Recent Developments in Derivative-free Multiobjective Optimization

A. L. Custódio [*] M. Emmerich [†] J. F. A. Madeira [‡]

April 5, 2012

Abstract

In practical applications it is common to have several conflicting objective functions to optimize. 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 we give an overview of some recent developments in Derivative-free Multiobjective Optimization. We introduce the basic concepts and ideas commonly considered for the algorithmic development in Multiobjective Optimization and review some recent classes of methods which do not make use of derivatives. In particular, we will focus on Direct Search Methods (DSM) of directional type and Evolutionary Multiobjective Optimization (EMO).

Keywords: Multiobjective optimization, Derivative-free optimization, Pareto dominance, direct search methods, evolutionary algorithms, genetic algorithms, covariance matrix adaptation.

[*] Department of Mathematics, FCT-UNL-CMA, Quinta da Torre, 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 PEst-OE/MAT/UI0297/2011 (CMA) and the grant PTDC/MAT/098214/2008.

[†] Leiden Institut for Advanced Computer Science (LIACS), Faculty of Science, Leiden University, Niels Bohrweg 1, 2333-CA The Netherlands, (emmerich@liacs.nl). Support for this project was provided by the Deliver EUREKA Project, Agentschap NL and by FCT under the grant SFRH/BPD/65330/2009.

[‡] IDMEC-IST, TU-Lisbon, Av. Rovisco Pais, 1040-001 Lisboa, Portugal and ISEL, Rua Conselheiro Emídio Navarro, 1, 1959-007 Lisboa (jaguilar@dem.ist.utl.pt). Support for this author was provided by ISEL, IDMEC-IST, and FCT-POCI 2010.

1 Acronyms

Below we list, in alphabetic order, the acronyms used through the paper.

BIMADS	Biobjective Mesh Adaptive Direct Search
CMA-ES	Covariance Matrix Adaptation Evolution Strategy
DFO	Derivative-free Optimization
DMS	Direct Multisearch
DSM	Direct Search Methods
EA	Evolutionary Algorithms
EMO	Evolutionary Multiobjective Optimization
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 Optimization
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 Optimization Algorithm
SPEA2	Strength Pareto Evolutionary Algorithm, version 2
VEGA	Vector Evaluated Genetic Algorithm

2 Introduction

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

The concept of Pareto dominance is of extreme importance in Multiobjective Optimization (MOO), especially when some (or all) of the objectives are mutually conflicting. In this case, we do not have, in general, a single point that yields the "optimum value" for all the functions involved in the problem definition. Instead, we have 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 Optimization. These methods are appropriated for optimizing 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).

We consider two different classes of methods, representing major distinct approaches that are currently being followed to tackle these problems: Direct Search Methods (DSM) of directional type and Evolutionary Multiobjective Optimization (EMO) algorithms. For each of these classes, we introduce the most relevant algorithms, pointing out strengths and weaknesses, and mentioning some of the improvements that could be considered. We also remark the differences and similarities between the two classes.

The paper is divided as follows. Section 3 introduces the concepts and terminology commonly considered for algorithmic development in MOO, and necessary for the following sections. Section 4 starts by presenting a brief review of DSM for single objective optimization, moving then to MOO where details for two algorithms are provided, also considering its convergence properties. Section 5 respects to Derivative-free Multiobjective Optimization methods with an heuristic and/or stochastic nature. The section begins with an introduction to Evolutionary Algorithms (EA), first in single objective optimization, 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.

3 Concepts and Terminology in Multiobjective Optimization

A Multiobjective Optimization (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 $f_j : \Omega \subseteq \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}, j = 1, \ldots, m$ to minimize. Recall that maximize f_j is equivalent to minimize $-f_j$. In the context of Derivative-free Optimization (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 optimization, 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 3.1 *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 $x, y \in \Omega$, $x \nprec y$ and $y \nprec x$ 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 minimization the goal is to find a feasible point, x_*, such that $f(x_*) \leq f(x)$, for all $x \in \Omega$, meaning a *global minimizer* of the problem. Classifying a point as a global minimizer is a difficult task, even when the corresponding point is located in an early stage of the optimization process. Thus the research is usually focused in identifying *local minimizers*, 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 neighborhood of x_*. The definition of Pareto dominance is usually considered when adapting these concepts to MOO.

Definition 3.2 *A point $x_* \in \Omega$ is said to be a global Pareto minimizer of F in Ω if there is no $y \in \Omega$ 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.*

Rather than identifying a single point as a local Pareto minimizer, 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.

4 Direct Search Methods (DSM)

Excluding the class of heuristics, Derivative-free algorithms for single objective optimization 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 optimization. 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 of Kelley [46]). Trust-region algorithms consider local approximation models for the objective function, which are minimized 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 we will focus on DSM since, to our knowledge, it is the only class for which advances have been made in extending it to MOO.

4.1 A Short Review of Direct Search Methods (DSM)

Direct Search Methods (DSM) characterize by not considering any explicit or implicit models for the objective function, neither attempting to use or approximate its derivatives. Minimization 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 optimization, 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, being the simplex algorithm of Nelder and Mead [57] probably the most well-known example. For a survey on DSM see, for instance, Kolda et al. [48].

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

Audet and Dennis [1] generalized the work of Torczon [67], by proposing a general framework for the class of DSM of 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 that 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 obligatory reduced at unsuccessful ones.

Different algorithmic instances could also result from considering different globalization strategies, associated to 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 summarized algorithmic description of a basic DSM of directional type is given in Algorithm 4.1.

Algorithm 4.1 DSM of directional type for single objective optimization

Initialization

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 generalized directional derivatives, adapted by Jahn [45] to the constrained case. We recall here the definition of Clarke-Jahn for a generalized directional derivative, 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 generalized 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 generalized 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, we would say that 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 we have mention 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 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 parallelized (see, for instance, NOMAD [27] or PSwarm [69]), but this topic is still subject of intensive research.

DSM for MOO are in the beginning of its 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 Optimization, namely *Multiobjective Mesh Adaptive Direct Search (MULTIMADS)* [4] and *Direct Multisearch (DMS)* [22], which will be the subject of the following subsections.

4.2 Multiobjective Mesh Adaptive Direct Search (MULTIMADS)

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

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

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, $f_j(x) \leq u_j, j \in \{1, 2, \ldots, m\}$). We also note 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 optimization.

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 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 optimization. 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 solver, but other approaches could be taken.

At the beginning of the iterative process, MADS is used to minimize each component of the objective function, inside the feasible region. The points evaluated during the course of the optimization are used to initialize 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 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 an 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 minimized, 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 optimization process.

Algorithm 4.2 presents a simplified description of BIMADS.

