–13 report performance profiles comparing this new approach with the original implementation of DMS.

Now, the two variants of DMS are extremely close in terms of performance regarding the quality of the solutions produced, but the new approach brings some advantage in terms of the Γ metric. However, considering the average computational time, the computation of derivatives always decreases the performance of the DMS variant that makes use of it. The advantages of the new approach are clearer if the computational budget is reduced from 20,000 to only 500 functions evaluations (see Figures 14–16).

Savings in function evaluations allow clear improvements in terms of efficiency, both for purity and hypervolume. Regarding the Γ metric, there is a clear advantage of the new variant over the classical DMS approach. As expected, the computation of derivatives continues to decrease the performance in terms of average computational time.

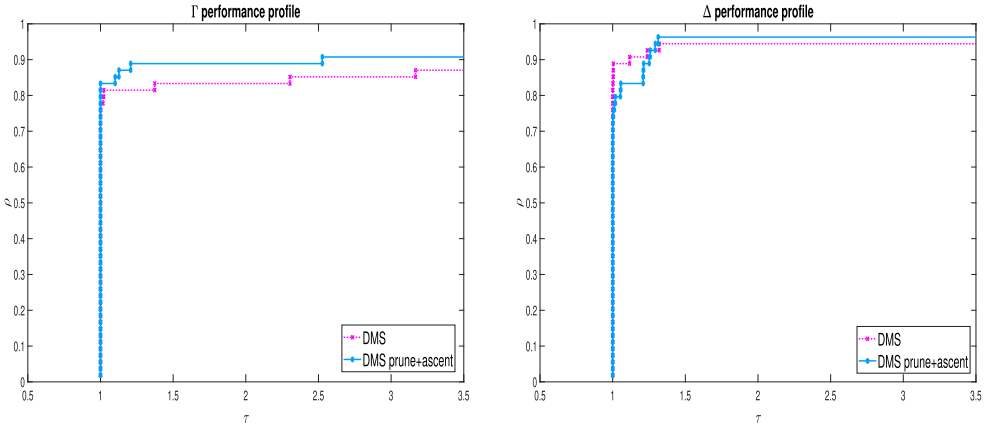

Figure 12. Performance profiles for Γ and Δ metrics, comparing the original DMS implementation and a new version, where poll directions are pruned using first-order information, but not at all the iterations (maximum budget of 20,000 function evaluations).

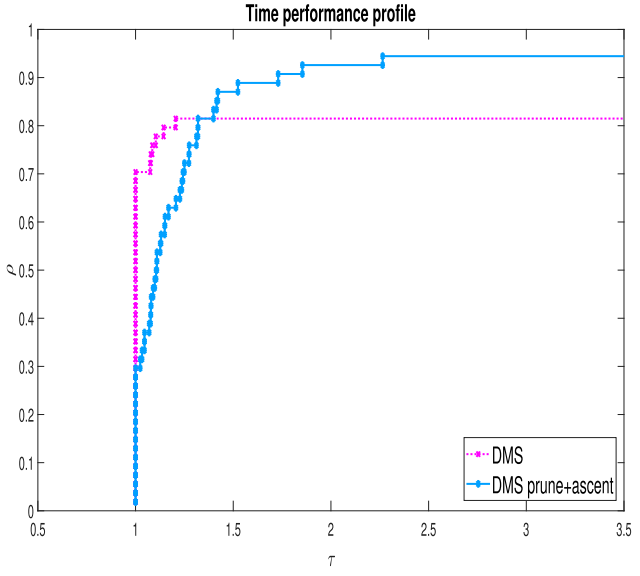

Figure 13. Performance profile for computational time (average of 30 runs), comparing the original DMS implementation and a new version, where poll directions are pruned using first-order information, but not at all the iterations (maximum budget of 20,000 function evaluations).

Comparing with MOSQP, again considering a budget of only 500 function evaluations, there is also a clear advantage of the new variant in three of the five metrics considered, namely purity, hypervolume and Γ metric, with an equal performance for the Δ metric and average computational time. Figures 17–19 report the results. Thus, the good performance of DMS variants over MOSQP is not the result of large budgets of function evaluations, but a consequence of richer sets of directions, including ascent ones.

