ON THE USE OF POLYNOMIAL MODELS IN MULTIOBJECTIVE DIRECTIONAL DIRECT SEARCH

C. P. BRÁS* AND A. L. CUSTÓDIO †

Abstract. Polynomial interpolation or regression models are an important tool in Derivative-free Optimization, acting as surrogates of the real function. In this work, we propose the use of these models in the multiobjective framework of directional direct search, namely the one of Direct Multisearch. Previously evaluated points are used to build quadratic polynomial models, which are minimized in an attempt of generating nondominated points of the true function, defining a search step for the algorithm. Numerical results state the competitiveness of the proposed approach.

Key words. Multiobjective optimization, derivative-free optimization, direct search methods, quadratic polynomial interpolation and regression, minimum Frobenius norm models.

AMS subject classifications. 90C29, 90C56, 65D05, 90C30.

1. Introduction. Multiobjective optimization is a common topic in practical applications, when several objectives need to be optimized and are conflicting among each other [7, 33]. Applications appear on different domains such as engineering, finance, or medicine [1, 27, 36, 8, 34].

In this work we address the Multiobjective Derivative-free Optimization problem

$$\begin{aligned} \min \quad & F(x) \equiv (f_1(x), \ldots, f_m(x))^\top \\ \text{s.t.} \quad & x \in \Omega \subseteq \mathbb{R}^n, \end{aligned} \tag{1.1}$$

where $m \geq 2$, $\Omega \subseteq \mathbb{R}^n$ represents the feasible region and each $f_i : \Omega \subseteq \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}, i = 1, 2, \ldots, m$ denotes a component of the objective function, for which derivatives are not available, neither can be numerically approximated. Often the objective function is nonsmooth or the corresponding evaluation is expensive and/or unreliable, justifying the derivative-free approach. A comprehensive review of single objective derivative-free optimization methods can be found in [15, 3].

Several methods were already proposed for this class of problems. Some, like [6, 38], rely on the aggregation of the different components of the objective function, addressing the multiobjective derivative-free optimization problem through a series of single objective minimizations. In [6] it is used a directional direct search method for solving the single objective optimization problems, whereas [38] resorts to a derivative-free trust-region approach.

The works [13, 17, 32] follow a different strategy, making use of the concept of Pareto dominance, without considering any aggregation techniques. Directional direct search methods are generalized in [17] to multiobjective optimization, while [32] consists in a linesearch-based method and [13] can be regarded as a multiobjective version of implicit filtering.

Being a directional direct search method, the algorithmic structure of each iteration of [17] is organized in a search and a poll step. However, in [17] no particular

*Department of Mathematics, FCT-UNL-CMA, Campus de Caparica, 2829-516 Caparica, Portugal (mb@fct.unl.pt).

†Department of Mathematics, FCT-UNL-CMA, Campus de Caparica, 2829-516 Caparica, Portugal (alcustodio@fct.unl.pt).

Support for both authors 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 and UID/MAT/00297/2020.

strategy was proposed for the definition of a search step and the numerical experiments reported are the result of only applying the poll step of the algorithm. In [16], a multistart approach was proposed as a search step, jointly with a merging strategy, in an attempt of conferring some global behavior to the algorithm, through the ability of identifying different local Pareto fronts for the problem. The current work will be based on [17] and will propose the minimization of quadratic polynomial models, built for the different components of the objective function, in the search step of the algorithm, trying to improve its ability in generating approximations to the Pareto fronts (not necessarily the global one). Models computation will reuse previously evaluated points, at no additional cost in terms of objective function values. These models will be minimized on their own, in an attempt of moving to the extreme points of the Pareto front, or combined among them, trying to simultaneously improve the different components of the objective function.

The use of surrogate models was already considered in multiobjective derivative-free optimization, but not in directional direct search methods. Radial basis functions have been explored jointly with evolutionary strategies [2, 19, 37] and Gaussian processes have been proposed jointly with Monte-Carlo techniques [24]. However, these are heuristic approaches, without a well-established general convergence analysis. On the other hand, providing convergence guarantees, quadratic polynomial models have been used in trust-region derivative-free approaches to multiobjective optimization problems, considering scalarization techniques [38, 40]. Recently, the work [38], which respects to biobjective optimization problems, was extended to general multiobjective optimization [22] (without rigourously establishing the corresponding convergence analysis). In this latter case, models are built with Shepard's method [39] and MADS [5], a single objective directional direct search method, is used to minimize weighted sums of these models.

Polynomial interpolation and regression models have also been successfully used in the definition of a search step for single objective directional direct search [18, 14]. In fact, the minimization of these models is fully explored in trust-region methods for derivative-free optimization [15]. This class of algorithms is particularly efficient if the function to be minimized is smooth, even if derivatives are not available. The combination of the minimization of polynomial interpolation and regression models with directional direct search takes advantage of the strengths of both algorithmic classes, allowing the method to exploit curvature and proceed, if there is enough smoothness associated to the problem, or use the good geometrical properties of positive spanning sets [20], when the search step based on the polynomial models fails in generating a new successful point.

The paper is organized as follows. Section 2 will revise Direct Multisearch, both in terms of algorithmic description and convergence analysis. Quadratic polynomial models are introduced in Section 3, stating the corresponding quality as approximations of smooth functions. The strategy proposed for its joint use with directional direct search for multiobjective optimization will be fully detailed in Section 4. Section 5 reports the numerical experiments that support the validity of the proposed search step. The paper ends in Section 6 with some conclusions.

2. Direct Multisearch. Direct Multisearch (DMS) was proposed in [17] as a new class of methods, generalizing directional direct search to multiobjective derivative-free optimization. Since then, it has been consistently used with good results both for benchmark of new solvers [13, 32] or in real applications [9, 28].

One of the main advantages of DMS is not considering any aggregation function

for the different components of the objective function, avoiding the definition of all parameters inherent to these strategies [33]. In fact, the algorithm solely relies in the concept of Pareto dominance to accept new points. Making use of the strict partial order induced by the cone $\mathbb{R}^m_+$, we say that point x dominates point y when $F(x) \prec_F F(y)$, i.e., when $F(y) - F(x) \in \mathbb{R}^m_+ \setminus \{0\}$.

