Volume 5 - 2012

ISSN 2044-8430

Computational Technology Reviews

Volume 5
2012

Volume Theme:
Engineering Optimisation

Guest Editor:
B.H.V. Topping

SAXE-COBURG PUBLICATIONS

Computational Technology Reviews

Volume Theme:
Engineering Optimisation

Guest Editor:
B.H.V. Topping

Volume 5
2012

SAXE-COBURG
PUBLICATIONS

published 2012 by
Saxe-Coburg Publications
Dun Eaglais
Station Brae, Kippen
Stirlingshire, FK8 3DY, UK

Saxe-Coburg Publications is an imprint of Civil-Comp Ltd

Computational Technology Reviews: 5
ISSN 2044-8430
ISBN 978-1-874672-59-3

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

Printed in Great Britain by Bell & Bain Ltd, Glasgow

Editorial

This fifth volume of *Computational Technology Reviews* includes a selection of papers on *engineering optimisation* which were originally presented as invited review lectures at *The Eleventh International Conference on Computational Structures Technology* (CST 2012) and *The Eighth International Conference on Engineering Computational Technology* (ECT 2012) held concurrently in Dubrovnik, Croatia from 4-7 September 2012. I am grateful to the authors and co-authors of the papers included in this volume. Their contribution to these conferences and to this volume of *Computational Technology Reviews* is greatly appreciated.

Other papers presented at these conferences are published as follows:

- *Other Invited Review Lectures from CST 2012 and ECT 2012 are published in:* Computational Technology Reviews, Volume 6, Saxe-Coburg Publications, Stirlingshire, Scotland, 2012.
- *The Invited Lectures from CST 2012 and ECT 2012 are published in:* Computational Methods for Engineering Science, B.H.V. Topping, (Editor), Saxe-Coburg Publications, Stirlingshire, Scotland, 2012.
- *The Contributed Papers from CST 2012 are published in:* Proceedings of the Eleventh International Conference on Computational Structures Technology, B.H.V. Topping, (Editor), (Book of Summaries with online delivery of full-text papers), Civil-Comp Press, Stirlingshire, Scotland, 2012.
- *The Contributed Papers from ECT 2012 are published in:* Proceedings of the Eighth International Conference on Engineering Computational Technology, B.H.V. Topping, (Editor), (Book of Summaries with online delivery of full-text papers), Civil-Comp Press, Stirlingshire, Scotland, 2012.

I am grateful to Jelle Muylle (Saxe-Coburg Publications) for his help in coordinating the publication of this volume.

UNIVERSITY OF PÉCS

HERIOT
WATT
UNIVERSITY

Professor B.H.V. Topping
University of Pécs, Hungary
& Heriot-Watt University, Edinburgh, UK

Contents

Computational Technology Reviews
Volume 5, 1-30, 2012.
doi:10.4203/ctr.5.1

SAXE-COBURG
PUBLICATIONS

Recent Developments in Derivative-Free Multiobjective Optimisation

A.L. Custódio[1], M. Emmerich[2] and J.F.A. Madeira[3,4]
[1] Department of Mathematics, CMA, FCT-UNL, Caparica, Portugal
[2] Leiden Institute for Advanced Computer Science (LIACS)
Faculty of Science, Leiden University, The Netherlands
[3] IDMEC-IST, TU-Lisbon, Portugal
[4] ISEL, Lisbon, Portugal

Abstract

In practical applications it is common to have to optimise several conflicting objective functions. Frequently, these functions are nondifferentiable or discontinuous, could be subject to numerical noise and, or be of black-box type, preventing the use of derivative-based techniques. In this paper an overview of some recent developments in derivative-free multiobjective optimisation is given. The basic concepts and ideas commonly considered for the algorithmic development in multiobjective optimisation are given and some recent classes of methods which do not make use of derivatives are reviewed. In particular, the focus is on direct search methods (DSM) of the directional type and evolutionary multiobjective optimisation (EMO).

Keywords: multiobjective optimisation, derivative-free optimisation, Pareto dominance, direct search methods, evolutionary algorithms, genetic algorithms, covariance matrix adaptation.