Algorithm 4.2 Biobjective Mesh Adaptive Direct Search (BIMADS)

```
Initialization
    Use MADS to solve $\min_{x \in \Omega} f_j(x)$, $j \in \{1,2\}$ and use the evaluated points to initialize 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 minimization: Use the reference point to compute a single objective formulation, $\min_{x \in \Omega} \Psi_r(x)$, for the biobjective optimization 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 generalized directional derivatives, Audet et al. [3] established an hierarchy of stationarity results for BIMADS, one of which is reproduced in the following theorem.

Theorem 4.1 (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 $\min_{x \in \Omega} \Psi_r(x)$ of the biobjective optimization 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 generalization 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) minimizer. If we assume strict differentiability for both components of the objective function (meaning that the corresponding Clarke generalized 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 generalized to more than two objectives. For allowing to solve MOO problems with more that 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 4.1 *Let z_* be the minimum value of $z = \sum_{j=1}^{m} s_j f_j(x)$, 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 refereed 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 to select 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 optimization process, are used to update the list of feasible nondominated points. Algorithm 4.3 resumes this procedure.

Algorithm 4.3 Multiobjective Mesh Adaptive Direct Search (MULTIMADS)

Initialization

Use MADS to compute x_{j_*}, by solving $\min_{x \in \Omega} f_j(x)$, $j \in \{1, \ldots, m\}$ 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 minimization:** *Use the reference point to compute a single objective formulation, $\min_{x \in \Omega} \Psi_r(x)$, for the multiobjective optimization 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 optimization problem, can be stated.

Theorem 4.2 (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 optimization 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 optimization of a portfolio selection problem in the presence of skewness (see [73]) and MULTIMADS was tested in the optimization of a styrene process (see [4]).

4.3 Direct Multisearch (DMS) for Multiobjective Optimization

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 generalize all DSM of directional type to MOO. Thus, each iteration of *Direct Multisearch (DMS)* is organized 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 optimization, the search step is optional and it 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, being removed the dominated ones.

As in single objective optimization, 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 4.4 corresponds to a short and concise description of DMS. Details about the use of globalization strategies, like considering implicit meshes or imposing a sufficient decrease condition on the objective function value to accept a new point, are omitted.

Algorithm 4.4 Direct Multisearch (DMS) for MOO

Initialization

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. Initialize 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 initialization of the iterate list (line sampling, random sampling, Latin hypercube sampling, or others), from the definition of a search step and from implementing an ordering strategy for the list of feasible, nondominated points, before selecting the new poll center, just to name a few. Concerning the latter, in an attempt to reduce the gaps between consecutive points lying in the current approximation to the Pareto front, the authors proposed the use of a spread metric. Gaps are measured componentwise and points are ordered according to the highest values of it.

Again, considering generalized directional derivatives, an hierarchy of stationarity results is derived for DMS. In the next theorem, we reproduce one of the results.

Theorem 4.3 (Theorem 4.9 in [22]) *Let F be Lipschitz continuous near a limit point $x_* \in \Omega$ generated by DMS, and assume that $int(T_\Omega(x_*)) \neq \emptyset$. If the set of refining directions for x_* is dense in $T_\Omega(x_*)$, then for any $d \in T_\Omega(x_*)$ there exists $j \in \{1, 2, \ldots, m\}$ such that $f_j^\circ(x_*; d) \geq 0$.*

Refining directions are limits of normalized directions, associated with feasible poll points, corresponding to unsuccessful iterations (for more details, see Audet and Dennis [2]).

In [22] the authors have also performed an intensive numerical comparison between DMS and eight other solvers commonly used in Derivative-free Multiobjective Optimization. In particular, it was tested BIMADS (see Subsection 4.2) and NSGA-II (see Subsection 5.3), the latter a MOO solver based on genetic algorithms. The test set included 100 bound constrained optimization problems, among of which there were convex, nonconvex, and discontinuous Pareto fronts. As comparison indicators the authors considered the purity metric (see [9]), which measures the percentage of points generated by a given solver in a reference Pareto front. For each problem, the reference Pareto front is built by gathering the results obtained by running the totality of the solvers considered in the computational experiments and removing all the dominated points. In [22], the authors have also considered two spread metrics, one of them generalizing the spread metric proposed in [26] to objective functions with more than two components. For the metrics considered, DMS has proved to be highly competitive with the remaining solvers, even without the implementation of a search step.

5 Evolutionary Multiobjective Optimization (EMO)

Evolutionary Algorithms (EA) are stochastic search and machine learning heuristics, inspired by theories of biological evolution, most prominently by the so-called modern evolutionary synthesis that combines natural selection and genetics (see [29]). Traditionally, EA have been categorized into three subfields: Genetic Algorithms (GA) [36] (mainly binary representations), Evolution Strategies (ES) [14] (mainly continuous representations) and Evolutionary Programming (EP) [34] (arbitrary representations). Nowadays, the boundaries between these subfields became more fluid and the methods are often grouped together using the term Evolutionary Algorithms (see [7]).

EA are used for machine learning and as simulation models in biology [36], but their major field of application is surely systems optimization. This class of methods does not require derivatives of the functions defining the problem and it is relatively robust and flexible for solving nonlinear optimization problems, due to the stochastic search operators involved in the algorithmic definition. Empirical results suggest that on nonsmooth and nonconvex problems EA can outperform classical deterministic methods (see [65]). For continuous optimization, advanced EA, such as the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [38], are nowadays considered to be state-of-the-art methods.

Evolutionary Multiobjective Optimization (EMO) applies EA for the solution of MOO problems. After first pioneering work in the eighties [64] and in the early nineties [35,49], these methods received strong attention during the last two decades [19]. What has emerged as a side-branch of the evolutionary computation community is today considered as a more independent field of research, at the intersection of Evolutionary Algorithms, Metaheuristics, and Multicriteria Decision Making. In addition, EMO methods became popular for solving practical problems in multiobjective optimization (see, e.g., Branke et al. [16]). The major focus in EMO has been the computation of finite approximations to Pareto optimal sets and Pareto fronts. Some algorithms, such as the second versions of the Nondominated Sorting Genetic Algorithm (NSGA-II) [26] and the Strength Pareto Evolutionary Algorithm (SPEA2) [76], became popular techniques for this task and will be reviewed here. We will also review more recent trends, exemplified by algorithms such as the hypervolume-based S-metric Selection EMO algorithm (SMS-EMOA) [30] and the Multiobjective CMA-ES (MO-CMA-ES) [43], the latter specially designed for solving continuous optimization problems.

5.1 Single Objective Evolutionary Algorithms (EA)

In EA mathematical objects are often referred to by biological metaphors, such as individual (solution), population (tuple of solutions), and fitness (objective function value or rank of an individual, within a population).

This adds a layer of abstraction to the algorithmic description, which can present different instantiations. As an example, an individual could represent a point in $\{0, 1\}^n$, a point in $\mathbb{R}^n$, or even an instance of a dynamic data-structure. In the sequel, however, the focus will be on continuous search spaces, meaning that individuals are represented as points in $\mathbb{R}^n$ and populations are considered to be fixed-cardinality multisets of individuals.

Algorithm 5.1 describes a generic EA for single objective optimization. Using the subsequently discussed genetic operators, the stochastic transition is designed in such a way that the expected value of the objective function of the population individuals improves over time. Within the framework of EA one distinguishes between parent populations X_t (of size $\mu > 0$) and offspring populations O_t (of size $\lambda > 0$), at iterations $t \in \{0, 1, 2, \dots\}$. The iteration counter is denoted by t to emphasize that $\{\mathrm{X}_t\}_{t\in\mathbb{N}}$ could be regarded as the trajectory in $(\mathbb{R}^n)^\mu$ of a stochastic process indexed in time. The offspring population, O_t, is an intermediate population that is generated by the variation operators (recombination and mutation) from the individuals of the parent population X_t. The next generation parent population, X_{t+1}, will be selected from the union of the parent and offspring populations, considering the fitness of the individuals.

Algorithm 5.1 Generic Evolutionary Algorithm (EA)

```
Initialization
    Define all required algorithmic parameters (parent population size ($\mu$), number
    of recombination partners ($\rho$), number of offsprings ($\lambda$), mutation strength ($\sigma$),
    etc.) and set the generation (iteration) counter $t = 0$. Initialize the parent
    population, $\mathrm{X}_0$, and assign a fitness to each of its $\mu$ individuals, based on the
    corresponding objective function value (and, possibly, other criteria).