In DMS two globalization strategies can be adopted [17]. In the case of a globalization strategy based on integer lattices, an iteration of the algorithm will be declared as successful when at least one new feasible nondominated point is found, which will be added to a list that represents the current approximation to the Pareto front of the problem. This list, which only includes feasible nondominated points, is updated every time that a new feasible nondominated point is found, by adding this point to it and removing from it all dominated points resulting from this adding. When globalization requires sufficient decrease for accepting new feasible points, a forcing function $\rho : (0, +\infty) \to (0, +\infty)$, i.e., a continuous and nondecreasing function, satisfying $\rho(t)/t \to 0$ when $t \downarrow 0$ (see [30]) will be used. Typical examples of forcing functions are $\rho(t) = t^{1+c}$, for $c > 0$. Let $D(L) \subset \mathbb{R}^m$ be the image of the set of points dominated by the list of points L and let $D(L; a)$ be the set of points whose distance in the ℓ_∞ norm to $D(L)$ is no larger than $a > 0$. In a simplified way, DMS declares an iteration as successful if a new point x is found such that $F(x) \notin D(L; \rho(\alpha))$, where α represents a stepsize parameter associated with the current iteration.

DMS addresses constraints using an extreme barrier approach [4], only evaluating the objective function at feasible points. Each iteration of the algorithm starts with the selection of an iterate point and the corresponding stepsize parameter, from the current list of previously evaluated feasible nondominated points. Different strategies can be considered to order this list of points, resulting in different algorithmic variants. In the current numerical implementation of DMS a spread metric is used for ordering, in an attempt of reducing the gaps between consecutive points lying in the current approximation to the Pareto front of the problem.

After selecting the iterate point, the algorithm performs a search step and, in case of failure of adding a new point to the list, a poll step around the current iterate point. The search step is not detailed in the original description of the algorithm, neither it is implemented in the corresponding numerical solver. The major requirement of this step is to be finite (and restricted to an implicit mesh, when using a globalization strategy based on integer lattices).

The poll step corresponds to the evaluation of the objective function along a set of directions scaled by a stepsize parameter. Different requirements should be satisfied by these poll directions, depending on the level of smoothness of the objective function and on the geometry of the feasible region around the poll center. Typically, positive spanning sets are considered [20]. Polling can be complete, meaning that all feasible poll points will be evaluated, or opportunistic, in which case sampling is stopped as soon as a success is declared, meaning that a new point was added to the list.

At the end of each iteration the stepsize is updated, by increasing it or by maintaining it constant for successful iterations and by decreasing it for unsuccessful ones.

Algorithm 1 provides a schematic, but simplified description of the DMS algorithm (for a complete description, see [17]).

DMS has a well-established convergence analysis [17], which follows the general lines of any other directional direct search method. For any of the two globalization strategies considered, it is possible to establish the existence of a subsequence of

Algorithm 1: A simplified description of Direct Multisearch (DMS).

Initialization

Choose $x_0 \in \Omega$ with $f_i(x_0) < +\infty, \forall i \in \{1, 2, \ldots, m\}$, $\alpha_0 > 0$ an initial stepsize, $0 < \beta_1 \leq \beta_2 < 1$ 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 $L_0 = \{(x_0; \alpha_0)\}$.

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. **Search step:** Compute a finite set of points $\{z_s\}_{s \in S}$ (lying in an implicit mesh if the globalization strategy is based on the use of integer lattices) and evaluate F at each feasible point in S. Compute L_{trial} by removing all dominated points (possibly using sufficient decrease, depending on the globalization strategy considered) from $L_k \cup \{(z_s; \alpha_k) : s \in S \wedge z_s \in \Omega\}$. If $L_{trial} \neq L_k$ declare the iteration (and the search step) successful, set $L_{k+1} = L_{trial}$, and skip the poll step.
3. **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 (possibly using sufficient decrease, depending on the globalization strategy considered) 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 $L_{k+1} = L_k$.
4. **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 $(x_k + \alpha_k d; \alpha_{k,new})$, when success is coming from the poll step, or $(z_s; \alpha_k)$ in L_{k+1} by $(z_s; \alpha_{k,new})$, when success is coming from the search. 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 $(x_k; \alpha_{k,new})$.

stepsize parameters converging to zero. For this, some assumptions are required, concerning the objective function and the feasible region.

ASSUMPTION 2.1. *The set* $\{x \in \Omega : F(x) \not\leq D(\{x_0\})\}$ *is compact.*

ASSUMPTION 2.2. *The function* F *is bounded in* $\{x \in \Omega : F(x) \not\leq D(\{x_0\})\}$.

The level of smoothness of the objective function imposes additional conditions on the poll directions. For continuously differentiable functions, Assumption 2.3 suffices, whereas the presence of nonsmoothness requires Assumption 2.4.

ASSUMPTION 2.3. *The set* $\mathcal{D} = D$ *of positive spanning sets is finite and the elements of* D *are of the form* $G\bar{z}_j$, $j = 1, \ldots, |D|$, *where* $G \in \mathbb{R}^{n \times n}$ *is a nonsingular*

matrix and each $\bar{z}_j$ is a vector in $\mathbb{Z}^n$.

ASSUMPTION 2.4. *Let D represent a finite set of positive spanning sets satisfying Assumption 2.3.*

The set $\mathcal{D}$ is so that the elements $d_k \in D_k \in \mathcal{D}$ satisfy the following conditions:

1. *d_k is a nonnegative integer combination of the columns of D.*
2. *The distance between x_k and the point $x_k + \alpha_k d_k$ tends to zero if and only if α_k does:*

$$\lim_{k \in K} \alpha_k \|d_k\| \ = \ 0 \iff \lim_{k \in K} \alpha_k \ = \ 0,$$

   for any infinite subsequence K.
3. *The limits of all convergent subsequences of $\bar{D}_k = \{d_k/\|d_k\| : \ d_k \in D_k\}$ are positive spanning sets for $\mathbb{R}^n$.*

The third requirement above is included for consistency with the smooth case and because it is part of the Mesh Adaptive Direct Search (MADS) original presentation [5].

The type of globalization strategy considered also imposes different requirements for convergence. In fact, when using sufficient decrease to accept new points the previous assumption simplifies to Assumption 2.5.

ASSUMPTION 2.5. *The distance between x_k and the point $x_k + \alpha_k d_k$ tends to zero if and only if α_k does:*

$$\lim_{k \in K} \alpha_k \|d_k\| \ = \ 0 \iff \lim_{k \in K} \alpha_k \ = \ 0,$$

for all $d_k \in D_k$ and for any infinite subsequence K.