Acronyms: The acronyms used through the paper are listed below in alphabetical order:

BIMADS	Biobjective Mesh Adaptive Direct Search
CMA-ES	Covariance Matrix Adaptation Evolution Strategy
DFO	Derivative-free Optimisation
DMS	Direct Multisearch
DSM	Direct Search Methods
EA	Evolutionary Algorithms
EMO	Evolutionary Multiobjective Optimisation
EP	Evolutionary Programming
ES	Evolution Strategies
GA	Genetic Algorithms

HypE	Hypervolume Estimation Algorithm
KKT	Karush-Kuhn-Tucker
MADS	Mesh Adaptive Direct Search
MO-CMA-ES	Multiobjective Covariance Matrix Adaptation Evolution Strategy
MOO	Multiobjective Optimisation
MULTIMADS	Multiobjective Mesh Adaptive Direct Search
NSGA	Nondominated Sorting Genetic Algorithm
NSGA-II	Nondominated Sorting Genetic Algorithm, version 2
ROSEA	Random Objective Selection Evolutionary Algorithm
SMS-EMOA	S-Metric Selection Evolutionary Multiobjective Optimisation Algorithm
SPEA2	Strength Pareto Evolutionary Algorithm, version 2
VEGA	Vector Evaluated Genetic Algorithm

1 Introduction

In practical applications, it is common to have multiple objective functions, which need to be optimised simultaneously. Examples can be found in several distinct areas such as engineering design, feature selection, financial and management tasks [54, 55, 59, 73]. In the design phase of a new product, for example, the designer does not want only to minimise the production cost, but additionally wishes to maximise both the performance and the safety, minimise the conception time, and maximise the life time of the product.

The concept of Pareto dominance is of extreme importance in multiobjective optimisation (MOO), especially when some (or all) of the objectives are mutually conflicting. In this case, in general, there is not a single point that yields the "optimum value" for all the functions involved in the problem definition. Instead, there is a set of points, named as the *Pareto optimal set*, such that selecting one point of this set instead of another will always sacrifice the quality of at least one of the objectives (while improving, at least, another).

In MOO, the goal is to identify such a set of points, from which the designer will pick a final solution for the problem. The Pareto optimal set presents the different alternatives, none being better than the others. The choice will rely on the designer perspective of the problem.

The current paper gives an overview of some recent developments in derivative-free multiobjective optimisation. These methods are appropriated for optimising several objectives, when computing the derivatives of some of the objective functions involved is expensive, unreliable, or even impossible (which is a common situation in real applications).

Two different classes of methods are considered, representing major distinct approaches that are currently being followed to tackle these problems: direct search methods (DSM) of the directional type and evolutionary multiobjective optimisation (EMO) algorithms. For each of these classes, the most relevant algorithms are introduced, pointing out strengths and weaknesses, and mentioning some of the improvements that could be considered. The differences and similarities between the two classes are pointed out.

The paper is divided as follows. Section 2 introduces the concepts and terminology commonly considered for algorithmic development in MOO, and necessary for the following sections. Section 3 starts by presenting a brief review of DSM for single objective optimisation, moving then to MOO where details for two algorithms are provided, also considering its convergence properties. Section 4 covers derivative-free multiobjective optimisation methods with an heuristic and, or stochastic nature. The section begins with an introduction to evolutionary algorithms (EA), first in single objective optimisation, after which classical EMO algorithms are discussed. Recent trends in algorithmic development for this area can be found at the end of the section. The paper concludes with some final comments and directions for future research.

2 Concepts and terminology in multiobjective optimisation

A multiobjective optimisation (MOO) problem can be mathematically formulated as (see [56] for a more complete treatment):

$$\begin{array}{ll} \min & F(x) \equiv (f_1(x), f_2(x), \ldots, f_m(x))^\top \\ \text{s. t.} & x \in \Omega \end{array}$$

where $\emptyset \neq \Omega \subseteq \mathbb{R}^n$ represents the feasible region, and m (≥ 2) the number of extended real-value functions $f_j : \Omega \subseteq \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}, j = 1, \ldots, m$ to minimise. Recall that maximise f_j is equivalent to minimise $-f_j$. In the context of derivative-free optimisation (DFO), derivatives are not available for use, at least for one of the components of the objective function. These components could be, for instance, the result of an expensive computer simulation, sometimes subject to numerical noise, which prevents the use of numerical techniques in the approximation of the corresponding derivatives.

