The Directed Search Method for Unconstrained Multi-Objective Optimization Problems

Oliver Schütze, Adriana Lara, and Carlos A. Coello Coello

CINVESTAV-IPN
Computer Science Department
Av. IPN 2508, C. P. 07360, Col. San Pedro Zacatenco
Mexico City, Mexico
{schuetze,ccoello}@cs.cinvestav.mx,
alara@computacion.cs.cinvestav.mx

Abstract. Here we propose a novel idea for iterative search procedures for the numerical treatment of multi-objective optimization problems, namely to steer the search along a predefined search direction given in objective space. Based on this idea we will present two methods: a descent method, i.e., an iterative procedure which seeks for improvements of the given model, and a novel continuation method which allows to search along the Pareto set of a given MOP. The advantage of the first procedure is that it has a physical meaning and the search can be steered according to the given situation, and the advantage of the latter procedure is that it does not require any 2nd gradient information and is hence also applicable to higher dimensional models without exploiting the sparsity of the model. We demonstrate the strength of the two methods on several examples.

Keywords: multi-objective optimization, descent method, continuation, goal programming.

1 Introduction

In a variety of engineering applications—as well as in other fields—one is faced with the problem that several objectives have to be optimized concurrently leading to a *multi-objective optimization problem* (MOP). Such MOPs have been considered for aerodynamic shape optimization, autonomous vehicle navigation, machine learning, and the design of blades, polymers, mechatronic systems, and space missions [6, 7, 11, 16, 26, 32, 37, 39], to name just a few applications. As a general example, two common goals in product design are certainly to maximize the quality of the product *and* to minimize its cost. Since these two goals are typically contradicting, it comes as no surprise that the solution set—the so-called *Pareto set*—of an MOP does in general not consist of one single solution but rather of an entire set of solutions (see Section 2 for a more detailed discussion).

So far, many numerical methods for the treatment of a given MOP have been proposed. There exist, for instance, many scalarization methods which transform the MOP into a 'classical' scalar optimization problem (SOP). By choosing a clever sequence of SOPs a suitable finite size approximation of the entire Pareto set can be obtained (see [9, 28, 15, 23, 14, 13] and references therein). Another approach to approximate the Pareto set is to use set oriented methods such as subdivision techniques ([11, 21]) or stochastic search methods (see [10, 31, 8, 35] and references therein). Since the Pareto set forms under some mild regularity conditions locally a $(k-1)$-manifold, where k is the number of objectives involved in the MOP, specialized continuation methods which perform a search along the Pareto set are very efficient if one (or more) solution is at hand ([1, 19]).

In certain cases one has to be content to improve a given state of a system only locally, and in other scenarios this is even desired. The former happens when the cost of a function evaluation is high, and the latter is for instance given in memetic strategies, i.e., evolutionary strategies which are hybridized with local search procedures. It has been observed that for these methods the proper balance of global and local search is a delicate problem, and that a too efficient local search procedure could have a negative influende on the entire candidate set ([20, 27]). For such problems, the integration of a descent direction in the numerical scheme can be of interest. A descent direction at a point x is a direction in which (ideally) all objective values improve. Descent directions for MOPs have been proposed in [15, 31, 3], however, one potential drawback of these approaches—at least in the current context—is that the actual direction of the search (apart from the fact that objectives acutally improve) is not given, which makes it hard for the numerical scheme to 'steer' the optimization process.

The method we present in this paper, the *Directed Search Method*, allows to steer the search process in every direction in objective space. To be more precise, given a point $x \in \mathbb{R}^n$ in parameter space, and a vector $d \in \mathbb{R}^k$ (in objective space) a direction vector $\nu \in \mathbb{R}^n$ can be computed such that

$$\lim_{t \searrow 0} \frac{f_i(x_0 + t\nu) - f_i(x_0)}{t} = d_i, \quad i = 1, \ldots, k, \tag{1}$$

where $f_i : \mathbb{R}^n \to \mathbb{R}$ denotes the i-th objective of the MOP, i.e., the relative change of each objective value for an infinitesimal step size is given by the values of d. We use this to propose further on two methods: first, we propose the Directed Search descent method, which can be viewed as a class of descent methods, and discuss possible choices of d, in particular greedy directions for the well-known Weighted Sum approach (e.g., [28]) and for Goal Programming ([5]). Second, we use the Directed Search Method to construct a novel multi-objective continuation method. The particular advantage of this method is that it does—in contrast to the methods in [1, 19]—not require any 2nd gradient information which makes it a competitive alternative at least for higher dimensional MOPs. The idea to adjust the search direction in image space a priori to obtain a numerical scheme has to our best knowledge never been discussed in literature. Since the underlying idea is not so far off, we conjecture that the lack of the

consideration of this approach has two possible reasons: (i) the numerics and (ii) the meaning of the approach. Ad (i): the computation of the search vector ν requires the solution of a possibly highly underdetermined system of linear equations, and the condition number of this system increases as the point x gets nearer to a local solution. However, we have observed that state-of-the-art numerical tools are able to handle such problems, even for higher dimensional problems (for instance, in Section 6 the Pareto set of an MOP with $n = 100,000$ parameters is presented). Ad (ii): it is ad hoc unclear how to 'steer' the optimization process, i.e., how to choose d. Here, we make a beginning and present some choices which are locally optimal in certain cases, however, some 'global' strategies are to be established in the future.
A very similar representation of the descent cone which is the basis for our descent method can be found in [3], however, the authors of [3] derived their result differently, and did not exploit the 'steering feature' in their work.

The remainder of this paper is organized as follows: in Section 2 we state the required background, and in Section 3 we present the central idea of the Directed Search method. Based on this we propose then a descent method in Section 4 and a continuation method in Section 5. In Section 6 we present some numerical results, and finally conclude in Section 7.

2 Notations and Background

In the following we consider unconstrained MOPs which are of the following form:

$$\min_{x\in\mathbb{R}^n}\{F(x)\}, \tag{MOP}$$

where F is defined as the vector of the objective functions

$$F : Q \to \mathbb{R}^k, \qquad F(x) = (f_1(x), \dots, f_k(x)),$$

and where each objective $f_i : Q \to \mathbb{R}$ is sufficiently smooth. The optimality of an MOP is defined by the concept of *dominance* ([30]).

Definition 1. *(a) Let $v, w \in \mathbb{R}^k$. Then the vector v is* less than *w ($v <_p w$), if $v_i < w_i$ for all $i \in \{1, \dots, k\}$. The relation $\le_p$ is defined analogously.*

(b) A vector $y \in \mathbb{R}^n$ is dominated *by a vector $x \in \mathbb{R}^n$ ($x \prec y$) with respect to (MOP) if*

$$F(x) \le_p F(y) \quad \text{and} \quad F(x) \neq F(y), \tag{2}$$

else y is called non-dominated by x.