If the globalization strategy is based on the use of integer lattices, strict rules need to be followed for updating the stepsize parameter (see Assumption 2.6) and for generating the points evaluated at the search step, which need to lie in an implicit mesh (see Assumption 2.7).

ASSUMPTION 2.6. *Let $\tau > 1$ be a rational number and $m^{max} \geq 0$ and $m^{min} \leq -1$ integers. If the iteration is successful, then the stepsize parameter is maintained or increased by considering $\alpha_{new} = \tau^{m^+}\alpha$, with $m^+ \in \{0, \ldots, m^{max}\}$. If the iteration is unsuccessful, then the stepsize parameter is decreased by setting $\alpha_{new} = \tau^{m^-}\alpha$, with $m^- \in \{m^{min}, \ldots, -1\}$.*

ASSUMPTION 2.7. *At iteration k, the search step in Algorithm 1 only evaluates points in*

$$M_k \ = \ \bigcup_{x \in E_k} \{x + \alpha_k D z : z \in \mathbb{N}_0^{|D|}\},$$

where E_k represents the set of all points evaluated by the algorithm previously to iteration k.

The first key result for establishing convergence can now be formalized in Theorem 2.1, for both globalization strategies.

THEOREM 2.1. *(see [17]) Let Assumption 2.1 hold and consider a globalization strategy based on integer lattices, thus under one of the Assumptions 2.3 or 2.4 combined with Assumptions 2.6–2.7. Alternatively, consider a globalization strategy based on imposing a sufficient decrease condition and let Assumptions 2.2, 2.5 hold.*

Algorithm 1 generates a sequence of iterates satisfying

$$\liminf_{k \to +\infty} \alpha_k \ = \ 0.$$

Assumption 2.1 and the previous result ensure the existence of a convergent refining subsequence of iterates.

DEFINITION 2.2. *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.*

Refining directions are associated with convergent refining subsequences and are the keystone to establish the convergence of the algorithm.

DEFINITION 2.3. *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_*.*

In fact, the convergence of DMS is established for limit points of convergent refining subsequences, assuming the density of the set of refining directions in the Clarke tangent cone to Ω computed at these limit points [12], which is typically accomplished by requiring the density in the unit sphere of the associated sets of refining directions.

DEFINITION 2.4. *A vector $d \in \mathbb{R}^n$ is said to be a Clarke tangent vector to the set $\Omega \subset \mathbb{R}^n$ at the point x_* in the closure of Ω if for every sequence $\{y_k\}$ of elements of Ω that converges to x_* and for every sequence of positive real numbers $\{t_k\}$ converging to zero, there exists a sequence of vectors $\{w_k\}$ converging to d such that $y_k + t_k w_k \in \Omega$.*

The Clarke tangent cone to Ω at x_* ($T_\Omega^{Cl}(x_*)$) is defined as the set of all Clarke tangent vectors to Ω at x_*. Its interior is denoted as the hypertangent cone to Ω at x_*.

THEOREM 2.5. *(see [17]) Consider a refining subsequence $\{x_k\}_{k\in K}$ converging to $x_* \in \Omega$. Assume that F is Lipschitz continuous near x_* and that the hypertangent 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 critical point, meaning,*

$$\forall d \in T_\Omega^{Cl}(x_*), \exists j(d) \in \{1, 2, \ldots, m\} : f_{j(d)}^\circ(x_*; d) \geq 0.$$

If, in addition, F is strictly differentiable at x_*, then this point is a Pareto-Clarke-KKT critical point, i.e,

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

For a detailed analysis of the convergence results, using Clarke-Jahn calculus [29], we refer the reader to [17]. We point out that the convergence results rely on the algorithmic behavior at unsuccessful poll steps and hold independently of the initialization, of the strategy adopted for selecting an iterate point, of the type of polling strategy considered (complete or opportunistic) or of the definition of a search step. The latter will be the subject of the current work and for that we will use quadratic polynomial models.

3. Polynomial Interpolation and Regression Models. In single objective derivative-free optimization, polynomial interpolation models have been considered to define derivative-free counterparts of trust-region methods. The best implementations of algorithms belonging to this class include DFO [15] and NEWUOA [35].

Consider a quadratic polynomial model centered at the point x_k:

$$m(x) = f(x_k) + g_k^\top (x - x_k) + \frac{1}{2}(x - x_k)^\top H_k (x - x_k).$$

The gradient g_k and the symmetric Hessian matrix H_k are determined by solving the linear system corresponding to the interpolation conditions

$$m(y^i) = f(y^i),\ i = 1, \ldots, p, \tag{3.1}$$

where each y^i represents a sample point for which the objective function has been previously evaluated. The computation of a complete quadratic model requires a sampling set of size $q = \frac{(n+1)(n+2)}{2}$, corresponding to the solution of a determined linear system. However, when $p < q$ a minimum norm solution could be computed, whereas, if $p > q$, regression techniques could be considered.

An important feature of underdetermined models is the balance between the number of points required for the corresponding computation (typically strictly between $n + 1$ and $(n + 1)(n + 2)/2$) and the ability of incorporating curvature information. The following result [15, Theorem 5.4][18] provides a general error bound for underdetermined quadratic polynomial interpolation models, built from Λ–poised sampling sets (for a connection between this geometric condition and the condition number of the corresponding linear system see [15, Theorem 3.14]).

THEOREM 3.1. *Let f be a continuously differentiable function with a Lipschitz continuous gradient (and Lipschitz constant $C_{\nabla f} > 0$) in the ball*

$$B(x_k; \Delta_k) = \{x \in \mathbb{R}^n : \|x - x_k\| < \Delta_k\}.$$

If the sample set $Y_k = \{y^1, \ldots, y^p\}$, with $n + 1 < p < \frac{(n+1)(n+2)}{2}$, is Λ_L–poised for linear interpolation or linear regression then

$$\|\nabla f(y) - \nabla m(y)\| \ \leq\ C_p \Lambda_L \left(C_{\nabla f} + \|H_k\|\right) \Delta_k, \qquad \forall y \in B(x_k; \Delta_k)$$

$$|f(y) - m(y)| \ \leq\ \left(C_p \Lambda_L + \frac{1}{2}\right) \left(C_{\nabla f} + \|H_k\|\right) \Delta_k^2, \qquad \forall y \in B(x_k; \Delta_k)$$

where H_k represents the Hessian matrix of the model and C_p is a positive constant dependent on p.