 For $t = 0, 1, 2, \dots$
```

- **Mating selection:** Select ρ individuals in X_t, that will serve as 'templates' for the individuals in X_{t+1}.
- **Recombination:** Combine the information of the selected individuals (e.g. by means of random crossover or of averaging) in order to create a new population, O_t, of λ offspring individuals.
- **Mutation:** Perturb (some of) the offspring individuals in O_t by means of small random modifications.
- **Fitness assignment:** Evaluate the fitness of each offspring individual, considering the corresponding objective function value (and, possibly, other criteria).
- **Environmental selection:** Select individuals from the offspring population O_t (and, possibly, also from the parent population X_t), in order to form the next generation parent population X_{t+1} (of size μ).

The initialization of an EA is typically done in an uniform random manner, with respect to the space of feasible solutions Ω. Important instantiations of *mating selection* are:

- **Random selection:** randomly selecting parents, in which case the same parent can be repeatedly chosen. This selection type is typically applied in ES [14].
- **Tournament selection:** two (or more) individuals are selected randomly from the parent population and the individual with the best fitness will be kept. This selection type is typically applied in EP [34].
- **Roulette wheel selection:** Individuals are selected with a probability or frequency that is proportional to their fitness or rank in the population. This selection type is typically applied in GA [36].

Besides the selection of individuals to be recombined and/or mutated, the environmental selection is used to compute the population X_{t+1} from the offspring population O_t, and also possibly from X_t. The $(\mu + \lambda)$-selection and the (μ, λ)-selection strategies are distinguished:

- $(\mu{+}\lambda)$**-selection**: The best μ solutions in the union $\mathrm{X}_t \cup \mathrm{O}_t$ are selected. Because the size of this union equals $\mu + \lambda$, this selection is termed $(\mu + \lambda)$-selection. For instance, a $(\mu + 2)$-selection denotes a selection of the μ best solutions from the union of a set of μ parents and a set of 2 offsprings.
- (μ, λ)**-selection**: The best μ solutions among the λ offspring solutions, O_t, are selected, meaning that the previous parent population X_t is not considered in the selection. In this case $\mu \leq \lambda$ must hold.

Offspring individuals are generated by means of *recombination* and *mutation* operators. Recombination is used to combine information of two or more parents in order to form an offspring, for instance by averaging or by crossover. Theoretically, it has been shown that for decomposable functions recombination can enhance the frequency of parts of the individuals that contribute to a good fitness [36] and decrease deteriorating effects of mutation [14]. Recombination also serves to create diversity and to exchange information across a population. In the context of EMO, it can create solutions that lie in the gaps between existing solutions.

Mutation modifies an individual by a small random perturbation. For instance, a Gaussian mutation of a given individual $x \in \mathbb{R}^n$ corresponds to:

$$x' = x + \sigma z \text{ with } z \sim \text{Normal}(0, \Sigma),$$

where $\text{Normal}(0, \Sigma)$ denotes a multivariate Gaussian distribution with mean 0 and covariance matrix Σ, and σ is a scaling factor. In ES it is common to adapt the scaling factor, σ_t, across the different generations. For this, several control schemes were proposed. One of these proposals is based on success-rates (e.g. [14, 44, 65]). In ES following a $1/\tau$th-success rule means that σ is multiplied by a constant $0 < \xi < 1$, if the success-rate after a fixed epoch of $c > \tau$ iterations is below $1/\tau$, and divided by ξ otherwise. A value of $\tau = 5$ proved to be a good default setting for a range of problems and is used as a standard setting – hence the name $1/5$th success rule [65]. Alternatively, mutative self-adaptation and derandomized self-adaptation are widely used. Here σ (or even Σ itself) gets part of the individual and undergoes an evolutionary process. For details, see [14] and [38], respectively.

The CMA-ES [38] also adapts the covariance matrix Σ during the generations, adjusting the mutation to the local quadratic form of the objective function. For this, it integrates the sample path from the history of the individuals (evolution path). In particular, for continuous objective functions with an high condition number and correlation between the optimization variables, the CMA-ES is often preferred to other ES instantiations.

The behaviour of almost all evolutionary algorithms can be modelled as an homogeneous Markovian process with kernel $K(\mathrm{X}, A)$, describing the transition probability of a given population $\mathrm{X} \in (\mathbb{R}^n)^\mu$ entering in some set $A \subset (\mathbb{R}^n)^\mu$ in the subsequent generation. For instantiations of transition kernels for EA see [61].

Based on this, conditions for complete convergence of EA can be established. Complete convergence is a strong type of probabilistic convergence. Lucacs [53] defines it as:

Definition 5.1 (Complete convergence) *Let Y denote a random variable (which might as well be a constant) and $\{Y_t\}_{t\in\mathbb{N}}$ a sequence of random variables defined on a probability space $(\mathcal{B}, \mathcal{A}, P)$. Then $\{Y_t\}_{t\in\mathbb{N}}$ is said to converge completely to Y, if for any $\epsilon > 0$:*

$$\lim_{t\to\infty} \sum_{i=1}^{t} P\{|Y_i - Y| > \epsilon\} < \infty.$$

Next, let us define $b : \wp((\mathbb{R}^n)^\mu) \to \mathbb{R}$ as the function that assigns to a population in $(\mathbb{R}^n)^\mu$ the objective function value of its best individual. Moreover, for $\epsilon > 0$ and f_* the global minimum of the objective function f (provided it exists), we define $A_\epsilon = \{\mathrm{X} \in (\mathbb{R}^n)^\mu \,|\, b(\mathrm{X}) \leq f_* + \epsilon\}$. Using these definitions, we can state the following result.

Theorem 5.1 (Corollary 6.3 in [61]) *An EA whose Markovian kernel satisfies the condition $K(\mathrm{X}, A_\epsilon) \geq \theta > 0$, for all $\mathrm{X} \in A_\epsilon^c = (\mathbb{R}^n)^\mu \setminus A_\epsilon$, and $K(\mathrm{X}, A_\epsilon) = 1$ for any $\mathrm{X} \in A_\epsilon$, for any $\epsilon > 0$, will converge completely to the global minimum of the real valued function f, regardless of the initial distribution of the individuals.*

It is interesting to note that the condition on the Markovian kernel is satisfied for ES using a $(\mu+\lambda)$-selection strategy, considering a positive definite covariance matrix Σ and bounded scaling factors $0 < \sigma_{min} < \sigma < \sigma_{max} < \infty$, provided they are applied to bound constrained problems with a continuous objective function (see [61]).