(c) A point $x \in Q$ is called (Pareto) optimal *or a* Pareto point *if there is no $y \in Q$ which dominates x.*

The set of all Pareto optimal solutions is called the *Pareto set*, and is denoted by $\mathcal{P}$. The image $F(\mathcal{P})$ of the Pareto set is called the *Pareto front*. Both sets form typically—i.e., under certain mild regularity assumptions on the objectives—a

$(k-1)$-dimensional object ([19]).
The derivative of F at a point x is given by

$$DF(x) = \begin{pmatrix} \nabla f_1(x)^T \\ \vdots \\ \nabla f_k(x)^T \end{pmatrix} \in \mathbb{R}^{k \times n}, \tag{3}$$

where $\nabla f_i(x)$ denotes the gradient of objective f_i. In case all the objectives of the MOP are differentiable the following famous theorem of Kuhn and Tucker [25] states a necessary condition for Pareto optimality for unconstrained MOPs.

Theorem 1. *Let x^* be a Pareto point of (MOP), then there exists a vector $\alpha \in \mathbb{R}^k$ with $\alpha_i \geq 0, i = 1, \ldots, k$, and $\sum_{i=1}^k \alpha_i = 1$ such that*

$$\sum_{i=1}^k \alpha_i \nabla f_i(x^*) = 0. \tag{4}$$

The theorem claims that the vector of zeros can be written as a convex combination of the gradients of the objectives at every Pareto point. Obviously, (4) is not a sufficient condition for Pareto optimality. On the other hand, points satisfying (4) are certainly 'Pareto candidates'.

Definition 2. *A point $x \in \mathbb{R}^n$ is called a* Karush–Kuhn–Tucker point[1] *(KKT–point) if there exist scalars $\alpha_1, \ldots, \alpha_k \geq 0$ such that $\sum_{i=1}^k \alpha_i = 1$ and that Equation (4) is satisfied.*

Finally, given a manifold $\mathcal{M}$ and a regular point $x \in \mathcal{M}$ we denote the tangent space of $\mathcal{M}$ at x by $T_x\mathcal{M}$.

3 The Central Idea of the Directed Search Method

Here we describe the underlying idea of the Directed Search Method which we will use in the following to construct iterative search procedures.
Assume a point $x_0 \in \mathbb{R}^n$ is given and a vector $d \in \mathbb{R}^k$ representing a desired search direction in image space. To be more precise, a search direction $\nu \in \mathbb{R}^n$ in parameter space is sought such that for $y_0 := x_0 + t\nu$, where $t \in \mathbb{R}_+$ is the step size, it holds:

$$\lim_{t \searrow 0} \frac{f_i(y_0) - f_i(x_0)}{t} = \langle \nabla f_i(x_0), \nu \rangle = d_i, \quad i = 1, \ldots, k \tag{5}$$

Using the Jacobian of F, Equation (5) can be stated in matrix vector notation as

$$DF(x_0)\nu = d. \tag{6}$$

[1] Named after the works of Karush [22] and Kuhn & Tucker [25].

Hence, such a search direction ν can be computed by solving a system of linear equations. Since typically the number of parameters is (much) higher than the number of objectives in a given MOP, i.e., $n >> k$, system (6) is (probably highly) underdetermined which implies that its solution is not unique. To prevent this, the solution with the lowest norm can be chosen leading to

$$\nu = DF(x_0)^+ d, \tag{7}$$

where $A^+ \in \mathbb{R}^{n \times k}$ denotes the pseudo inverse of a matrix $A \in \mathbb{R}^{k \times n}$, $k \leq n$. In case the rank of A is maximal, the pseudo inverse is given by $A^+ = A^T(AA^T)^{-1}$.

In case the MOP contains m active inequality constraints $g_1, \ldots, g_m : \mathbb{R}^n \to \mathbb{R}$ at a point x—which is not within the scope of this work—one has to solve instead of (6) the enlarged system

$$\begin{aligned} DF(x)\nu &= d \\ DG(x)\nu &\leq 0, \end{aligned} \tag{8}$$

where

$$DG(x) = \begin{pmatrix} \nabla g_1(x)^T \\ \vdots \\ \nabla g_m(x)^T \end{pmatrix} \in \mathbb{R}^{m \times n}. \tag{9}$$

For the solution of such systems we refer e.g. to [4]. An analog statement for equality constraints, however, does in general not hold since they typically reduce the dimension of the search space, and hence, the feasible choice of d may be restricted. In that case, a solution of the (extended) equation system (8) in a least squares manner seems to be suitable.

In the following we discuss possible descent strategies and a continuation method for the numerical treatment of a given MOP where we utilize (6). Apparently, the proper choice of the search direction d will be one central issue for all methods.

4 The Directed Search Descent Method

In this section we present the descent methods which arise from (6) and discuss several possible choices of the direction vector.

4.1 The Method

In the following we will re-formulate the descent cone $D(x)$ at a point x which allows to find a (descent) direction $\nu = \nu(\alpha) \in \mathbb{R}^n$ for every search direction $\alpha \in \mathbb{R}^k$. Based on this we will further on propose a new iterative search procedure.

The Set of Descent Directions The following little discussion shows that the direct approach (6) can be used to find every descent direction at a given point $x_0 \in \mathbb{R}^n$. By the choice of the dominance relation in Definition 4.1 it follows that the descent cone at x_0 is given by

$$\begin{aligned} D(x_0) = \{\nu \in \mathbb{R}^n \backslash \{0\}\ :\ & \langle \nabla f_i(x_0), \nu \rangle \leq 0, \text{ for all } i = 1, \ldots, k, \\ & \text{and } \langle \nabla f_j(x_0), \nu \rangle < 0, \text{ for a } j \in \{1, \ldots, k\}\}, \end{aligned} \tag{10}$$

which does ad hoc not allow to compute any such element of $D(x_0)$. Using the Jacobian, Equation (10) can be written in matrix vector notation as

$$DF(x_0)\nu \leq_p 0 \quad \text{and} \quad DF(x_0)\nu \neq 0 \tag{11}$$

After possible normalization (11) can be stated as

$$DF(x)\nu = -\alpha, \tag{12}$$

where $\alpha \in \mathbb{R}^k$ is a convex weight (i.e., $\alpha_i \geq 0$ and $\sum_{i=1}^k \alpha_i = 1$). Hence, the descent cone can be represented as follows:

$$D(x_0) = \{v \in \mathbb{R}^n \backslash \{0\}\ :\ \exists \alpha \in \mathbb{R}^k \backslash \{0\} :\ \alpha_i \geq 0,\ DF(x_0)\nu = -\alpha\} \tag{13}$$

and thus, every descent direction $\nu_\alpha \in D(x_0)$ can be computed by solving the underdetermined system of linear equations (6) for a given vector $-\alpha = d$. Note that α has to be determined first, but by this ν_α gains a physical meaning: by construction, the direction in image space is given by

$$\langle \nabla f_i(x_0), \nu_\alpha \rangle = -\alpha_i, \quad i = 1, \ldots, k. \tag{14}$$

A very similar representation of the descent cone (albeit using a different derivation) can be found in [3].