Thus, a possible approach to build an underdetermined quadratic interpolation model would be to minimize the entries of the Hessian matrix (considering the Frobenius norm), subject to the interpolation conditions:

$$\begin{aligned} \min \quad & \tfrac{1}{4}\|H_k\|_F^2 \\ \text{s.t.} \quad & f(x_k) + g_k^\top (y^i - x_k) + \tfrac{1}{2}(y^i - x_k)^\top H_k (y^i - x_k) \ =\ f(y^i), \quad i = 1, \ldots, p. \end{aligned} \tag{3.2}$$

This approach generates the so-called Minimum Frobenius Norm (MFN) models.

Requiring more about the geometry of the sample set, namely that it is Λ-poised in the minimum Frobenius norm sense (see [15, Definition 5.6]), it is possible to show that the Hessian matrices of these models are bounded, as stated in [15, Theorem 5.7].

THEOREM 3.2. *Let f be a continuously differentiable function with a Lipschitz continuous gradient (and Lipschitz constant $C_{\nabla f} > 0$) in the ball $B(x_k; \Delta_k)$. If the sample set Y_k is Λ_F–poised in the minimum Frobenius norm sense then*

$$\|H_k\| \ \leq\ C_{p,q} C_{\nabla f} \Lambda_F,$$

where H_k represents the Hessian matrix of the model and $C_{p,q}$ is a positive constant depending on p and q.

These two results yield the following error bounds for MFN models:

$$\|\nabla f(y) - \nabla m(y)\| \;\leq\; C_p \Lambda_L C_{\nabla f}\,(1 + C_{p,q}\Lambda_F)\;\Delta_k, \quad \forall y \in B(x_k;\Delta_k)$$

$$|f(y) - m(y)| \;\leq\; \left(C_p\Lambda_L + \frac{1}{2}\right) C_{\nabla f}\;(1 + C_{p,q}\Lambda_F)\,\Delta_k^2, \quad \forall y \in B(x_k;\Delta_k),$$

which allow us to conclude that MFN models correctly reproduce the accuracy of first-order Taylor models.

In [18] MFN models were successfully used to define a search step for single objective directional direct search, by reusing previously evaluated points. In fact, whenever more than $n+1$ points have been evaluated, a MFN quadratic model was computed and minimized in a trust-region,

$$B(x_k;\Delta_k) = \{x \in \mathbb{R}^n : \|x - x_k\| \leq \Delta_k\},$$

whose size was directly related to the stepsize parameter:

$$\Delta_k \;=\; \sigma_k\;\alpha_{k-1}\;\max_{d\in D_{k-1}} \|d\|, \tag{3.3}$$

where D_{k-1} was the set of poll vectors considered in the last iteration and σ_k took the value 1 if the previous iteration was unsuccessful, or 2 otherwise. This ensured the quality of the computed models, since a subsequence of the stepsize parameters converges to zero.

Eventually, the number of available points, where the objective function has already been evaluated, will exceed $p = \frac{(n+1)(n+2)}{2}$. In [18] two strategies have been proposed to address this situation: discard some of the points and always compute a determined quadratic interpolation model (with exactly $(n+1)(n+2)/2$ points) or use regression quadratic models.

In fact, when $p > (n+1)(n+2)/2$, [15, Theorem 4.13] establishes the quality of regression models as approximation to the real function. Once more, the geometry of the sample set is ensured by a Λ-poisedness condition (see [15, Definition 4.7]).

THEOREM 3.3. *Let f be a twice continuously differentiable function with a Lipschitz continuous Hessian (and Lipschitz constant $C_{\nabla^2 f} > 0$) in the ball $B(x_k;\Delta_k)$. If the sample set $Y_k = \{y^1,\ldots,y^p\}$, with $p > \frac{(n+1)(n+2)}{2}$ is Λ_R–poised for quadratic regression then*

$$\|\nabla^2 f(y) - \nabla^2 m(y)\| \;\leq\; C_p^1 \Lambda_R\, C_{\nabla^2 f}\,\Delta_k, \qquad \forall y \in B(x_k;\Delta_k)$$

$$\|\nabla f(y) - \nabla m(y)\| \;\leq\; C_p^2 \Lambda_R C_{\nabla^2 f}\,\Delta_k^2, \qquad \forall y \in B(x_k;\Delta_k)$$

$$|f(y) - m(y)| \;\leq\; \left(C_p^3 \Lambda_R C_{\nabla^2 f} + \frac{C_{\nabla^2 f}}{6}\right)\Delta_k^3, \qquad \forall y \in B(x_k;\Delta_k)$$

where $C_p^i, i \in \{1,2,3\}$ are positive constants dependent on p.

According to the computational experience reported in [18], the regression variant is recommended, but a maximum of $(n+1)(n+2)$ points is allowed in model computation. Whenever there are more points than the maximum number allowed for building the model, the authors select 80% of the desired points from the ones nearest to the current iterate, and the remaining 20% as the ones farthest, in an attempt of preserving geometry and diversifying the information.

4. Defining a search step in DMS based on quadratic polynomial models. In this section, we extend the approach proposed in [18] to multiobjective directional direct search, by defining a search step in DMS. As in [18], previous evaluations of the objective function will be used to build quadratic polynomial models for each of its components $f_i, i = 1, 2, \ldots, m$, which will be minimized inside a defined trust-region, now in an attempt of finding new nondominated points.

At each iteration, the quadratic polynomial models are centered at the iterate point x_k, selected at the beginning of the iteration (see Step 1 of Algorithm 1). A cache of previously evaluated points (not necessarily nondominated), denoted by L_{cache}, is maintained and used for model computation. Points are selected in $B(x_k; \theta\Delta_k)$, with $\theta \geq 1$ and $\Delta_k = \sigma_k \alpha_k \max_{d \in D_{k-1}} \|d\|$, with $\sigma_k = \frac{1}{\beta_1}$. On one hand, the inclusion of the parameter $\theta \geq 1$ allows a more diversification of points, again in an attempt of conferring quality to the models. On the other hand, $\sigma_k = \frac{1}{\beta_1}$ ensures that after each unsuccessful poll step, considering the update rule for the stepsize, when selecting the same poll center in the new iteration, at least minimum Frobenious norm models will be built, since all the feasible evaluated points at the unsuccessful poll step will be in $B(x_k; \Delta_k)$.

