MULTIGLODS: GLOBAL AND LOCAL MULTIOBJECTIVE OPTIMIZATION USING DIRECT SEARCH

A. L. CUSTÓDIO * AND J. F. A. MADEIRA †

Abstract. The optimization of multimodal functions is a challenging task, in particular when derivatives are not available for use. Recently, in a directional direct search framework, a clever multistart strategy was proposed for global derivative-free optimization of single objective functions. The goal of the current work is to generalize this approach to the computation of global Pareto fronts for multiobjective multimodal derivative-free optimization problems. The proposed algorithm alternates between initializing new searches, using a multistart strategy, and exploring promising subregions, resorting to directional direct search. Components of the objective function are not aggregated and new points are accepted using the concept of Pareto dominance. The initialized searches are not all conducted until the end, merging when start to be close to each other. The convergence of the method is analyzed under the common assumptions of directional direct search. Numerical experiments show its ability to generate approximations to the different Pareto fronts of a given problem.

Key words. Global optimization, multiobjective optimization, multistart strategies, direct-search methods, nonsmooth calculus.

AMS subject classifications. 90C29, 90C56, 90C26, 90C30.

1. Introduction.

When optimizing in engineering it is common the presence of additional challenges. Functions evaluated by conducting large numerical simulations are often associated with the absence of analytical expressions for derivatives, being numerical approximations unpractical due to the large computational cost involved. There are also cases where the function to be optimized is nonsmooth, which prevents the use of derivative-based techniques. Additionally, several times there is more than one objective in this optimization process, generally conflicting, which motivates the use of multiobjective derivative-free algorithms.

Solution techniques depend on the moment where the decision maker establishes preferences relating the different objectives to optimize [15]. Methods could be classified as having an *a prior articulation of preferences*, when objectives are aggregated into a single objective function which is then optimized, generating a single point as solution to the multiobjective optimization problem. Changes in preferences will cause changes in the aggregating function and the optimization procedure will need to be reapplied. Another approach consists in *a posteriori articulation of preferences*. The methods belonging to this class attempt to capture the whole Pareto front of the problem, never establishing preferences among the several objectives.

Direct MultiSearch (DMS) [6] is a well-established algorithm belonging to this last class. In [6] convergence results were derived, stating that at least one limit point of the sequence of iterates generated by DMS lies in a stationary form of a Pareto front. Intensive numerical testing showed its competitiveness with other state-of-art

*Department of Mathematics, FCT-UNL-CMA, Campus de Caparica, 2829-516 Caparica, Portugal (`alcustodio@fct.unl.pt`). Support for this author was provided by Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) under the project UID/MAT/00297/2013 (CMA).

†IDMEC, Instituto Superior Técnico, Universidade de Lisboa, Av. Rovisco Pais, 1040-001 Lisbon, Portugal and Department of Mathematics, ISEL, Rua Conselheiro Emídio Navarro, 1, 1959-007 Lisbon, Portugal (`aguilarmadeira@tecnico.ulisboa.pt`). Support for this author was provided by ISEL, IPL, Lisboa, Portugal and LAETA, IDMEC, Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal.

algorithms, like is the case of NSGA-II [9] or BIMADS [3].

Nevertheless, as mentioned in [8], the optimization of multimodal functions raises issues regarding the convergence to the true Pareto front of a problem (the global Pareto front). In fact, this question is not specific of multiobjective optimization, since global optimization is an active field of research for single objective optimization, both in presence or absence of derivatives.

Recently, in the context of single objective derivative-free optimization, GLODS algorithm [5] was proposed as a strategy for identifying global minimizers. GLODS is a directional direct search method [4] equipped with a *clever multistart strategy*. The different searches initialized with the multistart strategy are not all conducted until the end. Rather, when points generated by different searches start to be closed to each other, GLODS will merge searches, giving up on the ones that are not promising. This procedure showed to be competitive when compared with state-of-art solvers in global derivative-free optimization. Additionally, numerical experiments showed its capability of identifying all the local (and global) optimums of the problem, a distinguishing feature from the remaining global derivative-free optimization solvers.

The goal of the present work is to address the computation of the global Pareto front of the problem:

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

where $F : \mathbb{R}^n \to \mathbb{R}^m \cup \{(+\infty, \ldots, +\infty)\}, n, m \geq 1$ represents a real-extended multi-valued function, and the compact set $\Omega \subset \mathbb{R}^n$ denotes the feasible region of the problem. We assume that derivatives are not available for use, neither can be numerically approximated.

Constraints will be addressed through an extreme barrier approach, meaning that only feasible points will be evaluated. In a context of expensive function evaluation, this avoids unnecessary computations. We also note that when the objective function represents a real application, the evaluation of unfeasible points could be impossible, corresponding to points with no physical meaning. The extreme barrier approach will be implemented through the use of a barrier function:

$$F_\Omega(x) = \begin{cases} F(x) & \text{, if } x \in \Omega, \\ (+\infty, \ldots, +\infty) & \text{, otherwise.} \end{cases} \tag{1.2}$$

Using the concept of Pareto dominance, we will generalize the approach followed by GLODS to multiobjective directional direct search, conferring a global flavor to DMS. In Section 2 we will describe the proposed algorithmic structure. The theoretical results associated with the new algorithm will be presented in Section 3, and Section 4 illustrates the corresponding global features. We end in Section 5 with some conclusions.

2. MultiGLODS: Global and Local Multiobjective Optimization using Direct Search.

In multiobjective optimization, where the objective function presents several components, the concept of dominance is used for comparing pairs of points. We say that point x dominates point y, and represent it by $x \prec y$, if the following condition is satisfied:

$$x \prec y \Leftrightarrow F_\Omega(y) - F_\Omega(x) \in \mathbb{R}^m_+ \setminus \{0\}. \tag{2.1}$$

It is now possible to define what is a solution for problem 1.1.

Definition 2.1. *A point $x_* \in \Omega$ is said to be a global Pareto minimizer of F in Ω if $\nexists y \in \Omega$ such that $y \prec x_*$. If there exists a neighborhood $\mathcal{N}(x_*)$ of x_* such that the previous property holds in $\Omega \cap \mathcal{N}(x_*)$, then x_* is called a local Pareto minimizer of F.*

In general, a problem will have several points satisfying Definition 2.1. The set of all these points will define a Pareto front for the problem (global or local, depending on the type of condition satisfied).