A Curve of Dominating Points The above result can be used to define a curve of dominating points. Assume that a (not necessarily fixed) convex weight α is given and a search in that direction is desired. Using

$$\nu_\alpha(x) := -DF(x)^+\alpha, \tag{15}$$

one can thus try to solve numerically the following initial value problem:

$$\begin{aligned} x(0) &= x_0 \in \mathbb{R}^n \\ \dot{x}(t) &= \nu_\alpha(x(t)), \quad t > 0 \end{aligned} \tag{IVP$_\alpha$}$$

A solution of (IVP$_\alpha$) yields a curve of dominating points, and the proportion of the improvements of the single objectives is given by α as shown in (14). Clearly, if all objectives are continuously differentiable and if $\alpha = \alpha(x)$ is continuous in x the solution curve $x(t)$ is also continuously differentiable.
It has to be noted that even if an endpoint x^* of (IVP$_\alpha$) exists—for instance if

$F(\mathbb{R}^n)$ is bounded below—this point does not have to be Pareto optimal since the approach depends next to α on the initial point x_0. At least, a stopping criterion for the numerical treatment of (IVP$_\alpha$) can be given to detect if an endpoint of the curve is reached under certain (reasonable) assumptions: if (a) the number of parameters n is at least as large as the number of objectives k and (b) if the gradients of all objectives are linear independent at x_0, i.e., $rank(DF(x_0)) = k$ (which means that all objectives are indeed in conflict at x_0), then for every point $x(t)$ along the curve the rank of the Jacobian is k, except for the endpoint x^* (in particular $-\alpha$ is not in the image of $DF(x^*)$). The rank of a matrix can of course not be used to detect the endpoint of a curve numerically, but instead the condition number κ_2 of $DF(x)$ can be used: one can e.g. compute

$$\kappa_2(DF(x)) = \|A\|_2 \|A^+\|_2 = \frac{\sigma_1}{\sigma_k}, \tag{16}$$

where σ_1 and σ_k are the largest and smallest singular value of $DF(x)$, respectively, and stop the process if $\kappa_2(DF(x_i)) \geq tol$, where $tol \in \mathbb{R}_+$ is a given (large) threshold. This can be done since by the above discussion $\kappa_2(DF(x(t))) \to \infty$ for $x(t) \to x^*$.

This discussion shows one potential drawback of the approach, namely that the determination of the search direction by solving (12) gets inaccurate for points near the Pareto set due to the high condition number of $DF(x)$. However, our experience has shown that state-of-the-art numerical tools allow to come 'near enough' to the Paret set even for higher dimensional problems (here we refer to the numerical results presented in Section 6).

Algorithm 1 describes a possible algorithm to trace the solution curve of (IVP$_\alpha$). Hereby, for instance an Armijo-like step size control can be chosen for the choice of t ([2, 15]). Alternatively, of course problem (IVP$_\alpha$) can be solved using well-established numerical discretization methods (e.g., [12]).

So far, we have described a framework for possible search procedures since both (IVP$_\alpha$) and Algorithm 1 depend on the choice of α. In the following we discuss such possible choices.

Algorithm 1 Directed Search Method

```
Require: starting point $x_0 \in \mathbb{R}^n$ with $rank(DF(x_0)) = k$, $tol \in \mathbb{R}_+$, convex weight
    $\alpha_0 \in \mathbb{R}^k$.
$i := 0$
while $\kappa_2(DF(x_i)) < tol$ do
    compute $\nu_i = -DF(x_i)^+\alpha_i$
    compute $t_i \in \mathbb{R}_+$
    set $x_{i+1} := x_i + t_i\nu_i$
    choose $\alpha_{i+1} \in \mathbb{R}^k$
    set $i := i + 1$
end while
```

4.2 On the Choice of α and Implications for Related Methods

Here we discuss several possibilities for α and resulting implications and relations to well-known related methods. For the Directed Search Method and Goal Programming we will investigate the most greedy search direction indicating how the novel approach can be used in certain situations. As discussed above, the knowledge of a greedy direction can be beneficial if the function evaluation is costly and just a few iterations can be performed within a given time budget.

Weighted Sum Approach Given a convex weight α, the Weighted Sum method, which is probably the most famous scalarization method for MOPs, reads as follows:

$$\min_{x \in \mathbb{R}^n} \sum_{i=1}^{k} \alpha_i f_i(x). \tag{17}$$

Apparently, the relation of the Directed Search Method to the the Weighted Sum Approach is given by the search direction which is equal to the weight of the scalarization method: the vector ν_α for the same vector α can be viewed due to (14) as the most greedy search direction for problem (17).
This feature, however, does not neccessarily hold for the Weighted Sum Method: when e.g. solving problem (17) with the steepest descent method (i.e., choosing $\nu(x) := -\sum_{i=1}^{k} \alpha_i \nabla f_i(x) \in \mathbb{R}^n$ as search direction), the (infinitesimal) change in objective space is given by

$$\lim_{t \searrow 0} \frac{f_i(y_0) - f_i(x_0)}{t} = \langle \nabla f_i(x_0), -\sum_{i=1}^{k} \alpha_i \nabla f_i(x) \rangle, \tag{18}$$

which does certainly not have to be equal to $-\alpha_i$ for the i-th objective.
When comparing the two weighting approaches the main advantage of the Weighted Sum approach is certainly that every first order optimal point of (17) is a first order Pareto optimal solution of the MOP which does not neccessarily hold for an endpoint of a solution of (IVP$_\alpha$). On the other hand, it is known that it is not easy to detect optimal solutions x whose images $F(x)$ are located on a concave region of a Pareto front (since every global solution of (17) is located at the boundary of the Pareto front or within a convex part). This feature prevents the Weighted Sum Method from computing a suitable finite size representation of the entire Pareto front in general. By Equation (14), however, it follows that this problem does not hold for the novel approach.
Hence, if α is given by the application or the decision maker—the fact that the Weighted Sum approach is still used indicates that this is sometimes the case—we think that the Directed Search method can at least locally be seen as a potential alternative.