Let m_i define the quadratic polynomial model centered at x_k, corresponding to the objective function component f_i, built as defined in the previous section:

$$m_i(x) = f_i(x_k) + {g_k^i}^\top (x - x_k) + \frac{1}{2}(x - x_k)^\top H_k^i (x - x_k), \ i = 1, 2, \ldots, m. \tag{4.1}$$

We note that m_i could be a determined, an underdetermined MFN model, or an overdetermined model built using regression techniques, depending on the number of points available for model computation.

Similarly to the single objective case, where the quadratic polynomial model is minimized in an attempt of finding a point with a better objective function value, this idea can be simply applied to the multiobjective context by independently minimizing each model m_i, in the considered trust-region $B(x_k; \Delta_k)$. However, it seems also natural that the simultaneous minimization of a combination of the obtained models for the different components f_i of the objective function should also be considered, in an attempt of finding new nondominated points.

The combined minimization of a subset of components of the objective function can be stated in the following optimization problem:

$$\begin{array}{ll} \min & \zeta \\ \text{s.t.} & m_i(x) \ \leq \ \zeta, \quad i \in I \\ & \|x - x_k\| \ \leq \ \Delta_k \\ & x \in \Omega, \end{array} \tag{4.2}$$

where $I \subseteq \{1, 2, \ldots, m\}$.

This formulation corresponds to a weighted Chebyshev norm scalarization, for the multiobjective minimization problem defined using the models of the different components of the objective function. The Chebyshev norm approach produces weekly efficient solutions for convex and nonconvex multiobjective optimization problems, which could be the case of our models. Nevertheless, other approaches could be taken for solving the derivative-based multiobjective optimization problem related to the joint minimization of the models for the different components of the objective function [41, 25, 10].

The number of components of the objective function to be jointly minimized depends on the selection of the subset I. In fact, there are $C_l^m = \frac{m!}{l!(m-l)!}$ possibilities for each set of cardinality $1 \leq l \leq m$. Therefore, different strategies can be defined for this selection. Algorithm 2 details the proposed scheme, for a fixed iteration k.

Algorithm 2: Defining a **Search step** for DMS.

Initialization

Select a set of points from L_{cache} (according to the defined criteria to build the quadratic polynomial models m_i). Set $l = 0$.

For $i = 1, 2, \ldots, m$

Build the quadratic polynomial model corresponding to the component of the objective function f_i.

While $l < m$

Set $l := l + 1$, define J the set of all combinations of l quadratic polynomial models taken from the total set of m models and set $S = \emptyset$.

For $j = 1, 2, \ldots, |J|$

Compute the point s_j, solution of problem (4.2) considering I as the set composed by the polynomial models corresponding to combination j. Update $S = S \cup \{s_j\}$.

Check for success

Evaluate F at each point in S and update L_{cache}. Compute L_{trial} by removing all dominated points (possibly using sufficient decrease, depending on the globalization strategy considered) from $L_k \cup \{(z_s; \alpha_k) : s \in S\}$. If $L_{trial} \neq L_k$ set $L_{k+1} = L_{trial}$, stop the cycle loop **while**, declare the search step successful and skip the poll step.

The procedure initiates with the selection of a set of previously evaluated points from the cache, inside $B(x_k; \theta\Delta_k)$, used for computing the m quadratic polynomial models. The joint minimization of combinations of this set of models will be considered. Several strategies can be defined for it (and have been attempted).

Starting with the exploration of the lower level of combinations ($l = 1$), corresponding to the individual minimization of the m models, potentiates the computation of extreme points of the Pareto front, contributing to a better spread of solutions. The level l is increased, the set of C_l^m combinations of l models is considered, the corresponding minimization is performed, and finally the condition for success is tested. The process is interrupted once that a given level of combinations of models generates a new nondominated point or when level $l = m$ has been reached, corresponding to the joint minimization of the m models. Thus, an unsuccessful search step will correspond to a total of $2^m - 1$ new function evaluations, which is acceptable, since m is usually relatively small.

As mentioned in Section 2, the convergence results of DMS hold independently of the definition of a particular search step. Therefore, we can claim that at least one limit point of the sequence of iterates generated by the proposed algorithm is a Pareto-Clarke critical point.

5. Numerical Testing. The code was implemented in MATLAB and is freely available (under a GNU Lesser General Public License) at

`http://ferrari.dmat.fct.unl.pt/personal/alcustodio/BoostDFO.htm`.

Since the proposed strategy respects to the definition of a search step for DMS, we

considered all the default options of this code [17]. In particular, coordinate directions are used for complete polling. For distinguishing the two versions, we will name our proposal as BoostDMS.

Several strategies were defined and tested to select the points from L_{cache} to build the quadratic polynomial models used in the search step (a maximum of $(n+1)(n+2)$ points is allowed). These included the size of the radius $\theta\Delta_k$ of the ball centered at x_k where the points are considered and the percentage of points that should be chosen from this ball. According to our computational results, 80% of the points are selected in $B(x_k; 3\Delta_k)$. The remaining 20% lie outside this ball, conferring diversity to the quadratic models.

For minimizing the different models, two solvers were considered, depending on the level l of combinations of the objective function components. When $1 < l \leq m$, function `fmincon.m` from MATLAB was used to solve problem (4.2). As initialization we have provided the feasible point $[x_k \ \zeta]^\top$, with $\zeta = \max\{f_i(x_k) : i \in I\}$. If $l = 1$, corresponding to the individual minimization of each model, function `trust.m` from MATLAB was used, providing x_k as initialization, for coherency with what was done in the single objective case [18]. In this last situation, the minimization only considers the constraint representing the trust region. The resulting point is projected on the hyper-rectangle defined by any bounds, and feasibility is checked for the remaining constraints defining the feasible region.

As test set, we used the collection of 100 bound constrained multiobjective optimization problems, where the number of variables, n, is an integer between 1 and 30, and with a number of components of the objective function $m = 2, 3$, or 4, available at `http://www.mat.uc.pt/dms`.

The proposed algorithm was compared with the original version of direct multisearch DMS [17], which does not consider any particular definition of a search step, and with the recent implicit filtering algorithm MOIF [13] proposed for multiobjective derivative-free optimization with box constraints. For these two algorithms, all the parameters have been set to the default values, which correspond to the best performance, according to [13] and [17]. In particular, MOIF algorithm was initialized with the barycenter of the problem bounds, whereas DMS and BoostDMS were initialized with n equally spaced points on the line connecting the lower and upper bounds of each problem (the variant referenced as DMS(n,line) in [17]).