Like any other directional direct search method, each iteration of MultiGLODS is organized in a *search step* and a *poll step*. The convergence properties of the algorithm are a direct consequence of the poll step. In a context of global optimization, the search step is mainly responsible for spreading the search in the feasible region, ensuring that promising subregions will be located. The quality of the computed Pareto front (as being local or global to the problem) is, in general, a consequence of the search step.

The algorithm keeps a list of feasible points, L_k, which could be updated both at the search and the poll steps. Points are stored in this list as tuples $(x; \alpha_x; r_x; i_x)$, where x represents the point to be stored, α_x a step size, r_x a comparison radius and i_x a binary indicator, which takes the value 1 if the point is active and 0, otherwise.

Similarly to GLODS [5], a point is added to the list as active or inactive. A point is added as active if it is far from all the points already stored, meaning that it is located in a part of the feasible region not yet explored (see lines 1 and 2 in Algorithm 2.2). Radius r_y is used in point comparisons, as a measure of closeness (see lines 1 and 3 in Algorithm 2.2). Alternatively, points close to points already stored, which dominate active points are also added to the list. If the point dominates an active point and it is not dominated by any other close point already in the list, it will be added to the list as active (see lines 5 and 6 in Algorithm 2.2). A point that dominates an active point, but it is dominated by another one, will be added to the list as inactive (see line 6 in Algorithm 2.2). An active point already in the list could change its status to inactive, if it is dominated by a new point added to the list (see line 4 in Algorithm 2.2). An inactive point will never change its status to active.

The relevance of the active indicator i_x and the step size parameter α_x respects to the poll step of the algorithm. Each time this step is performed, one active point x will be selected as a poll center and a local search around it will be conducted. This local search corresponds to the test of directions belonging to a positive spanning set D [7], scaled by the step size parameter α_x. As it will be clear in the convergence analysis (see Section 3), the set of poll directions is not required to positively span $\mathbb{R}^n$ (although for coherency with the smooth case we will write it so in the algorithm below), and it is not necessarily drawn from a finite set of directions. Opportunistic or complete approaches can be used in the polling procedure. In the first case, the procedure will stop once that a new active point is added to the list. In the latter, all directions will be tested.

Different strategies can be used for generating new points at the search step. Random sampling [18], Latin hypercube sampling [16], Sobol sequences [12], Halton sequences [12] or 2^n-Centers [5] are some possibilities. The major requisite is using an asymptotically dense sequence in a compact set. Additionally, points could be required to belong to an implicit mesh, depending on the globalization strategy considered (see Section 3).

As result of the two steps described, each iteration is classified as *successful*, *unsuccessful*, or *merging*. Successful iterations correspond to at least one active point added to the list. In this case, the corresponding step size parameter could be maintained or increased. Unsuccessful iterations occur when the list has no changes. In this case, the step size corresponding to the poll center will obligatory be decreased. Adding only inactive points to the list corresponds to a merging iteration. In this case step size parameters are kept unchanged.

Comparison radius should always allow the comparison between the poll points and the corresponding poll center. Thus, if after a successful iteration, as a consequence of updating the step size parameter this property does not hold, the comparison radius will be increased to an adequate value.

At each iteration, if the search step fails in adding a new active point to the list of points, the poll step obligatory needs to be performed. The search step does not need to be executed at every iteration (when it is not conducted, it will be considered as a failure). Different strategies could be implemented to decide if the search step should be performed or not. Some possibilities could consider the frequency of unsuccessful iterations or the size of the step size parameters for active points.

A detailed description of the method proposed can be found in Algorithm 2.1.

Algorithm 2.1 (**MultiGLODS: Global and Local Multiobjective Optimization using Direct Search**).

Initialization

Let $\mathcal{D}$ be a (possibly infinite) set of positive spanning sets, such that $\forall d \in D \in \mathcal{D}, 0 < d_{min} \leq \|d\| \leq d_{max}$. Choose $\alpha_0 > 0$, $r_0 \geq d_{max}\alpha_0 > 0$, $0 < \beta_1 \leq \beta_2 < 1$, and $\gamma \geq 1$. Set $L_0 = \emptyset$.

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

1. **Search step:** Compute a finite set of distinct points $A_k = \{(x_j; 0; 0; 0) : F_\Omega(x_j) < (+\infty, \ldots, +\infty)\}$ (all x_j should be in a mesh if $\bar{\rho}(\cdot) \equiv 0$, see Section 3.1). Call L_{k+1} `=` `add(`L_k`,`A_k`)` to possibly add some new points in A_k to L_k. If $k = 0$, set $L_0 = L_1$ and go to the poll step. Otherwise, if there is a new active point in L_{k+1} declare the iteration (and the search step) as successful and skip the poll step.

2. **Poll step:** Order the list L_k and select an active point $(x; \alpha_x; r_x; 1) \in L_k$ as the current iterate, corresponding step size parameter and comparison radius (thus setting $(x_k; \alpha_k; r_k; i_k) = (x; \alpha_x; r_x; 1)$).
   Choose a positive spanning set D_k from the set $\mathcal{D}$. Compute the set of poll points $P_k = \{(x_k + \alpha_k d; \alpha_k; \alpha_k\|d\|; 0) : d \in D_k \wedge F_\Omega(x_k + \alpha_k d) < (+\infty, \ldots, +\infty)\}$. Call L_{k+1} `=` `add(`L_k`,`P_k`)` to possibly add some new points in P_k to L_k. If there is a new active point in L_{k+1} declare the iteration (and the poll step) as successful. If no new point was added to L_k declare the iteration (and the poll step) as unsuccessful. Otherwise declare the iteration (and the poll step) as merging.

3. **Step size parameter and radius update:** If the iteration was successful then maintain or increase the corresponding step size parameters: $\alpha_{new} \in [\alpha, \gamma\alpha]$ and replace all the new points $(x; \alpha_x; r_x; 1)$ in L_{k+1} by $(x; \alpha_{new}; d_{max}\alpha_{new}; 1)$, if $d_{max}\alpha_{new} > r_x$, or by $(x; \alpha_{new}; r_x; 1)$, when $d_{max}\alpha_{new} \leq r_x$.
   If the iteration was unsuccessful then decrease the corresponding step size parameter: $\alpha_{new} \in [\beta_1\alpha_k, \beta_2\alpha_k]$ and replace the poll center $(x_k; \alpha_k; r_k; 1)$

in L_{k+1} by $(x_k; \alpha_{new}; r_k; 1)$.