On simple quadratic functions, Beyer [13] derived expressions showing a linear convergence rate of ES with optimally adapted step sizes, using a $(\mu + \lambda)$ or (μ, λ)-selection strategies, for a dimension $n \to \infty$. For a summary, see also [14]. Empirically, it was shown that most of these results already hold for n of moderate size. Jägersküpper [44] proved that, on quadratic forms, ES using a $(1 + 1)$-selection combined with a $1/5$th success rule for step-size adaptation converge linearly with overwhelming probability, that is, the probability that the distance to the optimum does not halve in $\mathcal{O}(n)$ iterations approaches zero with exponential rate in n.

More than analysing the convergence properties of classical EA, proposals were made to change the design of some methods belonging to this class, in order to enforce convergence to a stationarity point. Hart [40] introduced a variant of ES, called Evolutionary Pattern Search, which generalizes the Stochastic Pattern Search method proposed in [39]. The new algorithm includes in the mutation operator a set of search directions which forms a positive basis for $\mathbb{R}^n$. It uses update schemes similar to the ones of DSM, but opposed to them considers a population of search points and randomizes some of the procedures.

Recently, Diouane et al. [28] have also proposed a slightly modified version of CMA-ES, and other similar ES instantiations. For objective functions which are locally Lipschitz continuous, they have proved the convergency for a Clarke stationarity point, independently of the points considered in initialization. The proposed modification consists in strategies to update the step size parameter based on different sufficient decrease conditions.

Besides these theoretical results, we note that in the field of EA, systematic experimental research plays an important role in the algorithmic development. See [10], for a treatise.

5.2 An Overview on Some (Classical) Evolutionary Multiobjective Optimization (EMO) Algorithms

Shifting the attention from single to multiobjective EA means, in the first place, to focus in the selection operator design. While in single objective optimization selection procedures can be directly based on the objective function value and in rankings depending on it, in MOO, where there are partially ordered objective function vectors, fitness assignment or ranking is not that straightforward. Three main alternatives can be identified in the EMO literature, which were proposed in the given order:

1. **Scalarization-based approaches:** The objective function vectors are aggregated into a single scalar value, using simple weighting or an utility function. In order to compute different solutions, the parameters of the utility function are dynamically or randomly changed. Examples are the Vector Evaluated Genetic Algorithm (VEGA) [64] and the Random Objective Selection Evolutionary Algorithm (ROSEA) [62], the latter based on an earlier framework proposed by [49] for multiobjective ES.

2. **Nondominance and diversity approaches:** Solutions are computed based on dominance and diversity. Often, nondominance is prioritised to diversity to achieve convergence to a set of Pareto optimal solutions. Examples of this approach are the Nondominated Sorting Genetic Algorithm (version 2) NSGA-II [26] and the Strength Pareto Evolutionary Algorithm (version 2) SPEA2 [76].

3. **Indicator-based approaches:** In this case, a selection is performed among sets, favouring the ones which present a better performance indicator. This performance indicator should measure the quality of a set as an approximation to the Pareto front of the problem. Indicator based evolutionary algorithms were suggested as an algorithm class in [75], being the hypervolume indicator often chosen as performance indicator. The Hypervolume Estimation Algorithm HypE [8] and the S-Metric Selection Evolutionary Multiobjective Optimization Algorithm SMS-EMOA [30] are instantiations of indicator based evolutionary algorithms. The same happens for a common variant of the Multiobjective Covariance Matrix Adaptation Evolution Strategy MO-CMA-ES [43, 72].

Aggregation based methods were described in the first pioneering papers on EMO. In VEGA [64], weights of the objective functions are randomly changed, promoting diversity inside the population and its better coverage of the Pareto front. With the same purpose, ROSEA [62] changes randomly the objective function used in the selection. Rudolph [62] analysed the convergence properties of ROSEA for a quadratic MOO problem, establishing a sublinear convergence rate. The major drawback of these aggregation based approaches relies on their inability to capture concave parts of the Pareto fronts. Thus, they have been widely abandoned in the EMO field.

The selection mechanisms incorporated in the NSGA, its successor NSGA-II, and SPEA2 are prominent examples of the second selection principle: a combination of nondominance and diversity. Basically, these algorithms differ in the ranking, based on nondominance level, and the subsequent diversity measurement. A more detailed discussion follows in Subsection 5.3. Other algorithms that fall into this class are the Pareto Archived Evolution Strategy (PAES) [47] and the ϵ-MOEA [25], which partition the objective function space into grid cells to measure diversity. A stronger focus on variation/sampling operators design gave rise to algorithms such as the Multiobjective Estimation of Distribution Algorithm [15], the Differential Evolutionary Multiobjective Optimization (DEMO) [60] and the Multiobjective Covariance Matrix Adaptation Evolution Strategy (MO-CMA-ES) [43, 72].

The SMS-EMOA [30] and HypE [8] can be seen as instantiations of indicator-based methods, the design of which is directly governed by a performance indicator. Because this design principle is a recent trend in the advancement of EMO algorithms, we will devote Subsection 5.4 to it. The hypervolume indicator was also proposed as a selection criterion in MO-CMA-ES [43, 72]. However, the main contribution present in this method is to generalize the single objective CMA-ES to MOO. We will discuss it in more detail in Subsection 5.5.

Another line of algorithmic designs has focused on interacting models of evolution and co-evolution, inspired by the predator-prey approaches. Here, different objectives are represented by different prey-individuals, that simultaneously perform the selection on different aggregated objective functions (see [37, 50]). Though these bio-inspired algorithms provide flexibility and robustness in practical settings [37], their convergence properties remain widely unexplored.

5.3 Combining Dominance and Diversity: NSGA-II and SPEA2

NSGA-II [26] is probably the most commonly used EMO algorithm, partially due to its straightforward, yet effective, design. NSGA-II is a GA considering a $(\mu + \mu)$-selection strategy, and binary tournament for mating. Its main innovation, as opposed to single objective optimization algorithms, consists in ranking solutions based on a vector valued fitness function, and thereby at the same time considers domination and diversity. Ranking is done in two steps:

1. **Nondominated sorting:** In this step the union of the parent and offspring populations $Q \in \Omega^{\mu+\mu}$ is partitioned into $Q^{(1)}$, ..., $Q^{(k)}$, where $Q^{(1)}$ is the set of nondominated solutions in Q, and $Q^{(i+1)}$ is recursively defined as the nondominated set of $Q - \cup_{j=1}^{i} Q^{(j)}$, for $i = 1, 2, \ldots$. Nondominated sorting will result in $0 < \kappa \leq |Q|$ nonempty sets $Q^{(1)}, \ldots, Q^{(i)}, \ldots ,Q^{(\kappa)}$. The index i of the set $Q^{(i)}$ determines the *nondominance level* of the solutions in $Q^{(i)}$.

2. **Crowding distance sorting:** Each set $Q^{(i)}$, $i \in \{1, \ldots, \kappa\}$, obtained by nondominated sorting, can be further sorted by means of the crowding distance. The crowding distance measures the contribution of a solution to the diversity of a population – the higher the crowding distance the better the solution. For some $q \in Q^{(i)}$ it is given by $crowd(q) = \sum_{d=1}^{n} (\Delta_d^{+}(q, Q^{(i)}) + \Delta_d^{-}(q, Q^{(i)}))$, where

$\Delta_d^+(q, Q^{(i)})$ denotes the next higher d-th coordinate of a point $q' \in Q^{(i)} - \{q\}$ and $\Delta_d^-(q, Q^{(i)})$ denotes the next lower d-th coordinate of a point $q'' \in Q^{(i)} - \{q\}$.

A crucial design principle of NSGA-II is to prioritize on the nondominance level. Only in case of incomparable solutions it favours those that better contribute to diversity. In common implementations of NSGA-II either classical genetic operators are used or – in the canonical version for continuous search spaces - simulated binary crossover in combination with polynomial mutation is considered. For details on these operators we refer to the literature [26].

Although NSGA-II is an elitist algorithm, it was shown that it can deteriorate, meaning that after some generations the set of approximate solutions might be strictly worse then the previous population [31]. Nevertheless, for population with larger sizes the effects of deterioration tend to disappear [31]. In fact, in a wide range of benchmarks and application problems, NSGA-II was reported to yield good approximations of Pareto fronts, in particular for the 2-D case. A similar algorithm to NSGA-II that also enjoys wide popularity is SPEA2 [76]. Instead of using nondominated sorting for ranking in the first step, it computes the *strength* of individuals, by counting how many other individuals it dominates and by how many individuals it is dominated. A clustering method is then used as a selection criterion among individuals of equal strength, in order to promote diversity.

5.4 Indicator-based Approaches: SMS-EMOA and HypE

In the algorithms belonging to this class, selection strategies use performance indicators to measure the quality of different sets, as approximations to the Pareto front of a given problem. To do it, without the knowledge of the actual Pareto front, the hypervolume indicator was suggested, due to its favorable properties [77]. Given a reference point $r \in \mathbb{R}^m$, that is dominated by all the approximations considered for the Pareto front (or at least by the actual Pareto front of the problem), the hypervolume indicator measures the size of the space that is dominated by the set approximating the Pareto front and upper bounded by the considered reference point.

Definition 5.2 (Hypervolume indicator) *The hypervolume indicator (or* S-metric*, from '*Size of space covered*') for some (approximation) set $A \subset \mathbb{R}^m$ and a reference point $r \in \mathbb{R}^m$ that is dominated by all the points in A is defined as:*

$$\mathit{HI}(A) = \mathit{Vol}\{b \in \mathbb{R}^m | b \leq r \wedge \exists a \in A : a \leq b\} = \mathit{Vol}(\cup_{a \in A}[a, r])$$

Here $\mathit{Vol}(.)$ denotes the Lebesgue measure of a m-dimensional set of points, and $[a, r]$ denotes the interval box with lower corner a and upper corner r.

In 2-D this is simply the covered area, and in 3-D the covered volume (see Figure 1 for examples). For reviewing some important properties related to this indicator, lets us introduce a comparison operator between sets.

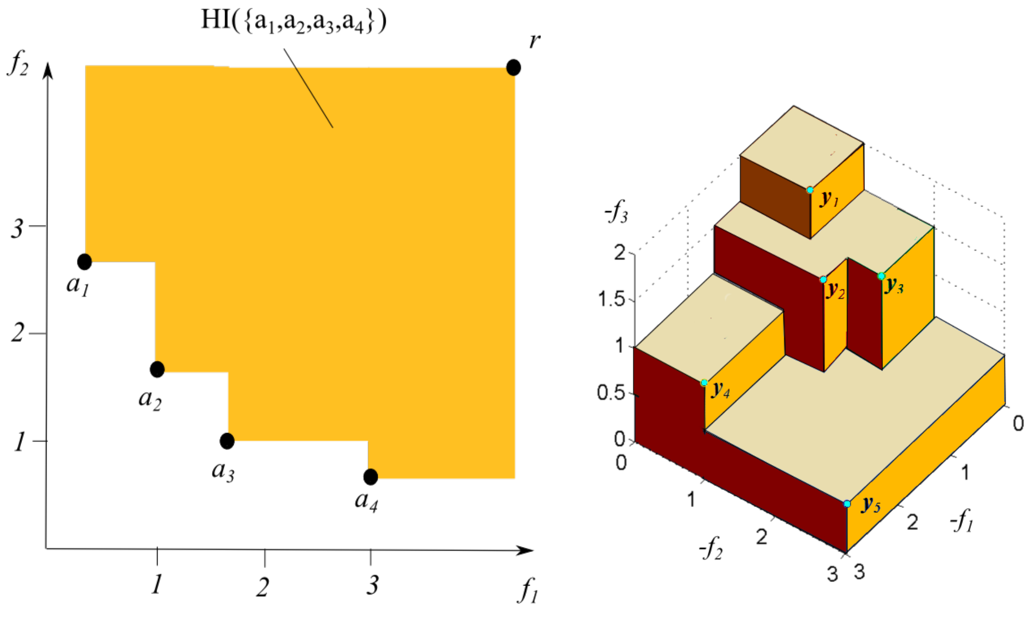

Figure 1: Hypervolume indicator in two dimensions for a set $A = \{a_1, ..., a_4\} \subset \mathbb{R}^2$ (left) and in three dimensions for a set $Y = \{y_1, \ldots, y_5\} \subset \mathbb{R}^3$ (right).

Definition 5.3 *Given two nondominated sets A and B, A is better than B, which is represented by $A \prec B$, if*

$$\forall b \in B : \exists a \in A : a \leq b \, \textit{(componentwise) and} \, \exists b \in B : \exists a \in A : a \prec b.$$

The hypervolume indicator is, up to an isomorphism, the only known indicator that presents the following monotonicity property.

Lemma 5.1 (Zitzler et al. [77]) *Let A and B be two nondominated sets with the properties $A \prec B$ and $\forall x \in A \cup B : x \prec r$, where r is the reference point used in the hypervolume computations. Then HI$(A) >$ HI(B).*

This property makes sure that strict improvements of approximating sets will result in an increment of the hypervolume. Moreover, the following property can be established for the hypervolume indicator.

Lemma 5.2 (Zitzler et al. [77]) *Let $\prec$ be defined as in Definition 5.3, and A and B be two nondominated sets with the property $\forall x \in A \cup B : x \prec r$, where r is the reference point used in the hypervolume computations. If HI$(A) >$ HI(B) then $B \not\prec A$.*

Basically, the lemma states that measuring by means of the hypervolume indicator will never favour an approximation set that is strictly worse than the another one.

This implies that search strategies that generate a sequence of approximation sets or populations with monotonously increasing hypervolume will not deteriorate [31].

Indicator-based EMO algorithms use indicators that measure the quality of the eventually obtained set to directly guide the selection [75]. The idea is instantiated in the S-Metric Selection Evolutionary Multiobjective Optimization Algorithm (SMS-EMOA) [30], which considers a $(\mu + 1)$-selection scheme. Like in NSGA-II, the ranking procedure could be subdivided into two steps.

- **Nondominated sorting:** The first step is nondominated sorting (like discussed for NSGA-II), giving rise to partitions of equal nondominance level $Q^{(1)}, \dots, Q^{(\kappa)}$.
- **Hypervolume-based ranking:** The partition with worst nondominance level, namely $Q^{(\kappa)}$, is reduced to a subset of $|Q^{(\kappa)}| - 1$ solutions. The subset that covers the biggest hypervolume survives.

The design of the algorithm makes sure that, regardless of the stochastic realization, the hypervolume of the approximating set will either grow or remain the same. From Lemma 5.2 it can be concluded that the search cannot deteriorate to strictly worse Pareto front approximations, as this might be the case in NSGA-II and SPEA2.