Goal Programming In Goal Programming (GP) the task is, roughly speaking, to find a point those image is as close as possible to a given target value $Z \in$

$\mathbb{R}^k$. This leads in the unconstrained case in its general form to the following optimization problem (see e.g. [5] for more information):

$$\min_{x \in \mathbb{R}^n} d(Z, F(x)), \tag{19}$$

where $d(\cdot,\cdot)$ is a chosen distance in $\mathbb{R}^k$. When choosing the Euclidean distance, apparently the (local) best search direction at a point x_0 is given by

$$\alpha_{Z,2} := \frac{F(x_0) - Z}{\|F(x_0) - Z\|_1}, \tag{20}$$

where we assume that $Z \leq_p F(x_0)$. To satisfy user preferences or to be able to reach several Pareto optimal points it is commonly desired to use a weighted metric instead of a fixed metric (e.g., [28]). When using the weighted 2-metric

$$d_D(x, y) := \sqrt{(x - y)^T D(x - y)}, \tag{21}$$

where D is a diagonal matrix with positive diagonal entries, the most greedy direction at x_0 is given by

$$\alpha_{Z,D} := \frac{D(F(x_0) - Z)}{\|D(F(x_0) - Z)\|_1} \tag{22}$$

This can be seen as follows: define for a given point x_0 and a fixed $\alpha \in \mathbb{R}^k$ the curve $c_\alpha : \mathbb{R} \to \mathbb{R}^k$ by

$$c_\alpha(t) = F(x_0) + t\alpha \tag{23}$$

Let $g_\alpha : \mathbb{R} \to \mathbb{R}$ be the square of the weighted 2-metric of Z and $c_\alpha(t)$, i.e.,

$$g_\alpha(t) = d_D(Z, c_\alpha(t))^2 = \sum_{i=1}^k d_i (Z_i - f_i(x_0) - t\alpha_i)^2, \tag{24}$$

where d_i is the i-th diagonal element of D. The derivative is given by

$$g'_\alpha(t) = -\sum_{i=1}^k 2 d_i \alpha_i (Z_i - f_i(x_0) - t\alpha_i), \tag{25}$$

and hence

$$g'_\alpha(0) = -2 \sum_{i=1}^k d_i \alpha_i (Z_i - f_i(x_0)) \tag{26}$$

Using (26) we can determine the most greedy choice of α: the steepest descent with respect to α is given by

$$-\nabla_\alpha (g'_\alpha(0)) = D(Z - F(x_0)), \tag{27}$$

and the claim follows (note that the directional vector is negated in (12)).

As for the Weighted Sum Method, the vectors $\nu(\alpha_{Z,2})$ and $\nu(\alpha_{Z,D})$ can be viewed as the ‘best’ local search direction for the Goal Programming (19) according to the given metric. While $\alpha_{Z,2}$ is constant, $\alpha_{Z,D}$ varies continuously with x in case F is continuous.

Steepest Descent Though the choices of α presented above are locally optimal according to their purposes, none of them guarantees that the solution curve of (IVP$_\alpha$) converges to a (local) Pareto point. Based on the famous theorem of Kuhn and Tucker, a suggesting method—at least in light of (12)—to choose α at a given point x is for instance to take

$$\begin{aligned} \alpha_{SD} \in \arg\min_{\alpha} \left\| \sum_{i=1}^{k} \alpha_i \nabla f_i(x) \right\|_2^2, \\ \text{s.t.}\ \ \alpha_i \geq 0,\ i = 1, \ldots, k \\ \sum_{i=1}^{k} \alpha_i = 1. \end{aligned} \tag{28}$$

Doing so, a numerical solution of (IVP$_\alpha$), where (28) is chosen for each point $x(t)$, can be viewed as a steepest descent method for MOPs. This has already been investigated in [15, 31], where the descent direction is instead of (12) computed by solving the scalar optimization problem (28) and using the search direction $\nu_{SD} = -\sum_{i=1}^{k} \alpha_{SD,i} \nabla f_i(x)$ (hence, for $k = 1$ the descent direction is always given by $-\nabla f_1(x)$ resulting in the classical steepest descent method for scalar optimization problems). In that case the solution curve always ends in a KKT point, and this holds also for suitable numerical solutions of the initial value problem ([15]). On the other hand, by choosing α_{SD}, its values are given implicitly which implies that an 'external steering' of the search is hardly possible. In particular for some set oriented approaches for the treatment of MOPs such as subdivision techniques ([11]) or memetic strategies ([3, 24, 38]) any bias of the descent method is undesired since this could lead to insufficient approximations of the Pareto set.

Normal Boundary Intersection There is at first sight a strong relation of the Directed Search method to the Normal Boundary Intersection (NBI, see [9]), namely the (quasi) normal direction η obtained by the convex hull of the individual minima of the objectives (CHIM) is a particular choice of α. On the other side, the direction η in NBI is a tool to obtain a suitable spread of the solutions obtained by a sequence of scalarizations of the MOP, and the search in direction η is not performed directly in contrast to the Directed Search Method. Nevertheless, we think the particular choice of η can for instance be beneficial when using the Directed Search Method within a memetic strategy: if a rough approximation of the Pareto front is at hand, the CHIM can be approximated, and thus, η can be chosen as in NBI in order to obtain a sufficient spread of the solutions. We do, however, not explore this idea here but leave it for future research.

5 A Continuation Algorithm Based on the Directed Search Approach

In this section we propose a new Predictor Corrector (PC) method for the continuation along (local) Pareto sets of a given MOP. The central difference to a classical method is that we suggest a new predictor direction which is based on the geometry of the Pareto front and realized by the Directed Search Method. Interesting is the fact that this new method—unlike classical PC methods—does not require to compute the Hessians of the objectives. This implies that this method can be used to handle even higher dimensional models ($n >> 1000$) without exploiting the possibly given sparsity of the system. In the following we present the alternative predictor direction and propose then a 'complete' PC method which does not require any 2nd derivative information.