Using (2.1), let us denote by $Dom(x)$ the subset of $\mathbb{R}^m$ corresponding to the images of the set of points dominated by x. Algorithm 2.2 corresponds to the procedure used both in the search and poll steps to add new points to the list.

Algorithm 2.2: $L_1 =$ `add` (L_1, L_2)
Procedure for adding new points, stored in L_2, to the current list, L_1.

```
  forall $(x; \alpha_x; r_x; 0) \in L_2$ do
1     if $\min_{y \in L_1} (\|x - y\| - r_y) > 0$ then
2         $L_1 = L_1 \cup \{(x; \alpha_0; r_0; 1)\}$
      else
          if $x \notin L_1$ then
              set $\alpha_a = 0$, $r_a = 0$, $i_{dom} = 0$ and $p_{dom} = 0$
              forall $(y; \alpha_y; r_y; i_y) \in L_1$ do
3                 if $\|x - y\| - r_y \leq 0$ then
                      if $F(y) \in Dom(x) + \bar{\rho}(\alpha_y)$ then
                          $i_{dom} = i_{dom} + i_y$
4                         $i_y = 0$
                          if $\alpha_y > \alpha_a$ then
                              $\alpha_a = \alpha_y$
                              $r_a = r_y$
                          end
                      else
                          if $F(x) \in Dom(y)$ then
                              $p_{dom} = 1$
                          end
                      end
                  end
              end
              if $p_{dom} = 0$ then
5                 $i_x = 1$
              end
              if $\alpha_a = 0$ then
                  $\alpha_a = \alpha_0$
                  $r_a = r_0$
              end
6             if $(i_{dom} > 0 \vee (p_{dom} = 0 \wedge \bar{\rho}(.) \equiv 0))$ then
                  if $\alpha_x = 0$ then
7                     $L_1 = L_1 \cup \{(x; \alpha_a; r_a; i_x)\}$
                  else
8                     $L_1 = L_1 \cup \{(x; \alpha_x; r_x; i_x)\}$
                  end
              end
          end
      end
  end
```

Function $\bar{\rho}(.)$ is related to the type of globalization strategy considered in the algorithm (see Section 3). If globalization is based on the use of integer lattices, it represents the constant zero function. When globalization results from requiring sufficient decrease for accepting new points, $\bar{\rho}(.) \equiv \rho(.)$ will be 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 [13]). Typical examples of forcing functions are $\rho(t) = t^{1+a}$, for $a > 0$.

Considering integer lattices as globalization strategy allows an additional situation where points are added to the list as active. This occurs when the new point is comparable with other points already stored in the list (independently of being active or inactive points) and it is not dominated by any of them (see line 6 in Algorithm 2.2).

When adding points to the list, the corresponding step size parameters and comparison radius need to be defined. Similarly to GLODS [5], if the point was not comparable with any of the points already in the list, meaning that it belongs to a part of the feasible region not yet explored, the algorithm uses the initialization values (line numbered as 2 in Algorithm 2.2). Otherwise, if the point was generated at the poll step, both parameters will be equal to the ones of the poll center (line numbered as 8 in Algorithm 2.2). When the new point x was generated in the search step, it inherits the parameters of the point y, presenting the largest step size, comparable with it, for which $F(y) \in Dom(x) + \bar{\rho}(\alpha_y)$ (line numbered as 7 in Algorithm 2.2).

3. Convergence analysis.

The convergence analysis of MultiGLODS relies on the properties of the poll step, which begins with the choice of a poll center from the active points stored in the list. Merging iterations in MultiGLODS correspond to situations where no new active points are added to the list and some stored active points change their status to inactive. Thus, it is crucial to ensure that at each iteration there is always an active point in the list that could be selected as poll center.

PROPOSITION 3.1. *At the end of each iteration of Algorithm 2.1, all elements of the set of nondominated points in the list are active.*

Proof. Suppose not. Let z be one inactive point of the set of nondominated points in the list, computed at the end of the current iteration. Two situations need to be analyzed:

- z could have been added as inactive to the list (and to the set of nondominated points), during the current iteration;
- z was an active point already in the list (and in the set of nondominated points), but has changed its status to inactive during the current iteration.

In the latter situation, there should have been a point x such that $F(z) \in Dom(x) + \bar{\rho}(\alpha_z) \subseteq Dom(x)$. Since z was active, x will be added to the list of points, contradicting the fact of z being nondominated.

In the former situation, there should have been y already in the list, such that $F(z) \in Dom(y)$, again contradicting the fact of z being nondominated. □

Similarly to any other directional direct search method, the convergence analysis of MultiGLODS starts by establishing the existence of a subsequence of step size parameters that converges to zero. This will allow us to ensure the existence of at least one limit point for the sequence of iterates generated by the algorithm. The stationarity properties of this limit point will be further analyzed in Section 3.4.

Two globalization strategies can be adopted in order to ensure the existence of a subsequence of step size parameters with the above mentioned property. The first considers that all points generated by the algorithm lie in an implicit mesh (corresponding to an integer lattice) and will be analyzed in Section 3.1. In this case $\bar{\rho}(.) \equiv 0$. Another possibility is to exchange the freedom in the type of points generated by the algorithm by a more strict condition when accepting new points. In this case $\bar{\rho}(.)$ corresponds to a forcing function (see Section 3.2).

For establishing the results, we will need the following two assumptions.

Assumption 3.1. *The set $\Omega \subset \mathbb{R}^n$ is compact.*

Assumption 3.2. *The function F is lower bounded in $\Omega \subset \mathbb{R}^n$.*

3.1. Using integer lattices. The type of positive spanning sets that can be used by the algorithm depend on the level of smoothness present in the objective function. If the function is continuously differentiable, a finite set of directions with appropriate integrality requirements will suffice [1, 13].

Assumption 3.3. *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 $\mathbb{Z}^n$.*

In the presence of nonsmooth functions, the integrality requirements should be respected but additionally the reunion of the sets of directions (after normalization) considered through the iterations needs to be asymptotically dense in the unit sphere [2].

Assumption 3.4. *Let D represent a finite set of positive spanning sets satisfying Assumption 3.3.*

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 as part of the MADS original presentation [2], for coherency with the smooth case, but it is not used in the convergence analysis for nonsmooth objective functions.

Integrality requirements also impose conditions in the update of the step size parameter.