The feasible region, Ω, represents the set of points that verify the problem constraints. Constraints can be defined by mathematical expressions, for which derivatives could be available for use, or, in the context of black-box optimisation, be regarded as an oracle, which simply evaluates if a point is feasible or not, without providing any quantitative measure of its feasibility.

In MOO, a point in $\mathbb{R}^m$ with components corresponding to the minimum of each objective function is named as an *ideal point*. A single feasible point in $\mathbb{R}^n$ whose image under F corresponds to the ideal point does not always exist for a given MOO problem and, even if it exists, computing it is generally a very hard task. When the objective function presents several conflicting components, given a point corresponding to values of the decision variables it could be impossible to find another one which simultaneously improves the value of all the corresponding objective function components. The concept of Pareto dominance is crucial for comparing any two points lying in the feasible region.

Definition 2.1 *Let $x, y \in \Omega$ be two points corresponding to values of the decision variables of a MOO problem. The point x dominates y, being represented by $x \prec y$, if $f_j(x) \leq f_j(y)$, for all $j \in \{1, \ldots, m\}$, and $f_j(x) < f_j(y)$, for at least one index $j \in \{1, \ldots, m\}$.*

Some authors state the previous definition by considering a strict partial order in the cone $\mathbb{R}^m_+ = \{y \in \mathbb{R}^m : y \geq 0\}$. In this case, given two points x, y in Ω, we have the following equivalencies:

$$x \prec y \iff F(x) \prec_F F(y) \iff F(y) - F(x) \in \mathbb{R}^m_+ \setminus \{0\}.$$

If, for $x, y \in \Omega$, $x \not\prec y$ and $y \not\prec x$ then x and y are said to be nondominated (or incomparable) points. A subset of Ω is said to be nondominated when any pair of points in this subset is nondominated.

In single objective minimisation the goal is to find a feasible point, x_*, such that $f(x_*) \leq f(x)$, for all $x \in \Omega$, meaning a *global minimiser* of the problem. Classifying a point as a global minimiser is a difficult task, even when the corresponding point is located in an early stage of the optimisation process. Thus the research is usually focused on identifying *local minimisers, i.e.*, points x_* such that $f(x_*) \leq f(x)$, for all $x \in \Omega \cap \mathcal{N}(x_*)$, where $\mathcal{N}(x_*)$ represents a neighbourhood of x_*. The definition of Pareto dominance is usually considered when adapting these concepts to MOO.

Definition 2.2 *A point $x_* \in \Omega$ is said to be a global Pareto minimiser of F in Ω if there is no $y \in \Omega$ such that $y \prec x_*$. If there exists a neighbourhood $\mathcal{N}(x_*)$ of x_* such that the previous property holds in $\Omega \cap \mathcal{N}(x_*)$, then x_* is called a local Pareto minimiser of F.*

Rather than identifying a single point as a local Pareto minimiser, MOO algorithms approximate the set of all feasible nondominated points, $\chi_{\mathcal{P}}$, referred to as the *Pareto optimal set*. The image of $\chi_{\mathcal{P}}$ under the function F is commonly named as the *Pareto front* (or the *Pareto frontier*) of the problem.

3 Direct search methods

Excluding the class of heuristics, derivative-free algorithms for single objective optimisation are typically divided in three major groups (see [21] for more details): direct search methods (DSM), line-search algorithms for DFO, and trust-region interpolation based methods. The last two classes are inspired by derivative-based optimisation. Line-search algorithms search for a better point along a particular direction, in this case computed without considering derivatives (see, for example, the recent book by Kelley [46]). Trust-region algorithms consider local approximation models for the objective function, which are minimised inside a trust region in order to find a better point. In DFO, since derivatives are not available for use, Taylor models are replaced by interpolation based models, computed from sets of points with good geometrical properties (see, for instance, [20, 58]). In the current section focus will be on DSM since, to the authors' knowledge, it is the only class for which advances have been made in extending it to MOO.