All results described in this section have been obtained by allowing a maximum of 20 000 function evaluations or a minimum value of 10^{-3} for the stepsize h_k in MOIF (see [13]) or for the stepsize parameters α_k for all the points in the list, in the case of DMS or BoostDMS.

5.1. Metrics and profiles used for solver comparison. Solvers comparison uses purity and spread metrics, as defined in [17], and also the hypervolume indicator [42, 43].

The purity metric measures the quality of the Pareto front computed by a given solver in terms of capability of generating nondominated points. Let $\mathcal{S}$ be the set of solvers and $\mathcal{P}$ be the set of problems to be tested. Let $F_{p,s}$ denote the approximation to the Pareto front computed for problem $p \in \mathcal{P}$ by solver $s \in \mathcal{S}$. Let F_p also denote an approximation to the true Pareto front of problem p, now calculated by first forming $\bigcup_{s\in\mathcal{S}} F_{p,s}$ and then removing from this set any dominated point. The purity metric, for problem $p \in \mathcal{P}$ and solver $s \in \mathcal{S}$, is defined by the ratio $\bar{t}_{p,s} = \frac{|F_{p,s} \cap F_p|}{|F_{p,s}|}$, which takes values between zero and one. Higher values of $\bar{t}_{p,s}$ indicate a better Pareto front in terms of the percentage of nondominated points.

However, the purity metric is not enough to access the quality of an approximation to the Pareto front of a given problem, since points could be concentrated in a given region. Spread metrics try to quantify how well distributed are the nondominated points across the Pareto front. Since we are interested in computing the complete Pareto front, spread metrics need to first consider the computation of 'extreme points' in the objective function space $\mathbb{R}^m$ (for a detailed explanation on how to compute these 'extreme points' see [17]). The spread Γ attempts to measure the maximum size of the 'holes' of an approximated Pareto front. Let us assume that solver $s \in \mathcal{S}$ has computed, for problem $p \in \mathcal{P}$, an approximated Pareto front with N points, indexed by $1, 2, \ldots, N$, to which we add the 'extreme points' mentioned above and indexed by 0 and $N+1$. The metric $\Gamma_{p,s} > 0$ for problem $p \in \mathcal{P}$ and solver $s \in \mathcal{S}$ is given by

$$\Gamma_{p,s} \;=\; \max_{j \in \{1,\ldots,m\}} \left(\max_{i \in \{0,\ldots,N\}} \{\delta_{i,j}\} \right), \tag{5.1}$$

where $\delta_{i,j} = (f_j(x_{i+1}) - f_j(x_i))$ (assuming that the objective function values have been sorted by increasing order for each objective j).

The second spread metric (proposed in [21] for $m = 2$) measures how well the points are distributed in the computed Pareto front. The concept was generalized in [17] for higher dimensional objective spaces and its computation, for problem $p \in \mathcal{P}$ and solver $s \in \mathcal{S}$, is given by:

$$\Delta_{p,s} \;=\; \max_{j \in \{1,\ldots,m\}} \left(\frac{\delta_{0,j} + \delta_{N,j} + \sum_{i=1}^{N-1} |\delta_{i,j} - \bar{\delta}_j|}{\delta_{0,j} + \delta_{N,j} + (N-1)\bar{\delta}_j} \right), \tag{5.2}$$

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

Finally, the hypervolume indicator is also commonly used to measure the quality of different Pareto fronts, due to its favorable properties [43]. Given a reference point $U_p \in \mathbb{R}^m$, that is dominated by all points belonging to the approximations computed for the Pareto front of a given problem $p \in \mathcal{P}$, the hypervolume indicator 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 is limited by the upper corner U_p (for two-objective minimization problems, the hypervolume can be seen as the area of the surface that lies above the Pareto front and below the reference point U_p). Therefore, the hypervolume indicator for some approximation to the Pareto front $F_{p,s}$ of problem $p \in \mathcal{P}$, computed by solver $s \in \mathcal{S}$, considering the reference point $U_p \in \mathbb{R}^m$ that is dominated by all the points in $F_{p,s}$ is defined as:

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

where $Vol(.)$ denotes the Lebesgue measure of a m-dimensional set of points and $[x, U_p]$ denotes the interval box with lower corner x and upper corner U_p.

The above definition clearly indicates that the rigorous computation of the hypervolume is time-consuming. In fact, it is well-known that there is no algorithm to compute the hypervolume indicator in a time polynomial in the number of components of the objective function [11, 31]. In this work, for its computation we considered the approach proposed in [26]. The reference point $U_p \in \mathbb{R}^m$, for problem $p \in \mathcal{P}$, was selected as a point that is dominated by all points belonging to the different approximations to the Pareto front of problem $p \in \mathcal{P}$, computed by each of the solvers.

Coordinate j of U_p, $u_j^p \in \mathbb{R},\ j = 1, 2, \ldots, m$, is defined as:

$$u_j^p = \max \left\{ f_j(x) : x \in \bigcup_{s \in \mathcal{S}} F_{p,s} \right\},$$

where again $F_{p,s}$ represents the approximation to the Pareto front of problem $p \in \mathcal{P}$, computed by solver $s \in \mathcal{S}$. Moreover, the hypervolume value, computed for the Pareto front $F_{p,s}$ and denoted by $HV_{p,s}$, was scaled, according to the following procedure. For $j = 1, 2, \ldots, m$, let

$$l_j^p = \min \left\{ f_j(x) : x \in \bigcup_{s \in \mathcal{S}} F_{p,s} \right\}$$

and

$$r_j^p = u_j^p - l_j^p.$$

We define $HV_{p,s} := \frac{HV_{p,s}}{\Pi_{j=1}^m r_j^p}$, which means that all the values of $HV_{p,s}$ will lie in $[0, 1]$.

Results on the different metrics will be reported using performance profiles [23]. Performance profiles are depicted by a plot of a cumulative distribution function $\rho(\tau)$, representing a performance ratio for the different solvers. Let $t_{p,s}$ denote the performance on the problem $p \in \mathcal{P}$ of the solver $s \in \mathcal{S}$, assuming that lower values of $t_{p,s}$ indicate better performance. The performance ratio is defined by

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

with $r_{p,s} = t_{p,s} / \min\{t_{p,\bar{s}} : \bar{s} \in \mathcal{S}\}$. Thus, the value of $\rho_s(1)$ represents the probability of the solver s winning over the remaining ones. On the other hand, solvers with the largest probabilities $\rho_s(\tau)$ for large values of τ are the most robust (the ones that solve the largest number of problems in $\mathcal{P}$).