Assumption 3.5. *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 step size 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 step size parameter is decreased by setting $\alpha_{new} = \tau^{m^-} \alpha$, with $m^- \in \{m^{min}, \ldots, -1\}$.*

Notice that the step size update strategy proposed in Algorithm 2.1 complies to the one of Assumption 3.5 by setting $\beta_1 = \tau^{m^{min}}$, $\beta_2 = \tau^{-1}$, and $\gamma = \tau^{m^{max}}$.

Additionally, the points generated by the search step need to lie in the implicit mesh considered at each iteration by the algorithm (trivially all the poll points generated by the algorithm will also lie in this implicit mesh).

Assumption 3.6. *At iteration k, the search step in Algorithm 2.1 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.

We are now in conditions of establishing the first result, regarding the sequence of step size parameters generated by MultiGLODS.

THEOREM 3.2. *Let Assumption 3.1 hold. Algorithm 2.1 under one of the Assumptions 3.3 or 3.4 combined with Assumptions 3.5–3.6 and $\bar{\rho}(\cdot) \equiv 0$ generates a sequence of iterates satisfying*

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

Proof. Let us assume that there is α_* such that $\alpha_k > \alpha_* > 0, \forall k$. Assumptions 3.3 or 3.4 combined with Assumptions 3.5–3.6 ensure that all points generated by Algorithm 2.1 lie in an integer lattice (see [1, 2]). The intersection of a compact set with an integer lattice is finite. Since Ω is compact, there is only a finite number of different points that could be generated by the algorithm. Successful or merging iterations correspond to at least one new feasible point added to the list. Once a point is added to this list it will always remain on it (eventually changing its status to inactive). Thus, the number of successful and merging iterations must be finite, and consequently there is an infinite number of unsuccessful iterations and a finite number of points in the list. The step size at unsuccessful iterations is reduced by at least $\beta_2 \in]0,1[$, which contradicts the existence of a lower bound for the step size parameter. □

3.2. Imposing sufficient decrease. When the globalization strategy is based on imposing a sufficient decrease condition, there is more flexibility in the update of the step size parameter and in the type of directions that could be considered by the algorithm. In fact, the only requirement is now expressed in Assumption 3.7, and is trivially satisfied since we are considering bounded sets of directions.

ASSUMPTION 3.7. *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.

A similar result to the one of Theorem 3.2 can now be derived.

THEOREM 3.3. *Let Assumptions 3.1–3.2 hold. Algorithm 2.1, when $\bar{\rho}(\cdot)$ is a forcing function and Assumption 3.7 holds, generates a sequence of iterates satisfying*

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

Proof. Assume that there is α_* such that $\alpha_k > \alpha_* > 0, \forall k$. Let us start by showing that there is an infinite number of successful iterations.

Suppose not. Active points are added to the list only at successful iterations. Thus, the number of active points in the list must be finite. At each merging iteration at least one of the active points in the list changes its status to inactive. Thus, the number of merging iterations is also finite.

Consequently, the number of unsuccessful iterations (where no points are added to the list) needs to be infinite. At each unsuccessful iteration the step size parameter of the corresponding active poll center is reduced by at least $\beta_2 \in]0,1[$, which contradicts the existence of the lower bound $\alpha_* > 0$ for the step size.

The previous arguments allow us to conclude that there is an infinite number of successful iterations. Let x represent a new feasible active point added to the list L_k, at iteration k. Then, $\min_{y \in L_k} (\|x - y\| - r_y) > 0$ or there should have been an active point $y \in L_k$ such that $\|x - y\| \leq r_y$ and $F(y) \in Dom(x) + \bar{\rho}(\alpha_y)$.

Let us assume that for each successful iteration k there is always a new active point, $x_{k+1} \in \Omega$, to be added to L_k, such that $\min_{y \in L_k} (\|x_{k+1} - y\| - r_y) > 0$. Thus,

$$\forall y \in L_k, \|x_{k+1} - y\| > r_y \geq d_{min}\alpha_y > d_{min}\alpha_* > 0.$$

Once a point is added to the point list it will always remain in it (eventually being inactive). Thus, at each successful iteration the measure of

$$\Omega \setminus \bigcup_{k \in S} B(x_{k+1}; d_{min}\alpha_*)$$

decreases by a strictly positive quantity. Here S represents the set of indexes corresponding to successful iterations and $B(x_{k+1}; d_{min}\alpha_*)$ the closed ball of radius $d_{min}\alpha_*$, centered at x_{k+1}. Since Ω is compact there should have been $k^* \in \mathbb{N}$ such that for each successful iteration $k \geq k^*$ and for each new feasible active point x added to L_k, there is an active point $y \in L_k$, which changes the corresponding status to inactive, with $\|x - y\| \leq r_y$ and $F(y) \in Dom(x) + \bar{\rho}(\alpha_y)$. Points are only added to the list at successful and merging iterations. For each point x added to L_k at a merging iteration, there should also have been an active point $y \in L_k$, which changes the corresponding status to inactive, and such that $\|x - y\| \leq r_y$ with $F(y) \in Dom(x) + \bar{\rho}(\alpha_y)$. Thus, each time that a point is added to the list, it will add an hypercube of side no smaller than ρ_* to the dominance region of an active point already in the list, which changes the corresponding status to inactive. After a finite number of iterations, an hypercube of side no smaller than ρ_* would have been added to the dominance region defined by all the points in the list. Since there is an infinite number of successful iterations, this contradicts Assumption 3.2. $\square$

3.3. Refining subsequences, refining directions and generalized directional derivatives. Convergence results for MultiGLODS are derived for limit points of the so-called *refining subsequences.*

DEFINITION 3.4. *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.*

Having established the existence of a subsequence of step size parameters converging to zero, the updating rules of the step size parameter allow us to ensure the existence of at least one refining subsequence (see, for example, [4]).

THEOREM 3.5. *Let the conditions required for establishing Theorem 3.2 or Theorem 3.3 hold. Algorithm 2.1 generates at least one refining subsequence $\{x_k\}_{k \in K}$, converging to $x_* \in \Omega$.*

MultiGLODS behavior will be analyzed at limit points of convergent refining subsequences, along *refining directions.* This last concept was introduced in [2], in the context of Mesh Adaptive Direct Search (MADS).

DEFINITION 3.6. *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_*.*

Given the nonsmoothness present in the objective function, we will need to use a generalized definition of Pareto stationarity. This definition makes use of the Clarke-Jahn [11] generalized directional derivative, computed for directions belonging to the tangent cone to the feasible region or to its interior.