3.1 A short review of direct search methods

Direct search methods (DSM) characterise by not considering any explicit or implicit models for the objective function, neither attempting to use or approximate its derivatives. Minimisation is achieved through an iterative process of function evaluation at finite sets of points, using the results to determine which new points should be evaluated at the next iteration. Rather than a quantitative assessment of the objective function value, it is sufficient to be able to compare any pair of points and decide which point presents a better value for the objective function.

In single objective optimisation, the term direct search was first introduced in 1961, by Hooke and Jeeves [41], but the first methods that fall into this class appeared before, in the fifties, with, for instance, the work of Fermi and Metropolis [33]. At this early stage, the algorithmic development was mainly empirical, driven by practical applications, and supported by geometrical considerations. Several algorithms were proposed, with probably the most well-known example being the simplex algorithm of Nelder and Mead [57]. For a survey on DSM see, for instance, Kolda *et al.* [48].

It is only in the nineties, with the PhD thesis by Torczon and the subsequent works [66, 67], that the first convergence theory was established for some algorithms belonging to this class, raising the interest of the numerical optimisation community. Since then, there has been an intensive and fruitful period of research, covering both aspects of theoretical development and practical applications.

Audet and Dennis [1] generalised the work of Torczon [67], by proposing a general framework for the class of DSM of the directional type, also designated as pattern search methods. Basically, they proposed to split each iteration of these algorithms in a search step and a poll step. The first is optional for ensuring the convergence, being typically used to improve the numerical performance. The implementation, at this step, of distinct strategies causes different algorithmic instances, all of them belonging to the class of DSM (see, for example, Custódio *et al.* [23] or Vaz and Vicente [68].

The poll step consists of a local search around the current iterate, by testing scaled poll directions associated with a positive basis or a positive spanning set. Positive spanning sets are sets of vectors, whose nonnegative linear combinations generate a given set. Positive bases are minimal positive spanning sets. Given any vector, a positive spanning set for $\mathbb{R}^n$ is guaranteed to have at least one element within a 90° angular distance of the considered vector. In the context of DFO, where the location of the gradient of the objective function is unknown, even when it exists, this property is crucial to ensure the algorithmic convergence. For more details about the properties of these sets of directions see Davis [24]. The scaling of the poll directions is achieved by considering a step size parameter.

At a given iteration, once a better point is found, the iteration is declared as successful. If the better point is found at the search step then the poll step could be omitted. When both steps fail to generate a better point, the iteration is named as unsuccessful. At unsuccessful iterations, additionally to the function evaluation performed at the search step, all the poll directions have been tested. The step size is increased or maintained at successful iterations and obligatorily reduced at unsuccessful ones.

Different algorithmic instances could also result from considering different globalisation strategies, associated with the type of decrease required for the objective function value, when deciding if a better point was found. If only simple decrease is required, the update of the step size parameter and the computation of the poll directions follow strict rules in order to ensure that all the evaluated points lie in an implicit mesh, mathematically defined as an integer lattice. Also, the points evaluated at the search step need to be restricted to this implicit mesh, or be projected on it. Requiring sufficient decrease relaxes these conditions.

A summarised algorithmic description of a basic DSM of the directional type is given in Algorithm 3.1.

Algorithm 3.1 DSM of directional type for single objective optimisation

```
Initialisation
    Choose $x_0 \in \Omega$ with $f(x_0) < +\infty$, an initial step size $\alpha_0 > 0$, $0 < \beta_1 \leq \beta_2 < 1$,
    and $\gamma \geq 1$. Let $\mathcal{D}$ be a (possibly infinite) set of positive spanning sets. Set $k = 0$.

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

        1. Search step: Evaluate $f$ at a finite set of points $\{x_s : s \in S\}$. If a better
           point $x_s$ is found, set $x_{k+1} = x_s$, declare the iteration as successful and
           skip the poll step.

        2. Poll step: Choose a positive spanning set $D_k$ from the set $\mathcal{D}$. Evaluate $f$ at
           the set of poll points $P_k = \{x_k + \alpha_k d : d \in D_k\}$, stopping the evaluating
           process if a better point is found. In this case, set $x_{k+1} = x_k + \alpha_k d$ and
           declare the iteration as successful. Otherwise, declare the iteration as
           unsuccessful and set $x_{k+1} = x_k$.

        3. Step size parameter update: If the iteration was successful then maintain
           or increase the step size parameter: $\alpha_{k+1} \in [\alpha_k, \gamma\alpha_k]$. Otherwise decrease
           the step size parameter: $\alpha_{k+1} \in [\beta_1\alpha_k, \beta_2\alpha_k]$.
```