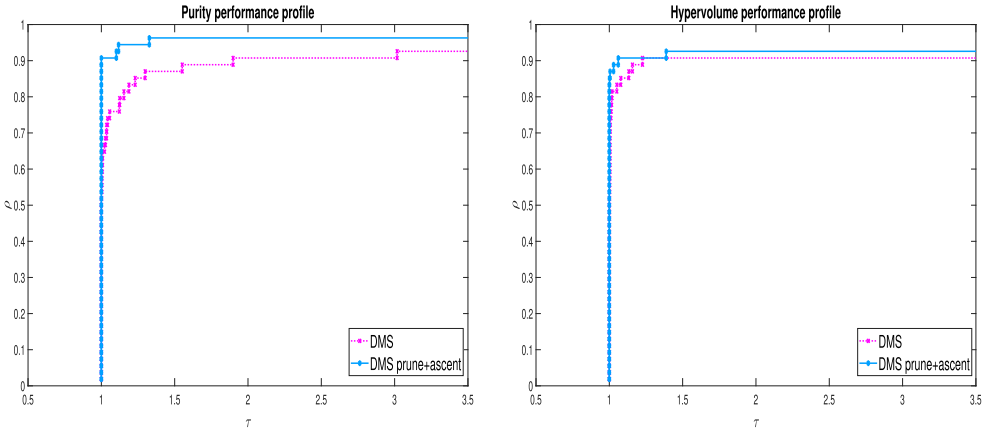

Figure 14. Performance profiles for purity and hypervolume metrics, comparing the original DMS implementation and a new version, where poll directions are pruned using first-order information, but not at all the iterations (maximum budget of 500 function evaluations).

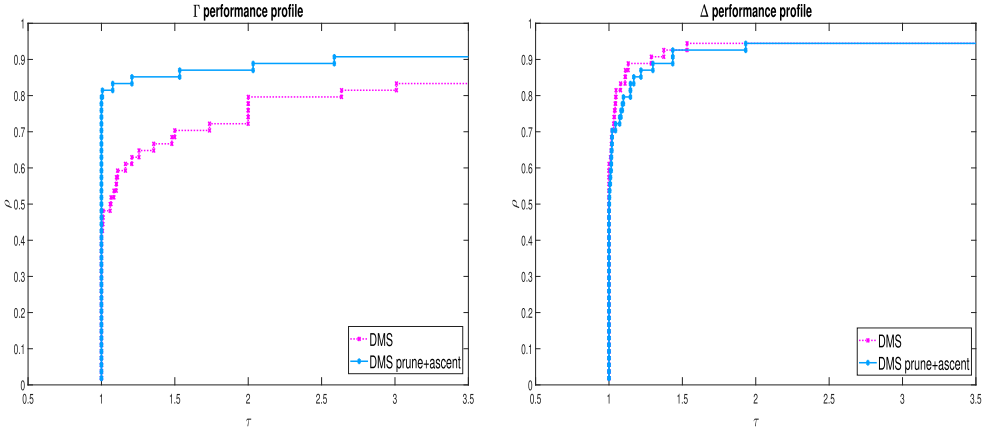

Figure 15. Performance profiles for Γ and Δ metrics, comparing the original DMS implementation and a new version, where poll directions are pruned using first-order information, but not at all the iterations (maximum budget of 500 function evaluations).

Considering that the experiments of the previous sections indicate that there could be a large difference in the computational time required by each solver to address the different problems, again we compared this final version of DMS against MOSQP, this time considering a budget of only 1 second of computational time as stopping criterion. Similarly to previous tests, 30 runs were performed and average results are reported in Figures 20 and 21.

In this case, MOSQP improves the performance in the purity metric, with a clear advantage over the DMS variant. In fact, MOSQP achieves convergence (i.e. it successfully completes its required iterations) in all the considered problems within 1 second, whereas the same does not apply to DMS version. However, the advantages of performance of this final DMS version for the hypervolume and the Γ spread metrics continue to be clear.