Since for purity and hypervolume larger values indicate better performance, when computing performance profiles for these two metrics, we set $t_{p,s} = 1/t_{p,s}$ as proposed in [17].

5.2. Numerical Results. Figures 5.1 and 5.2 report the comparison between BoostDMS and DMS algorithms, using performance profiles and the four metrics considered. BoostDMS clearly outperforms DMS in terms of efficiency for purity, hypervolume, and Γ metric. The results are very close in terms of robustness, with advantage of BoostDMS for purity.

Regarding the comparison between MOIF and BoostDMS, Figures 5.3 and 5.4 depict the corresponding performance profiles. The conclusions are similar to the previous ones. BoostDMS outperforms MOIF in terms of efficiency for any of the four metrics considered. In fact, reading the values of the curves for $\tau = 1$, we can observe that BoostDMS is able to attain the best purity, hypervolume, and spread Γ for approximately more 20% of problems than MOIF. This difference is of 15% in case of Δ metric. Regarding robustness, the results are similar for the two solvers, with exception of purity, where BoostDMS again presents a better performance than MOIF.

The reported results show that the definition of a search step based on quadratic polynomial models can be helpful in identifying nondominated points, allowing us to conclude that BoostDMS is an interesting alternative to both MOIF and DMS algorithms.

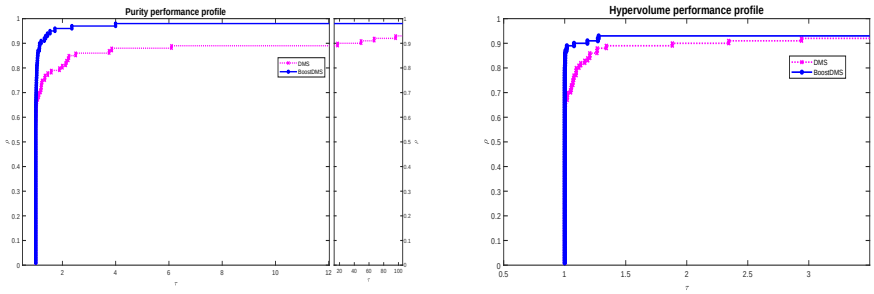

Fig. 5.1: Comparing BoostDMS and DMS based on performance profiles of the purity and hypervolume metrics.

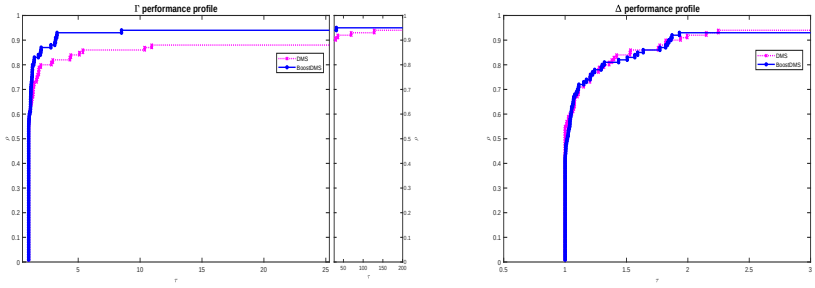

Fig. 5.2: Comparing BoostDMS and DMS based on performance profiles of spread metrics Γ and Δ.

6. Conclusions. In this paper we improved the DMS algorithm proposed in [17] to solve multiobjective derivative-free optimization problems, by defining a particular implementation of a search step based on the computation of quadratic polynomial interpolation or regression models. Models are built by reusing points where the objective function has been previously evaluated, during the course of the optimization. This approach extends to the multiobjective case the use of minimum Frobenius norm quadratic models or regression techniques, proposed in [18] for single objective directional direct search. The new variant of DMS, which considers the definition of a search step based on quadratic polynomial models, is named as BoostDMS and is freely available for use (under a GNU Lesser General Public License). Convergence results hold, given the independence of the definition of the search step in DMS (see [17]). Numerical results, considering the comparison with the original implementation of DMS and with another recent algorithm proposed for the same class of problems, support the effectiveness of the proposed strategy. Possible avenues for research include the parallelization of the algorithm and different approaches for solving the derivative-based minimization problem associated with the models of the different components of the objective function.

REFERENCES

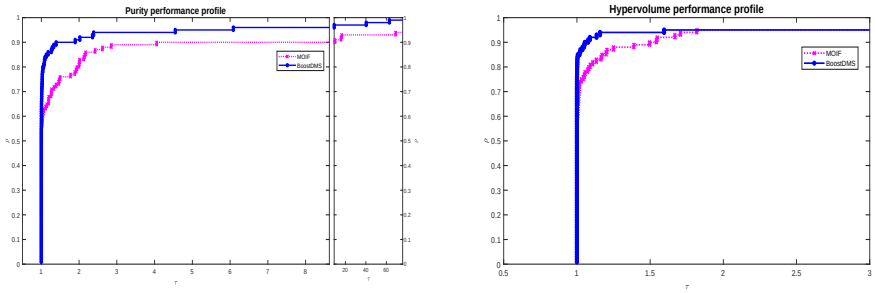

Fig. 5.3: Comparing BoostDMS and MOIF based on performance profiles of the purity and hypervolume metrics.

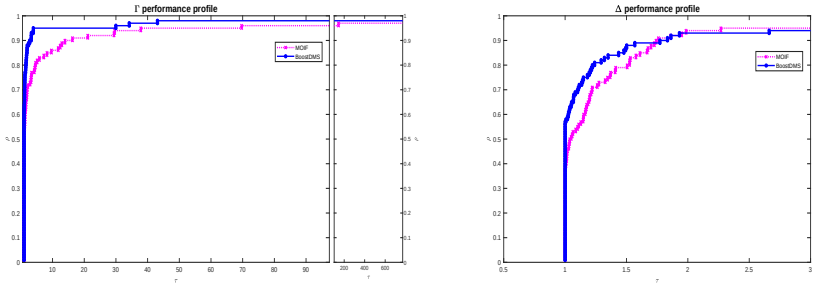

Fig. 5.4: Comparing BoostDMS and MOIF based on performance profiles of spread metrics Γ and Δ.