5.1 An Alternative Predictor Direction

Assume we are given a (local) Pareto point x and the related convex weight α, i.e., such that

$$\sum_{i=1}^{k} \alpha_i \nabla f_i(x) = 0 \tag{29}$$

and further we assume that

$$rank(DF(x)) = k - 1 \tag{30}$$

It is known (e.g., [19]) that in this case α is orthogonal to the Pareto front, i.e.,

$$\alpha \perp T_y \partial F(\mathbb{R}^n), \tag{31}$$

where $y = F(x)$ and $\partial F(\mathbb{R}^n)$ denotes the border of the image $F(\mathbb{R}^n)$. Thus, a search orthogonal to α (in objective space) could be promising to obtain new predictor points. To use the direct approach (12), for instance a QR-factorization of α can be computed, i.e.,

$$\alpha = QR, \tag{32}$$

where $Q = (q_1, \dots, q_k) \in \mathbb{R}^{k\times k}$ is an orthogonal matrix and $q_i,\ i = 1, \dots, k$, its column vectors, and $R = (r_{11}, 0 \dots, 0)^T \in \mathbb{R}^{k\times 1}$ with $r_1 1 \in \mathbb{R}\backslash\{0\}$ (for the computation of such a factorization we refer e.g. to [29]). Since by (32) $\alpha = r_{11}q_1$, i.e., $\alpha \in span\{q_1\}$, and Q orthogonal it follows that the column vectors $q_2, \dots, q_k$ build an orthonormal basis of the hyperplane which is orthogonal to α. Thus, a promising well-spread set of search directions ν_i may be the ones which satisfy

$$DF(x)\nu_i = q_i, \quad i = 2, \dots, k. \tag{33}$$

Since α is not in the image of $DF(x)$ (else x would not be a Pareto point) and by assumption (30) it follows that the vectors $q_2, \dots, q_k$ are in the image of $DF(x)$, i.e., Equation (33) can be solved for each $i \in \{2, \dots, k\}$. Note that by this choice

of predictor direction no second derivative of the objectives are required.
In contrast, for instance in [19] the zero set of $\tilde{F} : \mathbb{R}^{n+k} \to \mathbb{R}^{n+1}$ is traced, where

$$\tilde{F}(x, \alpha) = \begin{pmatrix} \sum_{i=1}^{k} \alpha_i \nabla f_i(x) \\ \sum_{i=1}^{k} \alpha_i - 1 \end{pmatrix}. \tag{34}$$

Predictor directions at a point (x_0, α_0) are found by linearizing the solution set in parameter space, and this can be realized by a QR-factorization of $D\tilde{F}(x_0, \alpha_0)$ which requires the second derivative of each objective.

5.2 A New PC Variant for Multi-Objective Optimization

Using the above observation for a possible predictor direction we construct in the following a PC method which does not require any second deriative information. Hereby, we concentrate on the bi-objective case (i.e., $k = 2$) since a consideration of $k > 2$ requires an additional data structure for the efficient representation of the approximation (for this we refer e.g. to [18, 34, 33]). Apart from that and the orientation all subsequent ideas will apply for models with $k > 2$.

Predictor Assume we are given a Pareto point x_0 with associated weight α_0 as in (29). The predictor direction can—except for its sign—be chosen as described above, i.e., one of the normalized vectors $\nu := \pm\nu_2/\|\nu_2\|_2$, where ν_2 satisfies (6) as described above for $d = q_2$. To orientate the curve (i.e., to determine the sign of ν) we can not proceed as for 'classical' PC methods since this would require as well the derivative of $\tilde{F}$ as in (34). Instead, one can define an orientation in the context of bi-objective optimization by the increase (or decrease) of one objective. For this, the signum of the according entry of the direction vector q_2 can be taken. If, for instance, an improvement according to f_2 is sought, then

$$p := x_0 - sgn(q_{2,2})t\nu \tag{35}$$

can be chosen as predictor, where $q_{2,2}$ denotes the 2nd entry of q_2, and t is the desired step size. Note that when choosing one of the data structures in [18, 34, 33] for models with $k > 2$ no orientation of the solution manifold is required.
To choose the step size t we suggest to proceed as follows: assume we are given x_0 and the search direction ν with $\|\nu\|_2 = 1$ associated to the direction q in objective space for the predictor, i.e., $p = x_0 + t\nu$, where $t \in \mathbb{R}_+$ has to be chosen. The underlying idea of the step size control is as follows: to obtain an adequate spread of the solutions the function values $f_j(x)$ and $f_j(p)$ of at least one objective $j \in \{1, \dots, k\}$ differ ideally by a (problem dependent) value ϵ while the difference for all other objectives do not exceed this threshold. Since this value can differ for each objective the demand on the spread can be stated (after possible renormalization) as follows:

$$d_w(F(p), F(x)) \approx \epsilon \tag{36}$$

where d_w is the weighted infinity distance, i.e.,

$$d_w(x, y) = \sum_{i=1}^{k} w_i |x_i - y_i|. \tag{37}$$

Assuming that each objective is Lipschitz continuous and that the step size t_i for the i-th objective is sufficiently small we obtain for $i = 1, \dots, k$:

$$\underbrace{|f_i(p) - f_i(x)|}_{\overset{!}{=} \frac{\epsilon}{w_i}} \approx L_{i,x} \underbrace{\|p - x\|_2}_{=t_i} \tag{38}$$

Since $L_{i,x}$ can be approximated by the norm of the directional derivative we obtain for each objective the control

$$t_i = \frac{\epsilon}{w_i |\langle \nabla f_i(x), \nu \rangle|}, \quad i = 1, \dots, k, \tag{39}$$

and hence for the entire MOP

$$t := \min_{i=1,\dots,k} t_i. \tag{40}$$

By construction, the difference vector $F(p) - F(x)$ is ideally orthogonal to α which can be used to determine if the chosen step size (40) is too large. If

$$|\langle \alpha, F(p) - F(x) \rangle| \leq tol, \tag{41}$$

where $tol \in \mathbb{R}_+$ is a given tolerance, the predictor p can be accepted. If (41) is not true, then p does probably not serve as a good predictor, and the step size has to be decreased accordingly.
Alternative step size controls for multi-objective continuation can be found e.g. in [19, 36].

Corrector Given a predictor p, the subsequent solution along the curve can be computed by solving numerically (IVP$_\alpha$), using p as initial value and choosing α_0, i.e., the weight from the previous solution x_0 leading to a new solution x_1. This together with the step size control (40) and the 'quasi-orthogonality' test (41) is intended to obtain an even spread of the solutions. In fact, this is the case if the value of ϵ in (36) and hence the step size t is sufficiently small. The new associated weight α_1 can be updated by solving the following quadratic optimization problem (28).

Algorithm 2 shows a possible realization of the continuation method which does not require the second derivatives of the objectives. Hereby, we start with an approximate minimizer of the first objective f_1 and trace the curve seeking for improvements according to the second objective f_2. Hence, a possible stopping criterion is that the associated weight of a candidate solution is approximately $\tilde{\alpha} = (0, 1)$. Other stopping criteria, however, are possible according to the given setting.