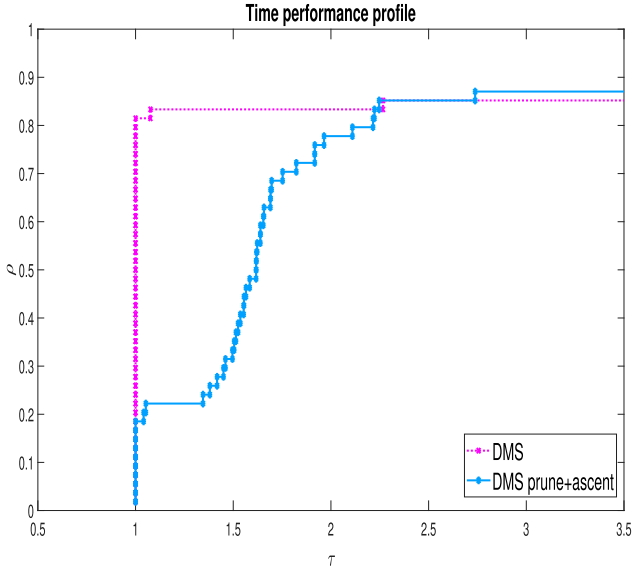

Figure 16. Performance profiles for computational time (average of 30 runs), comparing the original DMS implementation and a new version, where poll directions are pruned using first-order information, but not at all the iterations (maximum budget of 500 function evaluations).

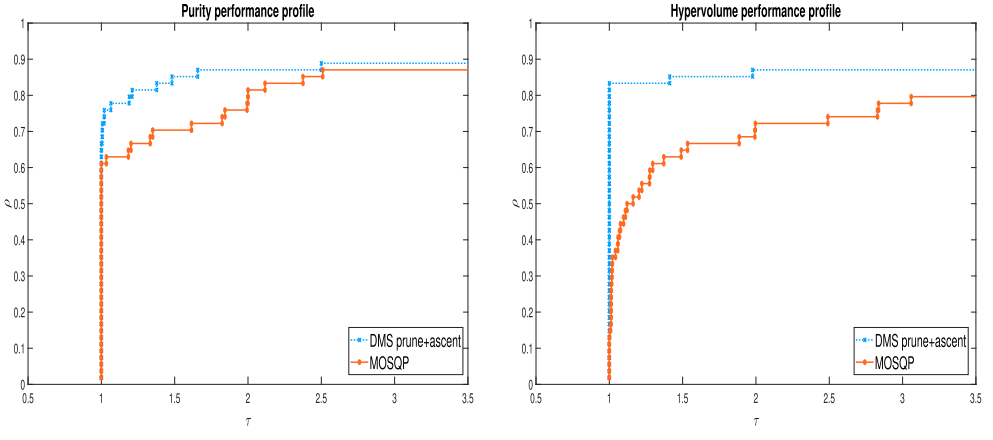

Figure 17. Performance profiles for purity and hypervolume metrics, comparing MOSQP and the new version of DMS, where poll directions are pruned using first-order information, but not at all the iterations (maximum budget of 500 function evaluations).

6. Concluding remarks

DMS was proposed in Ref. [18] as a robust and efficient algorithm, able to generate approximations to the complete Pareto front of MOO problems. Surprisingly, the numerical experiments conducted showed that it can be a strong competitor against the derivative-based solver MOSQP, evidencing that in MOO, when the goal is to generate an approximation to the complete Pareto front of a given problem, even if first-order derivatives are available, derivative-free solvers can be good alternatives to derivative-based approaches.

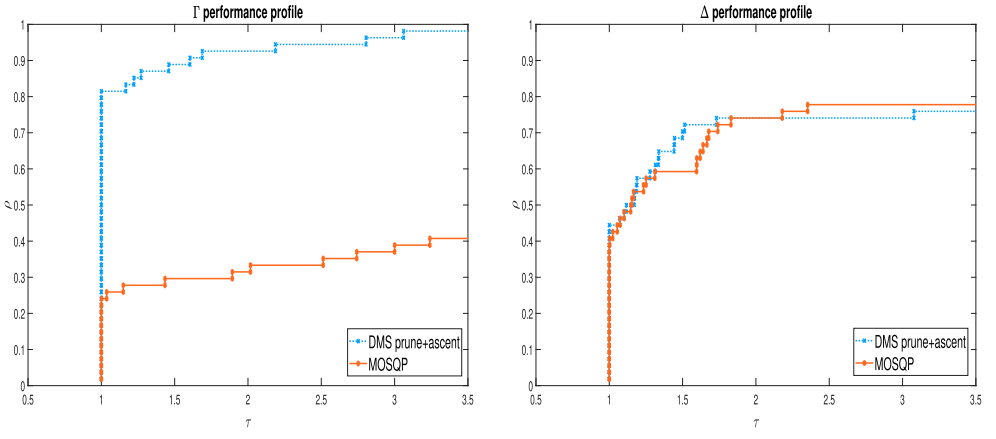

Figure 18. Performance profiles for Γ and Δ metrics, comparing MOSQP and the new version of DMS, where poll directions are pruned using first-order information, but not at all the iterations (maximum budget of 500 function evaluations).