The choice of a $(\mu+1)$-selection scheme is motivated by the efficiency of the subset selection for this special case. It suffices to delete the solution in $Q^{(\kappa)}$ that presents the smallest contribution to the corresponding hypervolume. The contribution of a point $a \in A \subset \mathbb{R}^m$ to the hypervolume defined by the set A is measured by:

$$\Delta\text{HI}(a, A) = \text{HI}(A) - \text{HI}(A - \{a\}) \tag{1}$$

The computation of all contributions can be accomplished in asymptotically optimal time $\mathcal{O}(|A| \log |A|)$, for $m = 2$ and $m = 3$ [32]. For higher dimensions, up to now, only algorithms with polynomial, but superquadratical, running time are known [11]. This limits the applicability of SMS-EMOA to problems with only a moderate number of objective functions.

Recently, several results have been obtain on the distribution of points that maximizes the hypervolume indicator. A set is said to be μ-optimal relatively to the hypervolume indicator if it presents the biggest hypervolume among all sets of size μ. It has been shown that, if the size of the actual Pareto front is equal or greater than μ, μ-optimal sets consist of only Pareto optimal points. Moreover, in 2-D, when $\mu \to \infty$, the set of points maximizing the hypervolume indicator will form a dense subset on the Pareto front [6].

Recent studies have shown that in some cases the SMS-EMOA is not guaranteed to converge to a μ-optimal set (e.g. [5]). This results from its $(\mu + 1)$-selection scheme. To guarantee convergence to a μ-optimal set, a $(\mu + \lambda)$-selection with $\mu \geq \lambda$ that selects the subset of size μ with maximal hypervolume is both sufficient and necessary. For $m = 2$, an algorithm for subset selection with running time $\mathcal{O}(\mu^3)$ is available [5], but for $m > 2$ only algorithms with exponential running time are known.

In contrast to SMS-EMOA, the Hypervolume Estimation Algorithm (HypE) [8] considers a general $(\mu + \lambda)$-selection scheme, where λ can be bigger than 1. To accomplish practical feasible running times, instead of using exact computation in the subset selection procedures, it applies Monte Carlo sampling for computations related to the hypervolume indicator and a modified fitness assignment. Instead of only considering contributions of single points to the hypervolume indicator, it also considers joint contributions of two or more points to this indicator. The volume of the subspace that is jointly dominated by k points is assigned in equal proportions to the points that dominate it.

As the computational complexity of the hypervolume indicator grows exponentially with the number of objective functions [17], and Monte Carlo approximation is less affected by dimensionality, HypE is considered to be a promising algorithm for MOO with an high number of components for the objective function [8].

First convergence results in hypervolume-based optimization were recently obtained. Beume et al. [12] introduced an analysis method that allows to generalize convergence results for single objective EA to EMO algorithms. Using this method, it has been proved the convergence of SMS-EMOA with a $(1 + 1)$-selection, and of a modified version of SMS-EMOA with a $(\mu + 1)$-selection, that works with multiple reference points and considers tournament selection based on pairwise comparisons. In particular, Beume et al. [12] extended the convergence results of Rudolph [61] and Jägersküpper [44], discussed in Subsection 5.1, to these two algorithms.

Whereas earlier theoretical research focused on either very simple discrete problems [51] or in algorithmic designs that do not address diversity maintenance [63], the findings of Beume et al. [12] can be regarded as first results towards convergence analysis for common EMO algorithms in continuous problem classes of practical relevance. For now, results are still confined to particular, uncommon algorithmic instantiations, and more general convergence results would be desirable.

5.5 Multiobjective CMA-ES

While the previously discussed NSGA-II, SPEA2, SMS-EMOA, and HypE can also be used in discrete search spaces, the MO-CMA-ES [43] is an EMO method that is especially designed for continuous optimization, since it is intended to generalize the single objective CMA-ES to MOO. As in SMS-EMOA, it uses a ranking strategy which is firstly based on nondominated sorting and considers contributions of the hypervolume indicator (see Definition 5.2) as a secondary ranking criterion. It establishes a complete ordering on the population by prioritising nondominance level to hypervolume contribution. Consider a set $Q^{(i)}$, for some $i \in \{1, \dots, \kappa\}$, with more than two solutions of equal dominance level. The solution q with the minimal hypervolume contribution is ranked worst, then the solution with minimal hypervolume contribution is detected from the set $Q^{(i)} - \{q\}$ and assigned the second worst rank, and so forth.

The CMA-ES and MO-CMA-ES work with a Gaussian mutation operator, for which the corresponding covariance matrix Σ is adapted by means of an elaborated scheme that depends on a evolution path and a success probability. A success is declared if an offspring is selected as one of the parents of the next generation. In case of success the scaling factor σ is decreased, while otherwise it is increased. The path of past evolution steps that led to an offspring is integrated, in order to approximate a conjugate direction that is used to scale and rotate the shape of the distribution represented by the covariance matrix Σ. This way, the mutation distribution is adapted, allowing to increase the progress rate for near-quadratic forms with high condition numbers.

Updates of the covariance matrix require, at least, quadratic time. Despite this large computational effort, the MO-CMA-ES is an interesting alternative to algorithms with a fixed covariance matrix, in cases where the scale of the variables largely differs and search proceeds in directions diagonal to the main coordinate axes.

A state-of-the-art instantiation with a detailed description of the update procedures of the MO-CMA-ES is found in [72]. So far, a theoretical convergence theory is not available for MO-CMA-ES, though there is empirical evidence of its good performance [43]. The improvement of the MO-CMA-ES is still a topic of active research. Interesting developments include the introduction of advanced recombination schemes [71], step-size adaptation schemes [72] and replacement of random mating selection by a selection scheme motivated by reinforcement learning [52].

6 Final Remarks

This paper presented two major classes of algorithms suited for Derivative-free Multiobjective Optimization, namely multiobjective DSM and EMO algorithms.