Algorithm 2 Bi-Objective Continuation

```
Require: Initial solution $(x_0, \alpha_0)$ with $\alpha_0 \approx 1$, threshold $\epsilon \in \mathbb{R}_+$, tolerance $\delta \in \mathbb{R}_+$.
Ensure: Set of candidate solutions $x_i$
$i := 0$
while $1 - \alpha_2 > \delta$ do
    compute $q_2$ as in (32)
    compute $\nu$ as in (33) (i.e., $\nu := \nu_2$)
    compute $t$ as in (40)
    $p_i := x_i - sgn(q_{2,2}) t \nu$
    compute $x_{i+1}$ by solving (IVP$_\alpha$) with initial value $p_i$ and using $\alpha_i$.
    compute $\alpha_{i+1}$ as in (28)
    set $i := i + 1$
end while
```

6 Numerical Results

Here we present some numerical results to illustrate both the Directed Search descent method as well as the novel continuation approach. All computations have been done using MATLAB[2].

6.1 Example 1

First we consider the following parameter dependent MOP ([40]):

$$
\begin{aligned}
&f_1, f_2 : \mathbb{R}^2 \to \mathbb{R} \\
&f_1(x, y) = \frac{1}{2}\left(\sqrt{1 + (x + y)^2} + \sqrt{1 + (x - y)^2} + x - y\right) + \lambda \cdot e^{(-x-y)^2} \\
&f_2(x, y) = \frac{1}{2}\left(\sqrt{1 + (x + y)^2} + \sqrt{1 + (x - y)^2} - x + y\right) + \lambda \cdot e^{(-x-y)^2}
\end{aligned}
\tag{42}
$$

For $\lambda = 0.85$ the Pareto front contains a dent. To be more precise, the Pareto front is connected and consists of one concave and two convex parts.

Figure 1 shows a comparison of the Weighted Sum approach using the steepest descent method and the Directed Search method. Starting point for both methods was $x_0 = (1.5, 1.5)$, and the weight vector was chosen as $\alpha = 1/\sqrt{2}(1, 1)^T$. For the stopping criterion (16) we have chosen for this as well as for the subsequent examples $tol = 1e8$. To demonstrate the solution curves we have chosen very small step sizes. While the solution curve of the Directed Search Method steers one a straight line from $F(x_0)$ to the corresponding point on the Pareto front (and in this case the same appears in parameter space), the solution curve of the Weighted Sum approach eludes this straight line leading to a (for this method) better solution. We have repeated this for 100 randomly chosen starting points within $Q = [-5, 5]^2$. Figure 2 shows the resulting endpoints for both methods. While it may be argued that for a single solution the Weighted Sum

[2] `http://www.mathworks.com`

approach obtained a better solution (Figure 1), this does apparently not hold in terms of a possible representation of the *entire* Pareto set since the concave part of the Pareto front is left out nearly completely.

6.2 Example 2

Next we consider the following convex MOP:

$$\begin{aligned} & f_1, f_2 : \mathbb{R}^n \to \mathbb{R} \\ & f_i(x) = \sum_{\substack{j=1 \\ j \neq i}}^{n} (x_j - a_j^i)^2 + (x_i - a_i^i)^4, \end{aligned} \tag{43}$$

where

$$\begin{aligned} a^1 &= (1,1,1,1,\ldots) && \in \mathbb{R}^n \\ a^2 &= (-1,-1,-1,-1,\ldots) && \in \mathbb{R}^n, \end{aligned}$$

First we turn our attention to the goal programming problem (19) where we choose the target value as $Z = (0,0)$. Figure 3 shows some numerical solution curves for several starting points where the Euclidean Distance has been chosen. In Figure 4 some numerical results are shown for several problems where x_0 has been fixed, but the weighed 2-metric has been chosen for different matrices. To be more precise, the curves $c_i, i = 1, \ldots, 5$ belong to the matrices D_i, where

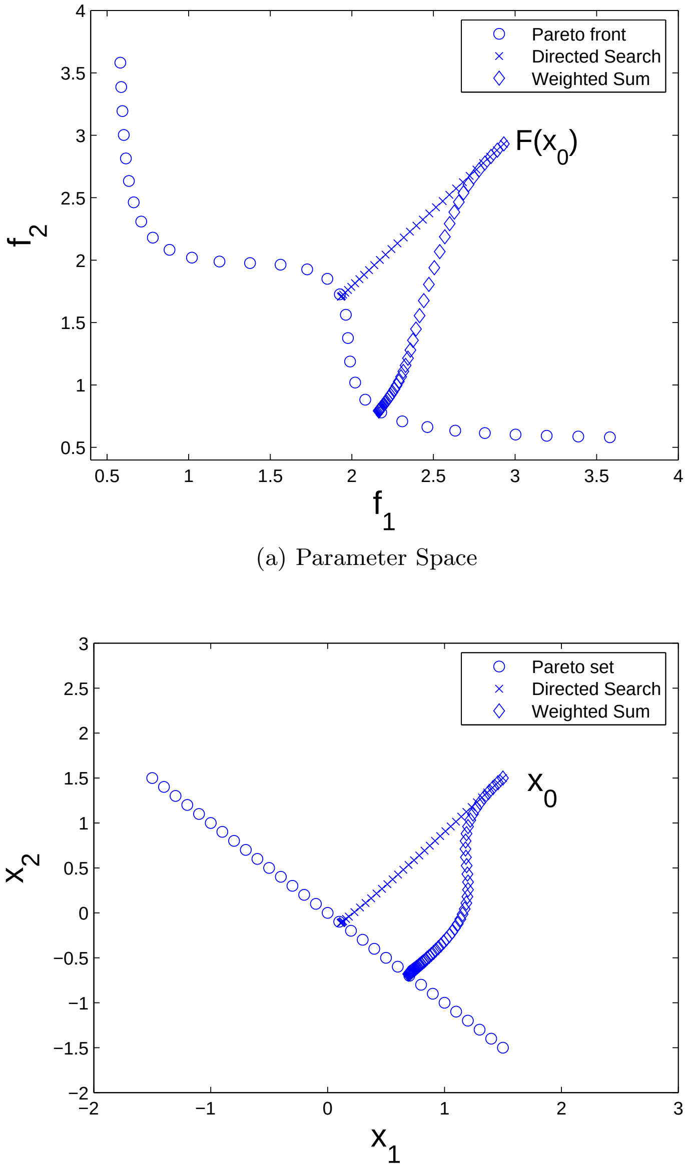
$$D_1 = \begin{pmatrix} 0.9 & 0 \\ 0 & 0.1 \end{pmatrix}, \quad D_2 = \begin{pmatrix} 0.7 & 0 \\ 0 & 0.3 \end{pmatrix}, \quad D_3 = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix},$$

$$D_4 = \begin{pmatrix} 0.3 & 0 \\ 0 & 0.7 \end{pmatrix}, \qquad D_5 = \begin{pmatrix} 0.3 & 0 \\ 0 & 0.7 \end{pmatrix}.$$

Next, we are interested in solving MOP (43) by continuation methods. Figure 5 shows a numerical result for $n = 10$. The figure shows the images of the solutions $F(x_i)$ as well as the images of the predictors $F(p_i)$ which are already near to the solutions.