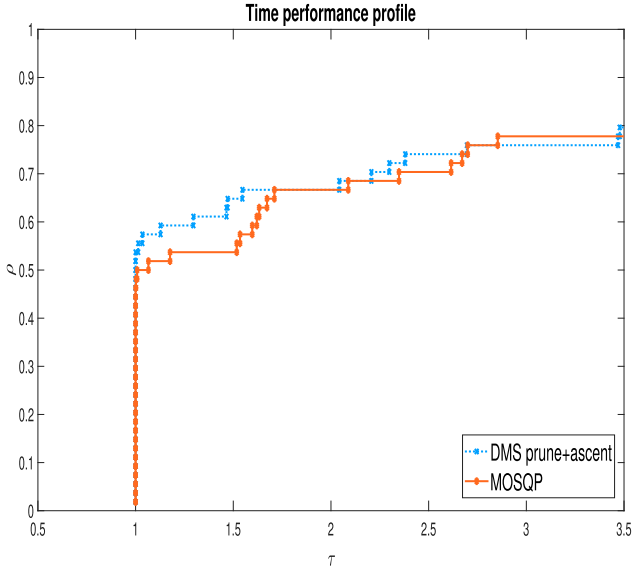

Figure 19. Performance profiles for computational time (average of 30 runs), comparing MOSQP and the new version of DMS, where poll directions are pruned using first-order information, but not at all the iterations (maximum budget of 500 function evaluations).

Derivatives can be used to prune the positive spanning sets to be considered as poll directions. However, care should be taken because ascent directions, that conform to the geometry of the nearby feasible region, can have an important role in the ability of generating a complete approximation to the Pareto front of a given problem.

The new variant of DMS, which prunes the poll set of directions, but that at some iterations considers its enrichment with ascent directions, showed to be competitive both with the derivative-based solver MOSQP and with the original implementation of DMS. For

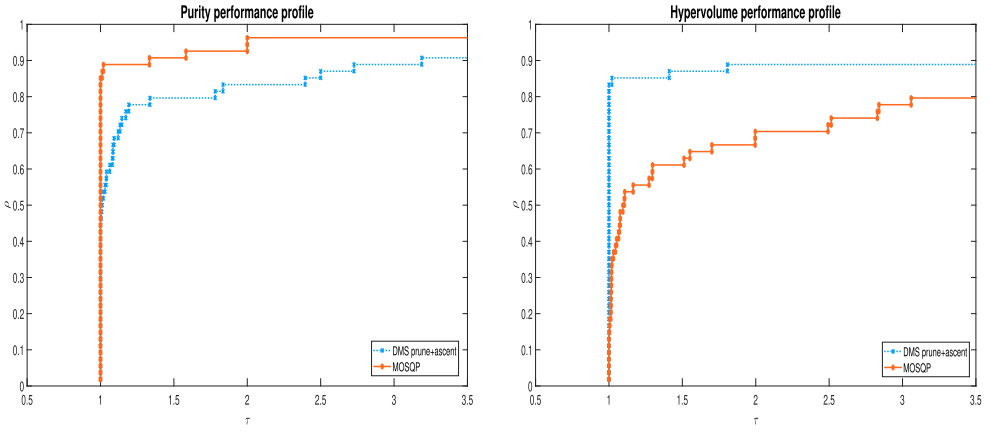

Figure 20. Performance profiles for purity and hypervolume metrics, comparing MOSQP and the new version of DMS, where poll directions are pruned using first-order information, but not at all the iterations (1 second of computational time).