Definition 3.7. *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 and it is a generalization of the tangent cone commonly used in Nonlinear Programming (see, e.g., [17, Definition 12.2 and Figure 12.8]).

The interior of this cone defines the hypertangent cone ($T_\Omega^H(x)$), which corresponds to the set of hypertangent vectors to Ω at x.

Definition 3.8. *A vector $d \in \mathbb{R}^n$ is said to be a hypertangent vector to the set $\Omega \subset \mathbb{R}^n$ at the point x in Ω if there exists a scalar $\epsilon > 0$ such that*

$$y + tw \in \Omega, \quad \forall y \in \Omega \cap B(x;\epsilon), \quad w \in B(d;\epsilon), \quad \textit{and} \quad 0 < t < \epsilon.$$

The Clarke tangent cone can also be regarded as the closure of the hypertangent cone.

The Clarke-Jahn generalized directional derivative is well defined for functions locally Lipschitz continuous. Function $F(x)$ is said to be Lipschitz continuous near x if each $f_i(x)$, $i = 1, \ldots, m$ is Lipschitz continuous in a neighborhood of x.

In these conditions, the Clarke-Jahn generalized directional derivative can be defined for a component of F, f_j, in a direction d belonging to the hypertangent cone to Ω at x as,

$$f_j^\circ(x;d) \;=\; \limsup_{\substack{x' \to x,\, x' \in \Omega \\ t \downarrow 0,\, x' + td \in \Omega}} \frac{f_j(x'+td) - f_j(x')}{t}, \; j = 1, \ldots, m. \tag{3.1}$$

The extension of this derivative to directions v in the tangent cone to Ω at x is computed by taking a limit, i.e., $f_j^\circ(x;v) = \lim_{d \in T_\Omega^H(x), d \to v} f_j^\circ(x;d)$, for $j = 1, \ldots, m$ (see Proposition 3.9 in [2]).

We are now in conditions of defining the type of stationarity results that we intend to obtain for MultiGLODS.

Definition 3.9. *Let F be Lipschitz continuous near a point $x_* \in \Omega$. We say that x_* is a Pareto-Clarke critical point of F in Ω if, for all directions $d \in T_\Omega^{Cl}(x_*)$, there exists a $j \in \{1, \ldots, m\}$ such that $f_j^\circ(x_*;d) \geq 0$.*

When each component of the objective function is strictly differentiable, the previous definition can be restated using the gradient vectors.

Definition 3.10. *Let F be strictly differentiable at a point $x_* \in \Omega$. We say that x_* is a Pareto-Clarke-KKT critical point of F in Ω if, for all directions $d \in T_\Omega^{Cl}(x_*)$, there exists a $j \in \{1, \ldots, m\}$ such that $\nabla f_j(x_*)^\top d \geq 0$.*

3.4. Convergence results. Lets us start by establishing a first stationarity result, not for the whole set of directions belonging to the Clarke tangent cone to the feasible region, but for refining directions in the hypertangent cone.

Theorem 3.11. *Consider a refining subsequence $\{x_k\}_{k \in K}$ converging to $x_* \in \Omega$ and let $d \in T_\Omega^H(x_*)$ be a refining direction for x_*. Assume that F is Lipschitz continuous near x_*. Then there exists a $j \in \{1, \ldots, m\}$ such that $f_j^\circ(x_*;d) \geq 0$.*

Proof. Let $\{x_k\}_{k \in K}$ be a refining subsequence converging to $x_* \in \Omega$ and

$$d = \lim_{k \in K''} d_k / \|d_k\| \in T_\Omega^H(x_*)$$

a refining direction for x_*, with $d_k \in D_k$ and $x_k + \alpha_k d_k \in \Omega$, $\forall k \in K'' \subseteq K$.

Since F is Lipschitz continuous near x_*, the Clarke-Jahn generalized directional derivative is well defined for each $f_j(x_*), j = 1, \ldots, m$ and we have:

$$\begin{aligned}
f_j^\circ(x_*; d) = & \limsup_{\substack{x \to x_*, x \in \Omega \\ t \downarrow 0, x+td \in \Omega}} \frac{f_j(x+td) - f_j(x)}{t} \\
& \geq \limsup_{k \in K''} \frac{f_j(x_k + \alpha_k \|d_k\| (d_k/\|d_k\|)) - f_j(x_k)}{\alpha_k \|d_k\|} + r_k \\
& = \limsup_{k \in K''} \frac{f_j(x_k + \alpha_k d_k) - f_j(x_k) + \bar{\rho}(\alpha_k)}{\alpha_k \|d_k\|} - \frac{\bar{\rho}(\alpha_k)}{\alpha_k \|d_k\|} + r_k.
\end{aligned}$$

The first inequality follows from $\{x_k\}_{k \in K''}$ being a feasible refining subsequence and the fact that $x_k + \alpha_k d_k$ is feasible for $k \in K''$. The term r_k is bounded above by $\nu ||d - d_k/\|d_k\|||$, where ν is the Lipschitz constant of F near x_*. Note, also, that the limit $\lim_{k \in K''} \bar{\rho}(\alpha_k)/(\alpha_k \|d_k\|)$ is 0 for both globalization strategies (Subsections 3.1 and 3.2). In the case of using integer lattices (Subsection 3.1), one uses $\bar{\rho}(\cdot) \equiv 0$. When imposing sufficient decrease (Subsection 3.2), this limit follows from the properties of the forcing function and the existence of d_{min}, a strictly positive lower bound for the norm of the poll directions.

Since $x_k + \alpha_k d_k$ corresponds to a point evaluated at the unsuccessful iteration k, it was necessarily compared with the active poll center x_k. Thus $F(x_k) \notin Dom(x_k + \alpha_k d_k) + \bar{\rho}(\alpha_k)$, meaning that there should be $j(k) \in \{1, \ldots, m\}$ such that$f_{j(k)}(x_k + \alpha_k d_k) - f_{j(k)}(x_k) + \bar{\rho}(\alpha_k) > 0$. Considering that the number of components of the objective function is finite, by passing to a subsequence indexed in $K''' \subseteq K''$, there exists $j = j(d)$ such that

$$f_{j(d)}^\circ(x_*; d) \geq \limsup_{k \in K'''} \frac{f_{j(d)}(x_k + \alpha_k d_k) - f_{j(d)}(x_k) + \bar{\rho}(\alpha_k)}{\alpha_k \|d_k\|} \geq 0.$$

□

Assuming strict differentiability of the objective function we can state a similar result but considering the gradient vectors of each component of F.