The attentive reader will have noticed some commonalities in the design of DSM and EMO algorithms. Searching in the neighborhood of existing solutions in order to find improvements is a fundamental design principle in both classes, and also the use of adaptive step sizes to zoom into local optima is featured in all DSM and in some EMO methods, most notably in MO-CMA-ES. In addition, besides Pareto nondominance, diversity maintenance strategies are used in both fields as a secondary guidance for search. Compare, for instance, the similarity in the design of DMS (using a componentwise gap to select individuals) and NSGA-II (using the crowding distance). Moreover, to a certain extent, both algorithms combine the information of existing solutions on the Pareto front in order to find new points in the gaps. In BIMADS and in MULTIMADS for computing reference points and in EMO to generate offspring via recombination.

However, there are also remarkable differences between EMO algorithms and DSM for MOO. The algorithms belonging to the class of DSM are deterministic, presenting a well established convergence analysis. EMO methods are randomized and, for the common EMO algorithmic instantiations, convergence proves have not yet been established. From the first results, obtained by generalizing the convergence analysis from single objective EA, it can be concluded that the convergence of EMO methods will be of probabilistic nature, also addressing global optimums.

EMO algorithms introduced some interesting concepts, such as indicator-based selection and randomization. Some of these might also prove to be beneficial for DSM design in MOO. Finally, hybrids of multiobjective DSM and EMO algorithms, as they already exists in single objective optimization, might yield search algorithms that enjoy at the same time the flexibility and robustness of EMO algorithms and the deterministic convergence properties of DSM.

Acknowledgments

The authors would like to thank Professor Carlos Fonseca for interesting discussions on Multiobjective Optimization.

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] C. Audet, G. Savard, and W. Zghal. A mesh adaptive direct search algorithm for multiobjective optimization. *European J. Oper. Res.*, 204:545–556, 2010.

[5] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler. Investigating and exploiting the bias of the weighted hypervolume to articulate user preferences. In *GECCO '09: Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*, pages 563–570. ACM, 2009.

[6] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler. Theory of the hypervolume indicator: optimal μ-distributions and the choice of the reference point. In *FOGA '09: Proceedings of the 10th ACM SIGEVO Workshop on Foundations of Genetic Algorithms*, pages 87–102. ACM, 2009.

[7] T. Bäck and H. P. Schwefel. An overview of evolutionary algorithms for parameter optimization. *Evolut. Comput.*, 1:1–23, 1993.

[8] J. Bader and E. Zitzler. HypE: An algorithm for fast hypervolume-based many-objective optimization. *Evolut. Comput.*, 19:45–76, 2011.

[9] S. Bandyopadhyay, S. K. Pal, and B. Aruna. Multiobjective GAs, quantative indices, and pattern classification. *IEEE Trans. on Syst., Man, and Cybern. – Part B: Cybernetics*, 34:2088–2099, 2004.

[10] T. Bartz-Beielstein. *Experimental Research in Evolutionary Computation: The New Experimentalism*. Springer, 2006.

[11] N. Beume. S-Metric calculation by considering dominated hypervolume as Klee's measure problem. *Evolut. Comput.*, 17:477–492, 2009.

[12] N. Beume, M. Laumanns, and G. Rudolph. Convergence rates of SMS-EMOA on continuous bi-objective problem classes. In *FOGA '11: Proceedings of the 11th ACM SIGEVO Workshop on Foundations of Genetic Algorithms*, pages 243–251. ACM, 2011.

[13] H. G. Beyer. *The Theory of Evolution Strategies*. Springer, 2001.

[14] H. G. Beyer and H. P. Schwefel. Evolution strategies – A compreehensive introduction. *Natural Computing*, 1:3–52, 2002.

[15] P. A. N. Bosman and D. Thierens. The naive MIDEA: a baseline multi-objective EA. In *EMO 2005: Third International Conference in Evolutionary Multi-Criterion Optimization*, LNCS, pages 428–442, 2005.

[16] J. Branke, K. Deb, K. Miettinen, and R. Slowinski (Eds.). *Multiobjective Optimization: Interactive and Evolutionary Approaches*, volume 5252 of *Lecture Notes in Computer Science*. 2008.

[17] K. Bringmann and T. Friedrich. Approximating the least hypervolume contributor: NP-Hard in general, but fast in practice. In *EMO 2009: Fifth International Conference in Evolutionary Multi-Criterion Optimization*, LNCS, pages 6–20, 2009.

[18] F. H. Clarke. *Optimization and Nonsmooth Analysis*. John Wiley & Sons, 1983. Reissued by SIAM, 1990.

[19] C. A. C. Coello. Evolutionary multi-objective optimization: a historical view of the field. *Computational Intelligence Magazine, IEEE*, 1:28–36, 2006.

[20] A. R. Conn, K. Scheinberg, and Ph. L. Toint. On the convergence of derivative-free methods for unconstrained optimization. In M. D. Buhmann and A. Iserles, editors, *Approximation Theory and Optimization: Tributes to M. J. D. Powell*, pages 83–108. Cambridge University Press, 1997.

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

[22] 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.

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

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

[25] K. Deb, M. Mohan, and S. Mishra. Evaluating the ϵ-domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions. *Evolut. Comput.*, 13:501–525, 2005.

[26] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. on Evolut. Comput.*, 6:182–197, 2002.

[27] S. Le Digabel. Algorithm 909: NOMAD: Nonlinear optimization with the MADS algorithm. *ACM Trans. Math. Software*, 37:44:1–44:15, 2011.

[28] Y. Diouane, S. Gratton, and L. N. Vicente. Globally convergent evolution strategies and CMA-ES. Technical report, Preprint 12-03, Dept. Mathematics, Univ. Coimbra, 2012.

[29] T. Dobzhansky. *Genetics of the Evolutionary Process*. Columbia Univ. Pr., 1970.

[30] M. Emmerich, N. Beume, and B. Naujoks. An EMO algorithm using the hypervolume measure as selection criterion. In *EMO 2005: Third International Conference in Evolutionary Multi-Criterion Optimization*, LNCS, pages 62–76, 2005.

[31] M. Emmerich, A. H. Deutz, R. Li, and J. W. Kruisselbrink. Getting lost or getting trapped: on the effect of moves to incomparable points in multiobjective hillclimbing. In *GECCO '10: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, pages 1963–1966. ACM, 2010.

[32] M. Emmerich and C. M. Fonseca. Computing hypervolume contributions in low dimensions: asymptotically optimal algorithm and complexity results. In *EMO 2011: Sixth International Conference in Evolutionary Multi-Criterion Optimization*, LNCS, pages 121–135, 2011.

[33] E. Fermi and N. Metropolis. Los Alamos unclassified report LS–1492. Technical report, Los Alamos National Laboratory, USA, 1952.

[34] G. B. Fogel, D. B. Fogel, and L. J. Fogel. Evolutionary programming. *Scholarpedia*, 6:1818, 2011.