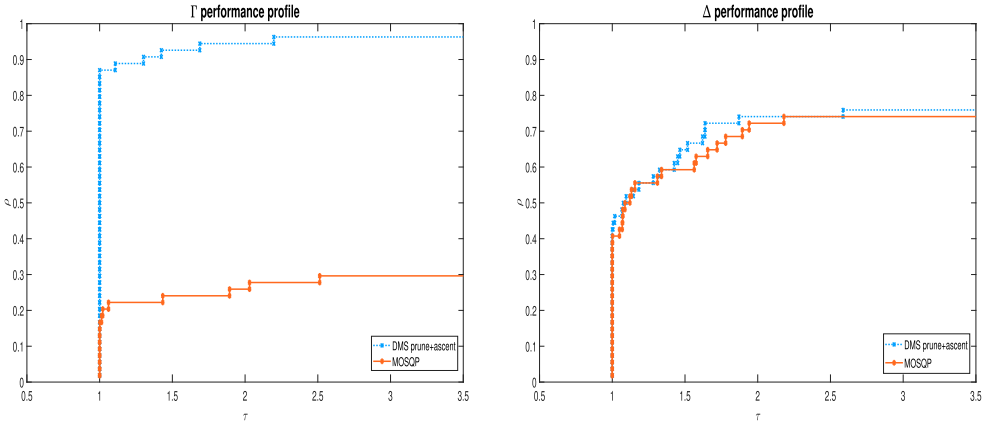

Figure 21. Performance profiles for Γ and Δ metrics, comparing MOSQP and the new version of DMS, where poll directions are pruned using first-order information, but not at all the iterations (1 second of computational time).

low computational budgets of function evaluations, it allows an increase in the percentage of nondominated points generated in the approximation to the Pareto front of the MOO problem and also a reduction in the largest gap across the generated Pareto front, when compared with the original implementation of DMS. In the case of MOSQP, there are additional advantages regarding the hypervolume associated to the computed approximation to the Pareto front. Nevertheless, in general, DMS variants are expected to require more computational time to solve a problem than MOSQP.

Future work could include the definition of a search step taking advantage of first-order information for building Taylor models, which will be minimized considering an approach similar to the one proposed and analyzed in Ref. [6].

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

Roberto Andreani was financially supported by (São Paulo Research Foundation) FAPESP (Projects 2013/05475-7 and 2017/18308-2) and CNPq (Project 301888/2017-5). Support for Ana Luísa Custódio was provided by national funds through FCT – Fundação para a Ciência e a Tecnologia I.P., under the scope of projects PTDC/MAT-APL/28400/2017, UIDB/00297/2020, and UIDP/00297/2020. Marcos Raydan was financially supported by the Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) through the projects UIDB/00297/2020, UIDP/00297/2020 (Centro de Matemática e Aplicações) and CEECIND/02211/2017.

Notes on contributors

Roberto Andreani graduated with a Bachelor's degree from the University of Buenos Aires, Argentina and obtained his PhD in Applied Mathematics in Campinas, Brazil. He was a visitor at University of Rice, Simón Bolívar University in Caracas, University of Buenos Aires, University of La Plata, NOVA School of Science and Technology, University of Coimbra, and University of Porto. He has published more than 60 articles in journals of Mathematics, Computer Science, Biology, among others.

Ana Luísa Custódio holds a PhD in Mathematics by NOVA School of Science and Technology, a MSc in Operations Research and Systems Engineering by Instituto Superior Técnico, and a BSc in Mathematics also by NOVA School of Science and Technology. Her main research area is Nonlinear Optimization, with a focus on Derivative-free Optimization. She is also interested in topics related to Multiobjective Optimization and Global Optimization. She is co-author of several computational codes, as a byproduct of the research conducted.

After obtaining the Bachelor degree and the Master degree in Caracas, Venezuela, ***Marcos Raydan*** finished the PhD in Computational and Applied Mathematics at Rice University (Houston, Texas), in 1991. After a post-doc in Lexington, Kentucky, he held academic positions at two different universities in Venezuela. For the last three years he has been a researcher at NOVA School of Science and Technology, Centre of Mathematics and Applications (CMA), Lisbon, Portugal. He has advised 12 PhD thesis and published over 80 papers and 3 books in scientific computing. His research interests include continuous optimization and numerical linear algebra, with a special focus on the development of algorithms for large-scale computations.