Corollary 3.1. *Consider a refining subsequence $\{x_k\}_{k \in K}$ converging to $x_* \in \Omega$ and let $d \in T_\Omega^H(x_*)$ be a refining direction for x_*. Assume that F is strictly differentiable at x_*. Then there exists a $j \in \{1, \ldots, m\}$ such that $\nabla f_j(x_*)^\top d \geq 0$.*

Proof. The result follows immediately from Theorem 3.11 by noticing that strict differentiability implies that $f_j^\circ(x_*; d) = \nabla f_j(x_*)^\top d$ (see [11]). □

The previous results can now be extended to the whole Clarke tangent cone by assuming density of the set of refining directions associated with x_*.

Theorem 3.12. *Consider a refining subsequence $\{x_k\}_{k \in K}$ converging to $x_* \in \Omega$. Assume that $T_\Omega^{Cl}(x_*) \neq \emptyset$ and F is Lipschitz continuous near x_*. If the set of refining directions for x_* is dense in $T_\Omega^{Cl}(x_*)$, then x_* is a Pareto-Clarke critical point.*

In addition, if F is strictly differentiable at x_*, then this point is a Pareto-Clarke-KKT critical point.

Proof. Given a direction $v \in T_\Omega^{Cl}(x_*)$, for each $j \in \{1, \ldots, m\}$ the Clarke-Jahn generalized directional derivative can be obtained as

$$f_j^\circ(x_*;v) \;=\; \lim_{\substack{d \to v \\ d \in T_\Omega^H(x_*)}} f_j^\circ(x_*;d).$$

Since the set of refining directions for x_* is dense in $T_\Omega^{Cl}(x_*)$ then $v \;=\; \lim_{r\in R} d_r$ with d_r a refining direction for x_* belonging to $T_\Omega^H(x_*)$. Considering the result of Theorem 3.11 and since the number of components of the objective function is finite, by passing to a subsequence $R' \subseteq R$ we have $v \;=\; \lim_{r\in R'} d_r$, with $d_r \in T_\Omega^H(x_*)$, and $f_{j(v)}^\circ(x_*;d_r) \geq 0, \forall r \in R'$. The first statement of the theorem results from considering limits of this sequence of generalized derivatives. The second statement of the theorem results trivially. □

4. Numerical experiments. In this section we intend to illustrate the numerical behavior of MultiGLODS. In particular, we aim at stating its ability to approximate local and global Pareto fronts of a given multiobjective derivative-free optimization problem. For that, we have considered 14 bound constrained problems:

$$\begin{array}{ll} \min & F(x) \equiv (f_1(x), \ldots, f_m(x)) \\ \text{s.t.} & l \leq x \leq u \end{array} \tag{4.1}$$

Table 4.1 reports the corresponding dimensions (n), the number of components of the objective function (m), and the bounds.

As part of our test set, we have problems belonging to the ZDT [19] and the DTLZ [10] collections, and also two additional problems described in [8] (Sections 4.1 and 5.1.2), named as Deb213 and Deb218, respectively. Additionally, the technique described in Section 4 of [8] was used to generate two new biobjective problems, both presenting local and global Pareto fronts. The goal of testing the two first collections is to state the quality of MultiGLODS as a general derivative-free multiobjective optimization solver. Problems proposed in [8] are useful to test its ability to identify local and global Pareto fronts of a given problem.

A Matlab numerical implementation of MultiGLODS is available at:

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

This numerical implementation was tested against version 0.3 of Direct MultiSearch (DMS) [6]. DMS is a well-established algorithm, suited for derivative-free multiobjective optimization, which has proved to be competitive with state-of-art solvers. Nowadays it is still used for benchmarking new multiobjective derivative-free algorithms [14].

In order to access the real impact of the clever multistart strategy, both algorithms were run with parameters similar to the defaults of DMS. Exception occurs in the use of a cache, which was disabled for both algorithms. Initialization considered a number of points equal to problem dimension, evenly spaced in a line joining the problem bounds. Inspired by the defaults of GLODS, MultiGLODS additionally considered the center of the feasible region. As globalization strategy, both algorithms used integer lattices. Coordinate directions were used as positive spanning sets and complete polling was performed.

Regarding MultiGLODS, each time that three consecutive unsuccessful iterations occurred, the search step was performed, using Sobol sequences to generate a number of feasible points equal to problem dimension. As poll center, MultiGLODS selected the active point, not yet identified as a local Pareto point, presenting the largest step

Problem	n	m	l	u
ZDT1	30	2	$[0,\dots,0]$	$[1,\dots,1]$
ZDT2	30	2	$[0,\dots,0]$	$[1,\dots,1]$
ZDT3	30	2	$[0,\dots,0]$	$[1,\dots,1]$
ZDT4	10	2	$[0,-5,\dots,-5]$	$[1,5,\dots,5]$
ZDT6	10	2	$[0,\dots,0]$	$[1,\dots,1]$
DTLZ1	7	3	$[0,\dots,0]$	$[1,\dots,1]$
DTLZ2	12	3	$[0,\dots,0]$	$[1,\dots,1]$
DTLZ3	12	3	$[0,\dots,0]$	$[1,\dots,1]$
DTLZ5	12	3	$[0,\dots,0]$	$[1,\dots,1]$
DTLZ7	22	3	$[0,\dots,0]$	$[1,\dots,1]$
Deb213	2	2	$[0.1,0]$	$[1,1]$
Deb218	2	2	$[0.1,0]$	$[1,1]$
CAM1	2	2	$[0.1,0]$	$[1,1]$
CAM2	2	2	$[0.1,0]$	$[1,1]$

Table 4.1: Problems considered in the numerical experimentation.

size. A point is identified as a local Pareto point when the corresponding step size parameter is below 10^{-3}.

The step size was initialized as $n \times \max_{i\in\{1,\dots,n\}}(u_i - l_i)$ in MultiGLODS and was set equal to 1 in DMS. Successful iterations kept constant the step size, which was halved at unsuccessful ones.

Both algorithms would stop when a maximum of 20000 function evaluations was reached or when the step sizes for all points were below 10^{-3} (in the case of MultiGLODS, only active points were considered).