[1] H. Afshari, W. Hare, and S. Tesfamariam, *Constrained multi-objective optimization algorithms: Review and comparison with application in reinforced concrete structures*, Appl. Soft Comp., 83 (2019), p. 105631.

[2] T. Akhtar and C. A. Shoemaker, *Multi objective optimization of computationally expensive multi-modal functions with RBF surrogates and multi-rule selection*, J. Global Optim., 64 (2016), pp. 17–32.

[3] C. Audet and W. Hare, *Derivative-Free and Blackbox Optimization*, Springer, Cham, Switzerland, 2017.

[4] C. Audet and J. E. Dennis Jr., *Analysis of generalized pattern search*, SIAM J. Optim., 13 (2003), pp. 889–903.

[5] ———, *Mesh adaptive direct search algorithms for constrained optimization*, SIAM J. Optim., 17 (2006), pp. 188–217.

[6] C. Audet, G. Savard, and W. Zghal, *Multiobjective optimization through a series of single-objective formulations*, SIAM J. Optim., 19 (2008), pp. 188–210.

[7] J. Branke, K. Deb, K. Miettinen, and R. Slowiński, eds., *Multiobjective Optimization: Interactive and Evolutionary Approaches*, Springer-Verlag, Berlin, Germany, 2008.

[8] R. P. Brito, H. Sebastião, and P. Godinho, *Efficient skewness/semivariance portfolios*, J. Asset Manag., 17 (2016), pp. 331–346.

[9] ———, *Portfolio management with higher moments: the cardinality impact*, Int. Trans. in Oper. Res., 26 (2019), pp. 2531–2560.

[10] R. S. Burachik, C. Y. Kaya, and M. M. Rizvi, *A new scalarization technique and new algorithms to generate Pareto fronts*, SIAM J. Optim., 27 (2017), pp. 1010–1034.

[11] T. M. Chan, *Klee's measure problem made easy*, in 2013 IEEE 54th Annual Symposium on Foundations of Computer Science, IEEE, 2013, pp. 410–419.

[12] F. H. Clarke, *Optimization and Nonsmooth Analysis*, SIAM, Philadelphia, USA, 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] A. R. Conn and S. Le Digabel, *Use of quadratic models with mesh-adaptive direct search for constrained black box optimization*, Optimization Methods and Software, 28 (2013), pp. 139–158.

[15] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to Derivative-Free Optimization*, MPS-SIAM Series on Optimization, SIAM, Philadelphia, USA, 2009.

[16] 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.

[17] 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.

[18] A. L. Custódio, H. Rocha, and L. N. Vicente, *Incorporating minimum Frobenius norm models in direct search*, Comput. Optim. Appl., 46 (2010), pp. 265–278.

[19] R. Datta and R. G. Regis, *A surrogate-assisted evolution strategy for constrained multiobjective optimization*, Expert Systems Appl., 57 (2016), pp. 270–284.

[20] C. Davis, *Theory of positive linear dependence*, Amer. 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 Transactions on Evolutionary Computation, 6 (2002), pp. 182–197.

[22] S. Deshpande, L. T. Watson, and R. A. Canfield, *Multiobjective optimization using an adaptive weighting scheme*, Optim. Methods Softw., 31 (2016), pp. 110–133.

[23] E. D. Dolan and J. J. Moré, *Benchmarking optimization software with performance profiles*, Math. Program., 91 (2002), pp. 201–213.

[24] P. Feliot, J. Bect, and E. Vazquez, *A Bayesian approach to constrained single and multiobjective optimization*, J. Global Optim., 26 (2017), pp. 97–133.

[25] 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.

[26] 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), IEEE, 2006, pp. 1157–1163.

[27] J. Handl, D. B. Kell, and J. Knowles, *Multiobjective optimization in bioinformatics and computational biology*, IEEE/ACM Trans. Computational Biology and Bioinformatics, 4 (2007), pp. 279–292.

[28] 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: Emerging Technol., 65 (2016), pp. 61–78.

[29] J. Jahn, *Introduction to the Theory of Nonlinear Optimization*, Springer-Verlag, Berlin, Germany, 1996.

[30] 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.

[31] R. Lacour, K. Klamroth, and C. M.Fonseca, *A box decomposition algorithm to compute the hypervolume indicator*, Computers & Operations Research, 79 (2017), pp. 347–360.

[32] G. Liuzzi, S. Lucidi, and F. Rinaldi, *A derivative-free approach to constrained multiobjective nonsmooth optimization*, SIAM J. Optim., 26 (2016), pp. 2744–2774.

[33] K. Miettinen, *Nonlinear Multiobjective Optimization*, International Series in Operations Research & Management Science, Springer US, New York, USA, 1998.

[34] 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.

[35] M. J. D. Powell, *Developments of NEWUOA for minimization without derivatives*, IMA J. Numer. Anal., 28 (2008), pp. 649–664.

[36] G. P. Rangaiah and A. Bonilla-Petriciolet, eds., *Multi-Objective Optimization in Chemical Engineering: Developments and Applications*, John Wiley & Sons, Chichester, United Kingdom, 2013.

[37] R. G. Regis, *Multi-objective constrained black-box optimization using radial basis function surrogates*, Journal of Computational Science, 16 (2016), pp. 140–155.

[38] J.-H. Ryu and S. Kim, *A derivative-free trust-region method for biobjective optimization*, SIAM J. Optim., 24 (2014), pp. 334–362.

[39] W. I. Thacker, J. Zhang, L. T.Watson, J. B. Birch, M. A. Iyer, and M. W. Berry, *Algorithm 905: SHEPPACK: Modified Shepard algorithm for interpolation of scattered multivariate data*, ACM Trans. Math. Software, 37 (2010), pp. 1–20.

[40] J. Thomann and G. Eichfelder, *A trust-region algorithm for heterogeneous multiobjective optimization*, SIAM J. Optim., 29 (2019), pp. 1017–1047.

[41] M. M. Wiecek, M. Ehrgott, and A. Engau, *Multiple Criteria Decision Analysis*, Springer, New York, USA, 2016, ch. Continuous multiobjective programming, pp. 739–815.

[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, The Netherlands, A. E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, eds., Springer, Berlin, Heidelberg, Germany, 1998, pp. 292–301.

[43] 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. on Evolut. Comput., 7 (2003), pp. 117–132.