We have observed that the corrector step uses approximately three iterations to be near enough to the Pareto set (to be more precise: in the above example we needed in average 2.4 iterations). We use this to make a comparison to the classical PC method:

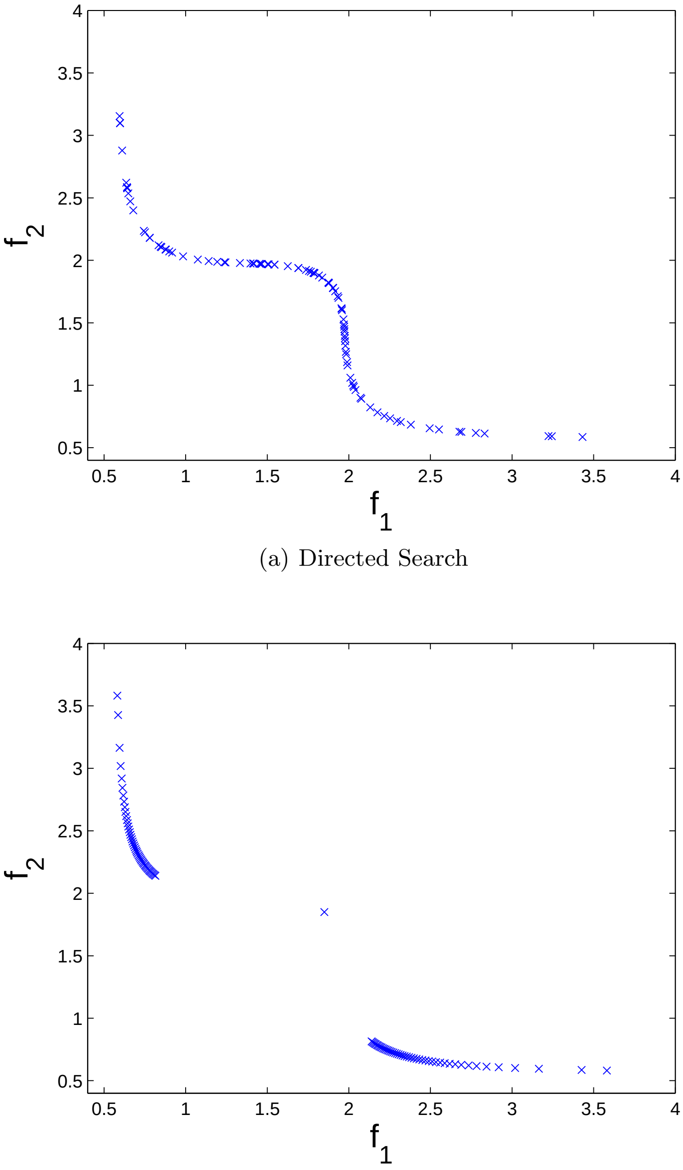
the classical PC method uses in each predictor step one QR-factorization of $(\tilde{F}')^T(x,\alpha)$ which implies one Jacobian call (for all objectives) and one Hessian call. For the corrector at least one Gauss-Newton step has to be performed which implies again one Jacobian and one Hessian call (here we neglect possible additional function calls due to backtracking strategies). In total, we can assume that for the generation of a new candidate solution two Jacobians and two Hessians have to be calculated. The PC method as described above requires one Jacobian call for the predictor and we estimate three Jacobians for the corrector (as observed in this example), which makes in total the computation of four Jacobians



(a) Parameter Space

(b) Objective Space

Fig. 1. Numerical result for MOP (42): comparison of the numerical solution paths of the weighted sum and the directed search approach for $x_0 = (1.5, 1.5)$ and $\alpha = 1/\sqrt{2}(1, 1)^T$.



(a) Directed Search

(b) Weighted Sum

Fig. 2. Numerical result for MOP (42): comparison of the solutions (end points) of the weighted sum and the directed search approach for $\alpha = 1/\sqrt{2}(1,1)^T$ and 100 randomly chosen initial points.

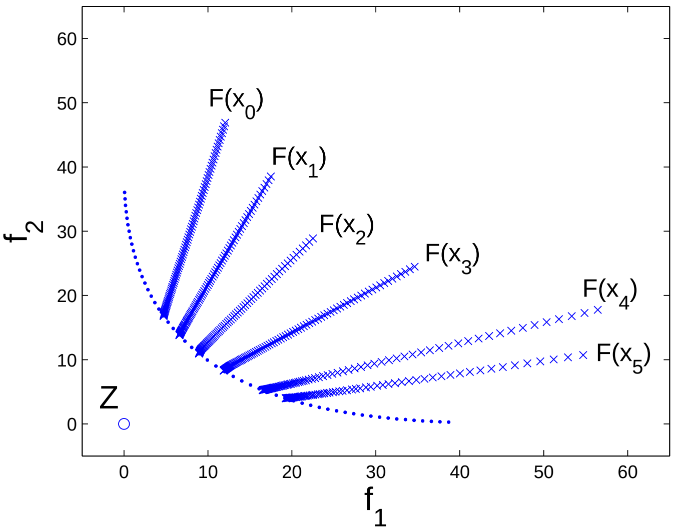


Fig. 3. Numerical result for MOP (43): solution paths for several initial conditions x_i, $i = 0, \ldots, 5$, using the 2-metric for d.