The convergence analysis proposed by Torczon [67] for pattern search methods assumed the continuity of the derivatives of the objective function, even if these derivatives were not known or used in the algorithmic definition. This could be a very strong assumption, considering the features of the practical applications to solve. Audet and Dennis [1] extended this convergence analysis by only requiring Lipschitz continuity of the objective function. For that, they have recurred to Clarke's calculus [18] and its generalised directional derivatives, adapted by Jahn [45] to the constrained case. The definition of Clarke-Jahn for a generalised directional derivative is here recalled, since it will be mentioned in the following subsections.

For a function f Lipschitz continuous near a point x, and d belonging to the interior of the tangent cone to Ω at x, $T_\Omega(x)$, the Clarke-Jahn generalised directional derivative, computed at x in the direction d, is defined as:

$$f^\circ(x;d) = \limsup_{\substack{y \to x,\, y \in \Omega \\ t \downarrow 0,\, y + td \in \Omega}} \frac{f(y+td) - f(y)}{t}.$$

For directions belonging to the border of $T_\Omega(x)$ the Clarke-Jahn generalised directional derivatives result from taking limits as $f^\circ(x;d) = \lim_{v \in int(T_\Omega(x)), v \to d} f^\circ(x;v)$ (see Audet and Dennis [2]).

Audet and Dennis [2] have also proposed a new class of DSM of directional type, namely mesh adaptive direct search (MADS), for which convergence is guaranteed for general constrained problems. In this case, the algorithm makes use of a set of poll directions which is asymptotically dense in the unit sphere. A step further was taken by Vicente and Custódio [70], by extending the convergence analysis to particular types of discontinuous functions (assuming that the objective function is directionally Lipschitz with respect to a particular limit direction).

Nowadays, the general theory, supporting the convergence properties of DSM of directional type, is reasonably well understood. The major challenge is to improve and analyse the efficiency of algorithms, allowing to tackle higher dimensional problems, and also moving to MOO. As mentioned before, carefully designed search steps could be a tool for improving the numerical performance of the solvers, but, given the nature of the poll step, parallel implementations should also be considered.

One of the first parallel implementations of a DSM of the directional type was asynchronous parallel pattern search [42]. The use of asynchronous strategies could be relevant when function evaluation presents considerably different times, for instance due to distinct loads and, or speed of processors or distinct computational effort to converge a numerical simulation. Following this work, several other serial implementations were parallelised (see, for instance, NOMAD [27] or PSwarm [69]), but this topic is still the subject of intensive research.

DSM for MOO are in the beginning of their development. Zhong *et al.* [74] proposed an empirical algorithm based on compass search, but for which no convergence analysis was provided. To our knowledge, only two DSM were proposed for general derivative-free multiobjective optimisation, namely multiobjective mesh adaptive direct search (MULTIMADS) [4] and direct multisearch (DMS) [22], which will be the subject of the following subsections.

3.2 Multiobjective mesh adaptive direct search

In MOO, when the user is able to prioritise the different objectives defining the problem, an aggregation function could be considered, combining the several components of the objective function into a single one. One possible approach to define this aggregation function is to consider a weighted geometrical mean.