References

[1] M.A. Abramson, C. Audet, and J.E. Dennis, Jr., *Generalized pattern searches with derivative information*, Math. Program. Ser. B 100 (2004), pp. 3–25.
[2] H. Afshari, W. Hare, and S. Tesfamariam, *Constrained multi-objective optimization algorithms: Review and comparison with application in reinforced concrete structures*, Appl. Soft. Comput. 83 (2019), pp. 105631.
[3] M.A.T. Ansary and G. Panda, *A globally convergent SQCQP method for multiobjective optimization problems*, SIAM J. Optim. 31 (2021), pp. 91–113.
[4] C. Audet and W. Hare, *Derivative-Free and Blackbox Optimization*, Springer Series in Operations Research and Financial Engineering, Springer, Cham, 2017.
[5] C. Audet, J. Bigeon, D. Cartier, S. Le Digabel, and L. Salomon, *Performance indicators in multiobjective optimization*, Eur. J. Oper. Res. 292 (2021), pp. 397–422.
[6] C.P. Brás and A.L. Custódio, *On the use of polynomial models in multiobjective directional direct search*, Comput. Optim. Appl. 77 (2020), pp. 897–918.

[7] R.P. Brito, H. Sebastião, and P. Godinho, *Portfolio management with higher moments: the cardinality impact*, Int. Trans. Oper. Res. 26 (2019), pp. 2531–2560.
[8] D. Brockhoff and E. Zitzler, *Objective reduction in evolutionary multiobjective optimization: theory and applications*, Evol. Comput. 17 (2009), pp. 135–166.
[9] G.A. Carrizo, P.A. Lotito, and M.C. Maciel, *Trust region globalization strategy for the nonconvex unconstrained multiobjective optimization problem*, Math. Program. 159 (2016), pp. 339–369.
[10] S. Chand and M. Wagner, *Evolutionary many-objective optimization: a quick-start guide*, Surveys Oper. Res. Manage. Sci. 20 (2015), pp. 35–42.
[11] S. Cheng, Y. Shi, and Q. Qin, *On the performance metrics of multiobjective optimization*, in *Advances in Swarm Intelligence, ICSI 2012, LNCS*, Y. Tan, Y. Shi, and Z. Ji, eds., Vol. 7331, Springer, Berlin, 2012, pp. 504–512.
[12] F.H. Clarke, *Optimization and Nonsmooth Analysis*, SIAM, Philadelphia, PA, 1990.
[13] G. Cocchi, G. Liuzzi, A. Papini, and M. Sciandrone, *An implicit filtering algorithm for derivative-free multiobjective optimization with box constraints*, Comput. Optim. Appl. 69 (2018), pp. 267–296.
[14] G. Cocchi, G. Liuzzi, S. Lucidi, and M. Sciandrone, *On the convergence of steepest descent methods for multiobjective optimization*, Comput. Optim. Appl. 77 (2020), pp. 1–27.
[15] A.R. Conn, K. Scheinberg, and L.N. Vicente, *Introduction to Derivative-Free Optimization*, MPS-SIAM Series on Optimization, SIAM, Philadelphia, PA, 2009.
[16] A.L. Custódio, Y. Diouane, R. Garmanjani, and E. Riccietti, *Worst-case complexity bounds of directional direct-search for multiobjective derivative-free optimization*, J. Optim. Theory Appl. 188 (2021), pp. 73–93.
[17] A.L. Custódio and J.F.A. Madeira, *MultiGLODS: global and local multiobjective optimization using direct search*, J. Global Optim. 72 (2018), pp. 323–345.
[18] A.L. Custódio, J.F.A. Madeira, A.I.F. Vaz, and L.N. Vicente, *Direct multisearch for multiobjective optimization*, SIAM J. Optim. 21 (2011), pp. 1109–1140.
[19] I. Das and J.E. Dennis, *Normal-boundary intersection: a new method for generating the Pareto surface in nonlinear multicriteria optimization problems*, SIAM J. Optim. 8 (1998), pp. 631–657.
[20] C. Davis, *Theory of positive linear dependence*, Am. J. Math. 76 (1954), pp. 733–746.
[21] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, *A fast and elitist multiobjective genetic algorithm: NSGA-II*, IEEE Trans. Evol. Comput. 6 (2002), pp. 182–197.
[22] E.D. Dolan and J.J. Moré, *Benchmarking optimization software with performance profiles*, Math. Program. 91 (2002), pp. 201–213.
[23] G. Eichfelder, *Adaptive Scalarization Methods in Multiobjective Optimization*, Springer-Verlag, Berlin, 2008.
[24] M.T.M. Emmerich and A.H. Deutz, *A tutorial on multiobjective optimization: fundamentals and evolutionary methods*, Nat. Comput. 17 (2018), pp. 585–609.
[25] N.S. Fazzio and M.L. Schuverdt, *Convergence analysis of a nonmonotone projected gradient method for multiobjective optimization problems*, Optim. Lett. 13 (2019), pp. 1365–1379.
[26] J. Fliege and B.F. Svaiter, *Steepest descent methods for multicriteria optimization*, Math. Methods Oper. Res. 51 (2000), pp. 479–494.
[27] J. Fliege and A.I.F. Vaz, *A method for constrained multiobjective optimization based on SQP techniques*, SIAM J. Optim. 26 (2016), pp. 2091–2119.
[28] J. Fliege, L.M.G. Drummond, and B.F. Svaiter, *Newton's method for multiobjective optimization*, SIAM J. Optim. 20 (2009), pp. 602–626.
[29] C.M. Fonseca, L. Paquete, and M. López-Ibánez, *An improved dimension-sweep algorithm for the hypervolume indicator*, in *Proceedings of the 2006 Congress on Evolutionary Computation (CEC'06)*, Vancouver, Canada. IEEE, 2006, pp. 1157–1163.
[30] B. Gebken, S. Peitz, and M. Dellnitz, *On the hierarchical structure of Pareto critical sets*, J. Global Optim. 73 (2019), pp. 891–913.
[31] D. Hirpa, W. Hare, Y. Lucet, Y. Pushak, and S. Tesfamariam, *A bi-objective optimization framework for three-dimensional road alignment design*, Transport. Res. Part C: Emerg. Technol. 65 (2016), pp. 61–78.