[35] C. M. Fonseca and P. J. Fleming. Genetic algorithms for multiobjective optimization: formulation, discussion and generalization. In *Genetic Algorithms: Proceedings of the Fifth International Conference*, pages 416–423. CA: Morgan Kaufmann, 1993.

[36] D. E. Goldberg and J. H. Holland. Genetic algorithms and machine learning. *Machine Learning*, 3:95–99, 1988.

[37] C. Grimme, J. Lepping, and A. Papaspyrou. Exploring the behavior of building blocks for multi-objective variation operator design using predator-prey dynamics. In *GECCO '07: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*, pages 805–812. ACM, 2007.

[38] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolut. Comput.*, 9:159–195, 2001.

[39] W. E. Hart. A convergence analysis of unconstrained and bound constrained evolutionary pattern search. *Evolut. Comput.*, 9:1–23, 2001.

[40] W. E. Hart. Evolutionary pattern search algorithms for unconstrained and linearly constrained optimization. In *Evolutionary Programming VII*, pages 303–312. Springer-Berlin, 2001.

[41] R. Hooke and T. A. Jeeves. “Direct search” solution of numerical and statistical problems. *Journal of the ACM*, 8:212–229, 1961.

[42] P. D. Hough, T. G. Kolda, and V. J. Torczon. Asynchronous parallel pattern search for nonlinear optimization. *SIAM J. Sci. Comput.*, 23:134–156, 2001.

[43] C. Igel, N. Hansen, and S. Roth. Covariance matrix adaptation for multi-objective optimization. *Evolut. Comput.*, 15:1–28, 2007.

[44] J. Jägersküpper. How the (1+1) ES using isotropic mutations minimizes positive definite quadratic forms. *Theoretical Computer Science*, 361:38–56, 2006.

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

[46] C. T. Kelley. *Implicit Filtering*. SIAM, 2011.

[47] J. D. Knowles and D. W. Corne. Approximating the nondominated front using the Pareto archived evolution strategy. *Evolut. Comput.*, 8:149–172, 2000.

[48] 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.

[49] F. Kursawe. A variant of evolution strategies for vector optimization. In *Proceedings of the 1st Workshop on Parallel Problem Solving from Nature*, PPSN I, pages 193–197. Springer-Verlag, 1991.

[50] M. Laumanns, G. Rudolph, and H. P. Schwefel. A spatial predator-prey approach to multi-objective optimization: a preliminary study. In *Proceedings of the 5th International Conference on Parallel Problem Solving from Nature*, PPSN V, pages 241–249. Springer-Verlag, 1998.

[51] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler. Combining convergence and diversity in evolutionary multiobjective optimization. *Evolut. Comput.*, 10:263–282, 2002.

[52] I. Loshchilov, M. Schoenauer, and M. Sebag. Not all parents are equal for MO-CMA-ES. In *EMO 2011: Sixth International Conference in Evolutionary Multi-Criterion Optimization*, LNCS, pages 31–45, 2011.

[53] E. Lukacs. *Stochastic Convergence*. Academic Press, 1975.

[54] J. F. A. Madeira, H. L. Pina, E. B. Pires, and J. M. Monteiro. Surgical correction of scoliosis: Numerical analysis and optimization of the procedure. *Int. J. Numer. Meth. Biomed. Engng.*, 40:227–240, 2010.

[55] J. F. A. Madeira, H. L. Pina, and H. C. Rodrigues. GA topology optimization using random keys for tree encoding of structures. *Struct. Multidisc. Opt.*, 7:308–313, 2010.

[56] K. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, 1999.

[57] J. A. Nelder and R. Mead. A simplex method for function minimization. *Comp. J.*, 7:308–313, 1965.

[58] M. J. D. Powell. Direct search algorithms for optimization calculations. *Acta Numerica*, 7:287–336, 1998.

[59] E. Reehuis, J. Kruisselbrink, A. Deutz, T. Bäck, and M. Emmerich. Multiobjective optimization of water distribution networks using SMS-EMOA. In *Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2011)*, pages 269–279. International Center for Numerical Methods in Engineering, 2011.

[60] T. Robic and B. Filipic. DEMO: Differential evolution for multi-objective optimization. In *EMO 2005: Third International Conference in Evolutionary Multi-Criterion Optimization*, LNCS, pages 520–533, 2005.

[61] G. Rudolph. *Convergence Properties of Evolutionary Algorithms*. Verlag Dr. Kovac, 1997.

[62] G. Rudolph. On a multi-objective evolutionary algorithm and its convergence to the Pareto set. In *Proceedings of the 5th IEEE Conference on Evolutionary Computation*, pages 511–516. IEEE Press, 1998.

[63] G. Rudolph and A. Agapie. Convergence properties of some multi-objective evolutionary algorithms. In *Congress on Evolutionary Computation (CEC 2000)*, pages 1010–1016. IEEE Press, 2000.

[64] J. D. Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of the First International Conference on Genetic Algorithms and their Applications*, pages 93–100. Lawrence Erlbaum Associates, 1985.

[65] H. P. Schwefel. *Evolution and Optimum Seeking*. Wiley, 1995.

[66] V. Torczon. *Multi-Directional Search: A Direct Search Algorithm for Parallel Machines*. PhD thesis, Department of Mathematical Sciences, Rice University, USA, 1989.

[67] V. Torczon. On the convergence of pattern search algorithms. *SIAM J. Optim.*, 7:1–25, 1997.

[68] A. I. F. Vaz and L. N. Vicente. A particle swarm pattern search method for bound constrained global optimization. *J. Global Optim.*, 39:197–219, 2007.

[69] A. I. F. Vaz and L. N. Vicente. PSwarm: A hybrid solver for linearly constrained global derivative-free optimization. *Optim. Methods Softw.*, 24:669–685, 2009.

[70] L. N. Vicente and A. L. Custódio. Analysis of direct searches for discontinuous functions. *Math. Program.*, on line, 2012.

[71] T. Voß, N. Hansen, and C. Igel. Recombination for learning strategy parameters in the MO-CMA-ES. In *EMO 2009: Fifth International Conference in Evolutionary Multi-Criterion Optimization*, LNCS, pages 155–168, 2009.

[72] T. Voß, N. Hansen, and C. Igel. Improved step size adaptation for the MO-CMA-ES. In *GECCO '10: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, pages 487–494. ACM, 2010.

[73] W. Zghal, C. Audet, and G. Savard. A new multi-objective approach for the portfolio selection problem with skewness. In *Advances in Quantitative Analysis of Finance and Accounting*, pages 792–802. Airiti Press, 2011.

[74] X. Zhong, W. Fan, J. Lin, and Z. Zhao. A novel multi-objective compass search. In *IEEE International Conference in Progress in Informatics and Computing*, pages 24–29. IEEE Press, 2010.

[75] E. Zitzler and S. Künzli. Indicator-based selection in multiobjective search. In *Proc. 8th International Conference on Parallel Problem Solving from Nature*, PPSN VIII, pages 832–842, 2004.

[76] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization. In *Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001)*, pages 95–100. International Center for Numerical Methods in Engineering, 2002.

[77] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. Grunert da Fonseca. Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Trans. on Evolut. Comput.*, 7:117–132, 2003.