4.1. Additional multimodal multiobjective problems. According to [8], let us consider the biobjective optimization problem defined as:

$$\begin{array}{ll} \min & F(x_1,x_2) \equiv \left(x_1, \dfrac{g(x_2)}{x_1}\right) \\ \text{s.t.} & (x_1,x_2) \in \Omega \subset \mathbb{R}^2, \end{array} \tag{4.2}$$

where $g(x_2) > 0$.

Theorem 1 in [8] states that this problem has local or global Pareto optimal solutions (x_1, x_2^*), where x_2^* corresponds to a local or global minimum of $g(x_2)$, respectively, and x_1 can take any feasible value.

Using the proposed technique, we have generated problems CAM1 and CAM2, considering the function g defined as below,

$$g(x_2) = 2 - e^{-\left(\frac{x_2-0.2}{0.004}\right)^2} - c_1\, e^{-\left(\frac{x_2-0.6}{0.4}\right)^2} - c_2\, e^{-\left(\frac{x_2-0.9}{0.002}\right)^2}.$$

For CAM1, $c_1 = 1.9$ and $c_2 = 0$, while in CAM2, $c_1 = 0.8$ and $c_2 = 1.2$. In both problems $x_1 \in [0.1, 1]$ and $x_2 \in [0, 1]$.

Figure 4.1 provides a graphical representation of function g for CAM1 and CAM2, respectively. Comparing with the function g considered in [8], we see that for CAM1

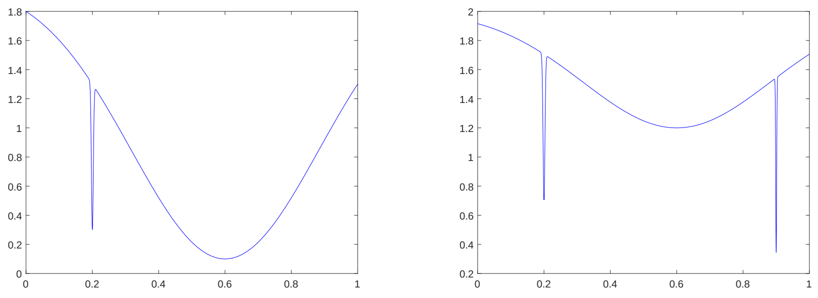


Fig. 4.1: Plot of function g, used in the definition of problems CAM1 (left) and CAM2 (right).

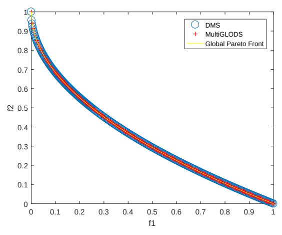
the global minimum is no longer in a narrow valley. Problem CAM2 presents three local minimums, which correspond to two local and one global Pareto fronts.

4.2. Results on the ZDT and DTLZ collections. Starting with the ZDT collection, Figure 4.2 represents the approximations to the Pareto front generated by DMS and MultiGLODS.

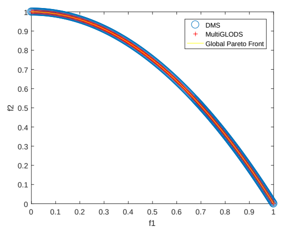
As can be observed, the results obtained with each solver are very similar, supporting the quality of the final Pareto fronts generated by MultiGLODS. For problem ZDT6, MultiGLODS presents some points far from the true global Pareto front. This is a direct consequence of the stopping criteria considered. If a higher number of function evaluations would have been allowed, eventually these points would change their status to inactive, not being part of the final approximation to the Pareto front generated by MultiGLODS. In the case of problem ZDT4, MultiGLODS is able not only to generate an approximation to the global Pareto front but also to some of the 21^9 local Pareto fronts reported in [19].

Figure 4.3 reports the results obtained with both solvers for the DTLZ collection. The behavior of MultiGLODS for problems DTLZ1 and DTLZ3 is quite peculiar. In fact, these problems present several local Pareto fronts, all parallel to the global one [10]. MultiGLODS is able to identify one of these local Pareto fronts and also to move to the global one, as we can see in Figure 4.4, where the plots are zoomed in.

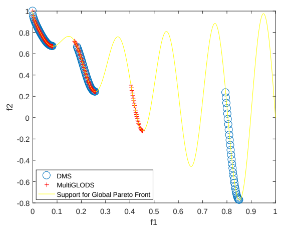
Problem DTLZ7 presents four disconnected Pareto-optimal regions. MultiGLODS was able to provided a quite good representation of each one of these areas. In problems DTLZ2 and DTLZ4, similarly to problem ZDT6, the results are a consequence of the stopping criteria, which considers a maximum number of function evaluations.



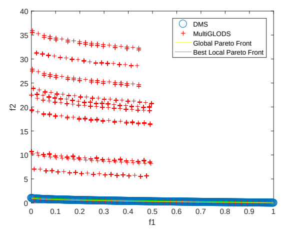
(a) ZDT1



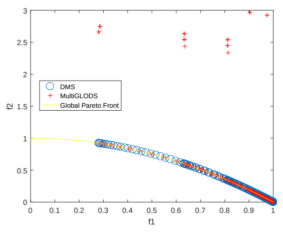
(b) ZDT2



(c) ZDT3



(d) ZDT4



(e) ZDT6

Fig. 4.2: Approximations to the Pareto front generated by DMS and MultiGLODS for the ZDT collection. For each problem, the true global Pareto fronts are represented in yellow.

4.3. Results on the multimodal problems. Having stated the quality of the Pareto fronts generated by MultiGLODS, by comparison with the results obtained with DMS in ZDT and DTLZ collections, we would like to test its capability to identify local and global Pareto fronts of a given problem.

We had already a flavor of it, with the results reported for problems ZDT4, DTLZ1, and DTLZ3. The last two problems present 3 objectives and are of dimension 7 and 12, respectively, for which a computational budget of 20000 function evaluations could somehow be limited. So we decided to use the two dimension, biobjective problems reported in [8] (Sections 4.1 and 5.1.2) and the two additional problems described in Section 4.1. Figure 4.5 presents the final approximations to the Pareto fronts obtained with DMS and MultiGLODS.

In all problems, MultiGLODS was able to identify the global Pareto front of the problems. The identification of all local Pareto fronts was well succeeded in the majority of the situations. The exception occurs in problem CAM1, where the nature of the narrow local minimum of function g prevented MultiGLODS from successfully identifying the corresponding local Pareto front.

5. Conclusions. In this paper we have proposed, analyzed, and numerically tested a new algorithm for optimizing multimodal, multiobjective, derivative-free functions.