Let $u = \left(\max_{x \in \mathcal{X_P}} f_1(x), \max_{x \in \mathcal{X_P}} f_2(x), \ldots, \max_{x \in \mathcal{X_P}} f_m(x) \right)^\top$ be the *Nadir point* of the problem and $\lambda_j, j \in \{1, \ldots, m\}$ be fixed weights. The idea is to maximise the weighted geometrical mean of the differences between the components of the objective function and this *reference point*:

$$\begin{array}{ll} \max & \Pi_{j=1}^{m} (u_j - f_j(x))^{\lambda_j} \\ \text{s. t.} & f_j(x) \leq u_j, j \in \{1, 2, \ldots, m\} \\ & x \in \Omega \end{array}$$

If all the components of the objective function are convex, the solution of the previous problem would generate a point in the Pareto front, but would also have required the addition of m general constraints to the original problem (namely, $f_j(x) \leq u_j, j \in \{1, 2, \ldots, m\}$). It is noted that, typically, it is difficult to compute the Nadir point of a MOO problem. Inspired by this approach, Audet *et al.* [3] developed BIMADS (biobjective mesh adaptive direct search), a DSM suited for biobjective optimisation.

BIMADS computes an approximation to the Pareto front of a given biobjective problem by solving a sequence of single objective DFO problems, preserving important features of the original objective function. Each of these subproblems is defined through an aggregation function $\Psi_r(x) = \phi_r(f_1(x), f_2(x), \ldots, f_m(x))$, where $\phi_r : \mathbb{R}^m \rightarrow \mathbb{R}$ depends on a *reference point* $r \in \mathbb{R}^m$.

The function Ψ_r should present the following characteristics: i) whenever all the components of the objective function are Lipschitz continuous near a feasible point x, Ψ_r should also be Lipschitz continuous near x; ii) if all the components of the objective function are Lipschitz continuous near $x \in \Omega$ with $F(x) < r$ componentwise, and if d belongs to the tangent cone to the feasible region computed at x, whenever $f_j^\circ(x; d) < 0$, for all $j = 1, \ldots, m$ then $\Psi_r^\circ(x; d) < 0$. These properties would allow one to inherit the convergence results derived for the aggregation function to $F = (f_1(x), f_2(x), \ldots, f_m(x))$, the function defining the original MOO problem.

Audet *et al.* [3] proposed two different aggregation functions, which define different single objective formulations, for use in biobjective optimisation. One of these aggregation functions resembles the weighted geometrical mean approach, without the m additional constraints, which are implicitly included in the single objective function. These single objective formulations would be solved by a DFO solver, considering increasingly stringent stopping criteria. Audet *et al.* [3] selected MADS [2] as the solver, but other approaches could be taken.

At the beginning of the iterative process, MADS is used to minimise each component of the objective function, inside the feasible region. The points evaluated during the course of the optimisation are used to initialise a list of feasible nondominated points. This list is updated at each iteration by adding new feasible points, removing dominated ones, and sorting the feasible nondominated points in ascending order of f_1 and descending value for f_2. It represents the current approximation to the Pareto front of the problem.

At each iteration, the ordering strategy allows one to easily access the size of the gaps between consecutive points lying in the approximation of the Pareto front and to select a point corresponding to the largest ones. This point will be used to compute a reference point in the objective function space, which will be used in the single objective formulation of the biobjective problem. Again, this formulation will be solved with MADS. The underlying idea is to achieve a uniform coverage of the Pareto front, even when it is represented by a nonconvex or a discontinuous function. If the cardinality of the list of points equals one then each component of the objective function is again minimised, with a stringent stopping criteria. At the end of each iteration, the list of feasible nondominated points is updated with all the points evaluated during the optimisation process. Algorithm 3.2 presents a simplified description of BIMADS.

Algorithm 3.2 Biobjective mesh adaptive direct search

Initialisation

Use MADS to solve $\min_{x\in\Omega} f_j(x)$, $j \in \{1,2\}$ *and use the evaluated points to initialise a list of feasible nondominated points,* L_0. *Order* L_0 *by increasing order of* f_1 *and decreasing order of* f_2. *Set* $k = 0$.

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

1. **Selection of a reference point:** *If* $|L_k| = 1$, *use MADS to again solve* $\min_{x\in\Omega} f_j(x)$, $j \in \{1,2\}$, *with a stringent stopping criteria and skip the next step. Otherwise, compute a reference point based on the largest gaps in* L_k.
2. **Single objective formulation minimisation:** *Use the reference point to compute a single objective formulation,* $\min_{x\in\Omega} \Psi_r(x)$, *for the biobjective optimisation problem. Use MADS to solve the single objective formulation.*
3. **Update of the list of feasible nondominated points:** *Use the feasible evaluated points to update* L_k *by adding nondominated points and removing dominated ones. Order* $L_{k+1} = L_k$ *by increasing order of* f_1 *and decreasing order of* f_2.