[32] T.G. Kolda, R.M. Lewis, and V. Torczon, *Optimization by direct search: new perspectives on some classical and modern methods*, SIAM Rev. 45 (2003), pp. 385–482.
[33] G. Liuzzi, S. Lucidi, and F. Rinaldi, *A derivative-free approach to constrained multiobjective nonsmooth optimization*, SIAM J. Optim. 26 (2016), pp. 2744–2774.
[34] F. Maglia, F. Capra, M. Gatti, and E. Martelli, *Process selection, modelling and optimization of a water scrubbing process for energy-self-sufficient biogas upgrading plants*, Sustain. Energy Technol. Assess. 27 (2018), pp. 63–73.
[35] R.T. Marler and J.S. Arora, *Survey of multi-objective optimization methods for engineering*, Struct. Multidiscipl. Optim. 26 (2004), pp. 369–395.
[36] E. Miglierina, E. Molho, and M.C. Recchioni, *Box-constrained multi-objective optimization: a gradient-like method without "a priori" scalarization*, Eur. J. Oper. Res. 188 (2008), pp. 662–682.
[37] L.M.G. Drummond and A.N. Iusem, *A projected gradient method for vector optimization problems*, Comput. Optim. Appl. 28 (2004), pp. 5–29.
[38] P.S. Potrebko, J. Fiege, M. Biagioli, and J. Poleszczuk, *Investigating multi-objective fluence and beam orientation IMRT optimization*, Phys. Med. Biol. 62 (2017), pp. 5228–5244.
[39] A. Ravanbakhsh and S. Franchini, *Multiobjective optimization applied to structural sizing of low cost university-class microsatellite projects*, Acta Astronaut. 79 (2012), pp. 212–220.
[40] N. Riquelme, C.V. Lücken, and B. Baran, *Performance metrics in multi-objective optimization*, in *Latin American Computing Conference (CLEI)*, Arequipa, Peru. IEEE, 2015, pp. 1–11.
[41] D.K. Saxena, J.A. Duro, A. Tiwari, K. Deb, and Q. Zhang, *Objective reduction in many-objective optimization: linear and nonlinear algorithms*, IEEE Trans. Evol. Comput. 17 (2013), pp. 77–99.
[42] E. Zitzler and L. Thiele, *Multiobjective optimization using evolutionary algorithms – a comparative case study*, in *Parallel Problem Solving from Nature--PPSN V: 5th International Conference*, Amesterdam, A. E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, eds., Springer, Berlin, 1998, pp. 292–301.
[43] E. Zitzler, K. Deb, and L. Thiele, *Comparison of multiobjective evolutionary algorithms: empirical results*, Evol. Comput. 8 (2000), pp. 173–195.
[44] E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, and V.G. da Fonseca, *Performance assessment of multiobjective optimizers: an analysis and review*, IEEE Trans. Evol. Comput. 7 (2003), pp. 117–132.