The new directional direct-search method generalizes GLODS [5] to multiobjective optimization and confers a global behavior to DMS [6]. Multistart is used to initialize new searches, generally not conducted until the end, since they merge when start to be close to each other. A comparison radius, directly related to the step size parameter, is the keystone in this merging procedure. Points sufficiently close to each other are compared and only non-dominated points will remain active. In the end of the optimization process, the set of all active points will define the approximations to the Pareto fronts of the problem (local and global).

Under the common assumptions of directional direct-search, convergence results were derived. Numerical experiments evidence the quality of the final solutions generated by the new algorithm and its capability in identifying approximations to global and local Pareto fronts of a given problem.

REFERENCES

[1] C. Audet and J. E. Dennis Jr. Analysis of generalized pattern searches. *SIAM J. Optim.*, 13:889–903, 2003.

[2] C. Audet and J. E. Dennis Jr. Mesh adaptive direct search algorithms for constrained optimization. *SIAM J. Optim.*, 17:188–217, 2006.

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

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

[5] A. L. Custódio and J. F. A. Madeira. GLODS: Global and local optimization using direct search. *J. Global Optim.*, 62:1–28, 2015.

[6] A. L. Custódio, J. F. A. Madeira, A. I. F. Vaz, and L. N. Vicente. Direct multisearch for multiobjective optimization. *SIAM J. Optim.*, 21:1109–1140, 2011.

[7] C. Davis. Theory of positive linear dependence. *Amer. J. Math.*, 76:733–746, 1954.

[8] K. Deb. Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary Computation*, 7:205–230, 1999.

[9] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6:182–197, 2002.

[10] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable multi-objective optimization test problems. In *Congress on Evolutionary Computation (CEC'2002)*, volume 2, pages 825–830, Los Alamitos,USA, 2002. IEEE Computer Society.

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

[12] L. Kocis and W. J. Whiten. Computational investigations of low-discrepancy sequences. *ACM Trans. Math. Software*, 23:266–294, 1997.

[13] T. G. Kolda, R. M. Lewis, and V. Torczon. Optimization by direct search: New perspectives on some classical and modern methods. *SIAM Rev.*, 45:385–482, 2003.

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

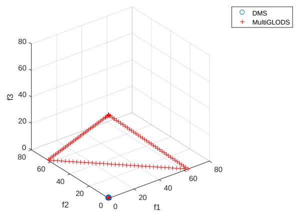
[15] R. T. Marler and J. S. Arora. Survey of multi-objective optimization methods for engineering. *Structural Multidisciplinary Optimization*, 26:369–395, 2004.

[16] M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21:239–245, 1979.

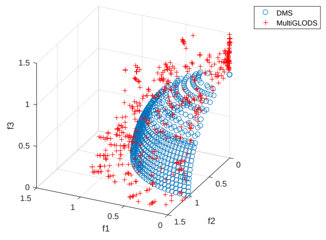
[17] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer-Verlag, Berlin, second edition, 2006.

[18] T. J. Santner, B. J. Williams, and W. I. Notz. *The Design and Analysis of Computer Experiments*. Springer-Verlag, New York, 2003.

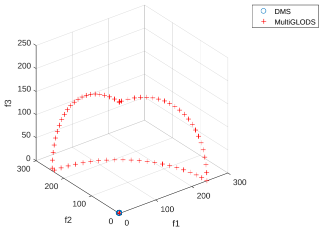
[19] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8:173–195, 2000.



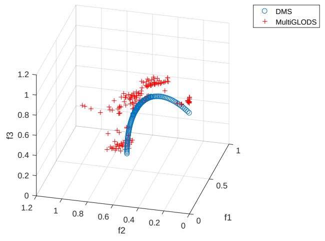
(a) DTLZ1



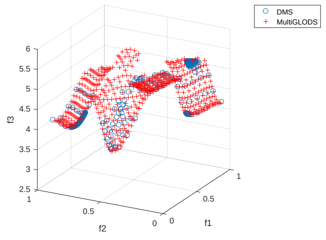
(b) DTLZ2



(c) DTLZ3

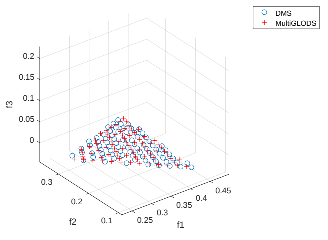


(d) DTLZ5

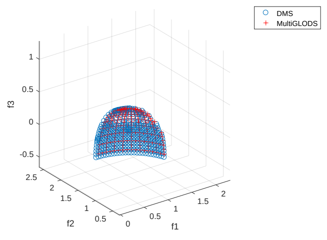


(e) DTLZ7

Fig. 4.3: Approximations to the Pareto front generated by DMS and MultiGLODS for the DTLZ collection.

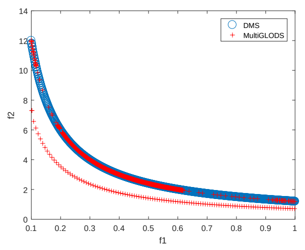


(a) DTLZ1

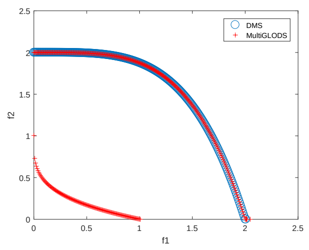


(b) DTLZ3

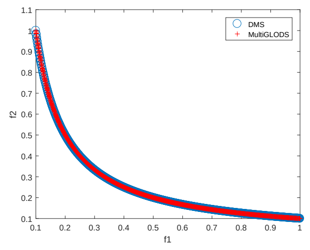
Fig. 4.4: Partial representation of the approximations to the Pareto front generated by DMS and MultiGLODS for problems DTLZ1 and DTLZ3.



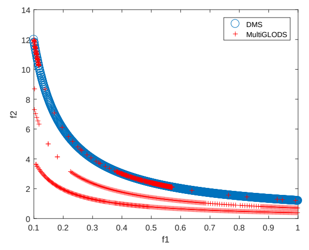
(a) Deb213, problem reported in [8], Section 4.1



(b) Deb 218, problem reported in [8], Section 5.1.2



(c) Problem CAM1



(d) Problem CAM2

Fig. 4.5: Approximations to the Pareto front generated by DMS and MultiGLODS for multimodal, two dimension, biobjective problems.