Using generalised directional derivatives, Audet *et al.* [3] established an hierarchy of stationarity results for BIMADS, one of which is reproduced in the following theorem.

Theorem 3.1 (Theorem 4.3 in [3]) *Let F be Lipschitz continuous near a limit point $x_* \in \Omega$, generated by MADS when applied to a single objective formulation $\min\limits_{x\in\Omega} \Psi_r(x)$ of the biobjective optimisation problem, at some reference point $r \in \mathbb{R}^2$. Assume that $int(T_\Omega(x_*)) \neq \emptyset$. Then, for any $d \in T_\Omega(x_*)$ there exists $j \in \{1,2\}$ such that $f_j^\circ(x_*; d) \geq 0$.*

This stationarity result, which can be regarded as a generalisation of the Karush-Kuhn-Tucker (KKT) conditions, states that there is no direction in the tangent cone which is simultaneously descent for both components of the objective function. Thus, it is a necessary condition for a point to be a Pareto (local or global) minimiser. If strict differentiability is assumed for both components of the objective function (meaning that the corresponding Clarke generalised gradient is a singleton), then the previous theorem can be recast as a KKT-type stationarity result, using the gradient vectors.

The ordering strategy considered for the list of feasible nondominated points is crucial when identifying the reference points to be used in the single objective formulations. Nevertheless, it is not easily generalised to more than two objectives. To allow the solution of MOO problems with more than two components in the objective function, Audet *et al.* [4] had to consider a new set, the tangent hull, from which the reference points would be selected.

Definition 3.1 *Let z_* be the minimum value of $z = \sum\limits_{j=1}^{m} s_j f_j(x)$, where s_j are positive scaling factors, for $j \in \{1, 2, \ldots, m\}$, and let $B = \{\beta \in \mathbb{R}^m : \sum_{j=1}^m \beta_j = 1, \beta_j \geq 0\}$. The set $\{z_* \beta I_m : \beta \in B\}$, where I_m denotes the identity matrix of order m, is referred to as the tangent hull.*

At each iteration of MULTIMADS, the proposed solver for MOO, a convex combination vector $\beta \in B$ is generated to select a reference point, r, from the tangent hull, which will be used to define a single objective formulation. The authors propose a new single objective formulation which provides a more flexible optimality condition, by allowing the selection of a reference point anywhere in the objective function space. Again, MADS will be used to solve this single objective DFO problem and the evaluated points, generated during the course of the optimisation process, are used to update the list of feasible nondominated points. Algorithm 3.3 summarises this procedure.

Algorithm 3.3 Multiobjective mesh adaptive direct search

Initialisation

Use MADS to compute x_{j_*}, by solving $\min\limits_{x\in\Omega} f_j(x)$, $j \in \{1, \ldots, m\}$ and let $F_* = (f_1(x_{1_*}), \ldots, f_m(x_{m_*}))$. Redefine $f_j = f_j - F_{*_j}$, for $j \in \{1, \ldots, m\}$. Use MADS to compute, z_*, by solving $\min_{x\in\Omega} \sum_{j=1}^m s_j f_j(x)$, where s_j are positive scaling factors, ensuring that the components of the objective function have *similar magnitudes. Use the evaluated points to initialize a list of feasible nondominated points, L_0. Set $k = 0$.*

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

1. **Selection of a reference point:** *Generate a reference point, $r = F_* + z_* \beta I_m$, belonging to the tangent hull.*
2. **Single objective formulation minimisation:** *Use the reference point to compute a single objective formulation, $\min\limits_{x \in \Omega} \Psi_r(x)$, for the multiobjective optimisation problem. Use MADS to solve the single objective formulation.*
3. **Update of the list of feasible nondominated points:** *Use the feasible evaluated points to update L_k by adding nondominated points and removing dominated ones.*

Stationarity results, similar to the ones derived for the biojective optimisation problem, can be stated.

Theorem 3.2 (Theorem 3.4 in [4]) *Let F be Lipschitz continuous near a limit point $x_* \in \Omega$, generated by MADS when applied to a single objective formulation $\min_{x \in \Omega} \Psi_r(x)$ of the multiobjective optimisation problem, at some reference point $r \in \mathbb{R}^m$. Assume that $int(T_\Omega(x_*)) \neq \emptyset$. Then, for any $d \in T_\Omega(x_*)$ there exists $j \in \{1, 2, \ldots, m\}$ such that $f_j^\circ(x_*; d) \geq 0$.*

In the original papers, where BIMADS and MULTIMADS were proposed, implementations were tested in some academic problems. The codes have also been used to solve two real applications. BIMADS was used in the optimisation of a portfolio selection problem in the presence of skewness (see [73]) and MULTIMADS was tested in the optimisation of a styrene process (see [4]).

3.3 Direct multisearch for multiobjective optimisation

Custódio *et al.* [22] did not want to aggregate any components of the objective function or define priorities for the several objectives involved. The goal was to generalise all DSM of directional type to MOO. Thus, each iteration of direct multisearch (DMS) is organised around a search step and a poll step. Like in the works of Audet *et al.* [3,4], the algorithm keeps a list of feasible, nondominated points, which represents the current approximation to the Pareto front and from which poll centers will be chosen. At each iteration, the new feasible evaluated points are added to this list and the dominated ones are removed. An iteration is said to be successful if the iterate list changes, meaning that a new feasible nondominated point was found. Otherwise, the iteration is declared as unsuccessful.

Similarly to single objective optimisation, the search step is optional and is not required for ensuring the convergence of the algorithm. It could be used, for instance, to improve the numerical performance or to disseminate points across the Pareto front. When no new feasible nondominated point is found at the search step, the poll step will be executed. Convergence properties of the algorithm result from it. The algorithm performs a local search around a selected poll center by testing directions belonging to a positive basis, or a positive spanning set, scaled by a step size parameter. Again, new feasible nondominated points are added to the current iterate list, with the dominated ones being removed.

As in single objective optimisation, at the end of an unsuccessful iteration the corresponding step size parameters are decreased. For successful iterations the step sizes are kept constant, or can even be increased.

Algorithm 3.4 corresponds to a short and concise description of DMS. Details about the use of globalisation strategies, like considering implicit meshes or imposing a sufficient decrease condition on the objective function value to accept a new point, are omitted.

Algorithm 3.4 Direct multisearch for MOO

Initialisation

Choose an initial step size parameter $\alpha_0 > 0$, $0 < \beta_1 \leq \beta_2 < 1$, *and* $\gamma \geq 1$. *Let* $\mathcal{D}$ *be a (possibly infinite) set of positive spanning sets. Initialise the list of feasible nondominated points and corresponding step size parameters* $L_0 = \{(x_i; \alpha_i) : i \in I\}$. *Set* $k = 0$.

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

1. **Selection of an iterate point:** *Select an iterate point* $(x_k; \alpha_k) \in L_k$ *as the current poll center and step size parameter.*
2. **Search step:** *Evaluate* F *at a finite set of points* $\{x_s : x_s \in S\}$. *Use the feasible evaluated points to update* L_k *by adding nondominated points and removing dominated ones. If* L_k *changed, declare the iteration as successful and skip the poll step.*
3. **Poll step:** *Choose a positive spanning set* D_k *from the set* $\mathcal{D}$. *Evaluate* F *at the set of poll points* $P_k = \{x_k + \alpha_k d : d \in D_k\}$. *Use the feasible evaluated points to update* L_k *by adding nondominated points and removing dominated ones. If* L_k *changed, declare the iteration as successful. Otherwise, declare the iteration as unsuccessful and set* $L_{k+1} = L_k$.
4. **Step size parameter update:** *If the iteration was successful then maintain or increase the corresponding step size parameters. Otherwise decrease the corresponding step size parameters.*

This algorithmic framework is very general and encompasses several variants. Different algorithmic instances result from considering different strategies for the initialisation of the iterate list (line sampling, random sampling, Latin hypercube sampling,