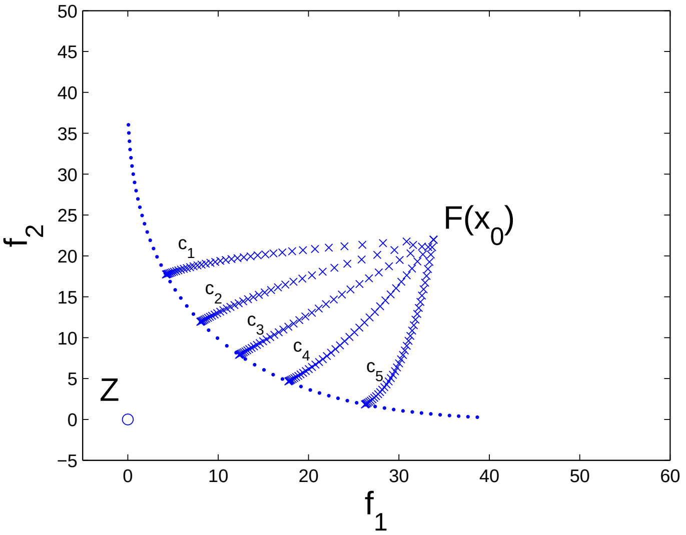


Fig. 4. Numerical result for MOP (43): solution paths for one initial point and for several weighted metrics (see text).

to obtain a new solution. For sake of comparison, we assume that the gradient information is not available but has to be calculated or approximated, and hence measure the cost to find a new candidate solutions in terms of the number of required function calls. If for instance automatic differentiation (AD) is used to compute the derivatives, we can estimate $5 * k$ function calls for each derivative call and $k*(4+6n)$ function calls for each calculation of the Hessian ([17]). Then we obtain for our example ($n = 10$, $k = 2$):

$$|fc(PC_{classical})| = 276, \quad |fc(PC_{new})| = 40, \tag{44}$$

where $fc(A)$ denotes the number of required function calls for method A. These values change when using finite differences (FD). If for instance the forward difference quotient

$$\frac{\partial f}{\partial x_i}(x) \approx \frac{f(x_1, \dots, x_i + \delta_i, \dots, x_n) - f(x_1, \dots, x_n)}{\delta_i}, \quad i \in \{1, \dots, n\} \tag{45}$$

where $\delta_i \in \mathbb{R}_+$ is a small value, is used to estimate the gradient, apparently n function calls are required to estimate each gradient. The central difference quotient leads to more accurate approximations, but does in turn require $2n$ function calls ([17]). A forward difference quotient approximation of the second derivative requires a total of n^2 function calls (and $2n^2$ or $4n^2$ function calls when using the central difference quotient, depending on how the rule is applied). Hence, using the forward difference rule in FD we obtain

$$|fc(PC_{classical})| = 440, \quad |fc(PC_{new})| = 80. \tag{46}$$

In both cases,i.e., AD and FD, the new PC method requires a much smaller amount of function calls for the computation of the subsequent solution.

Figure 6 shows a numerical result for the same model and the same setting but using $n = 100,000$ parameters. Also here, the cost to obtain a new candidate solution is approximately given by four Jacobian calls (we omit here a comparison as in (44) and (46)). Since a naive storage of the Hessian matrix (i.e., without exploiting the sparsity) requires the size of n^2 floats, a straightforward implementation of the classical PC method is restricted on a standard computer to approximately $2,000$ free parameters, which does not hold for the new method. Since the second derivatives are not needed for the latter approach, we think that this one is—independent of the storage problem—a promising alternative to the classical PC method in particular for higher dimensional problems.

6.3 Example 3

Finally, we reconsider MOP (42) from Example 1 and turn our attention again to Goal Programming. Though it was shown above, that α_Z is *locally* the best search direction, it is not even guaranteed that a solution of (IVP$_\alpha$) leads to a local solution of (19). This is due to the fact that the approach depends next to x_0 and Z also on the shape of the Pareto front which is clearly a priori not known.

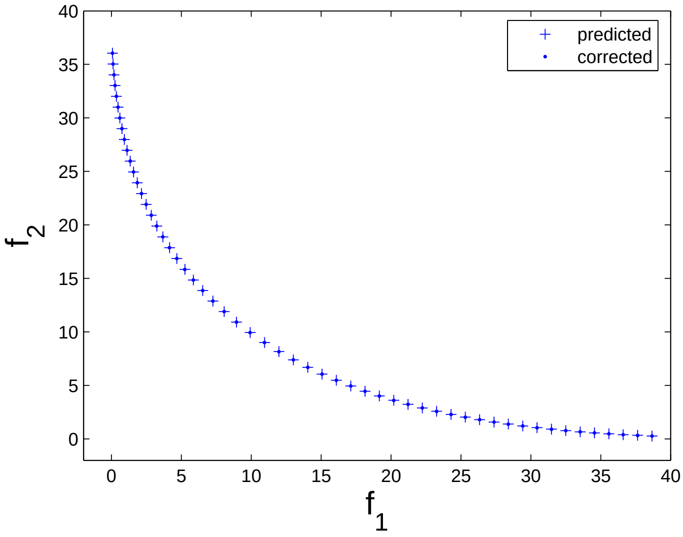


Fig. 5. Numerical result for MOP (43): solution of the continuation algorithm for $n = 10$. Shown are the images of the predictors and the solutions.

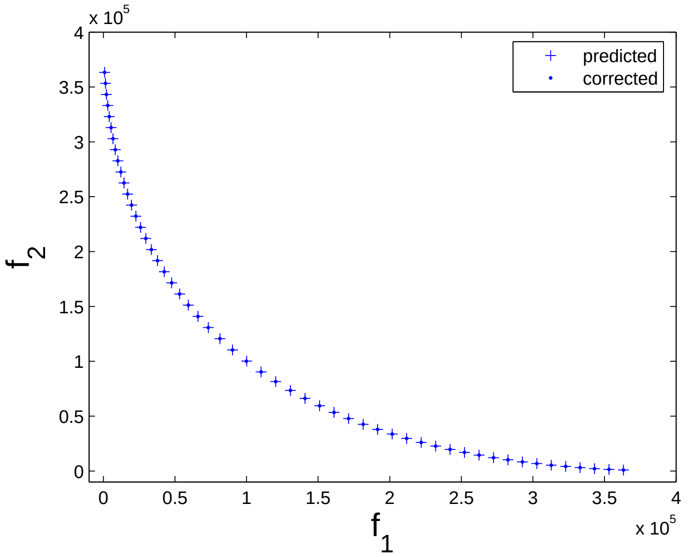


Fig. 6. Numerical result for MOP (43): solution of the continuation algorithm for $n = 100,000$. Shown are the images of the predictors and the solutions.

As a possible remedy it seems possible to combine the two methods proposed in this work to obtain a two-stage algorithm for the detection of local solutions of (19): first, one can compute, starting from x_0, the endpoint x^* those image is on the border of $F(\mathbb{R}^n)$. In the second step, a movement along $\partial F(\mathbb{R}^n)$ can be performed seeking for a decrease of the distance toward Z.
Figure 7 shows such an example for MOP (42). The continuation has been done as described above but we have changed the orientation according to the problem, and stopped the process as soon as the search direction flipped (monitored by the value of q_2). Though the first result is quite promising, we feel that much more investigation has to be done in that direction which we leave for future research.

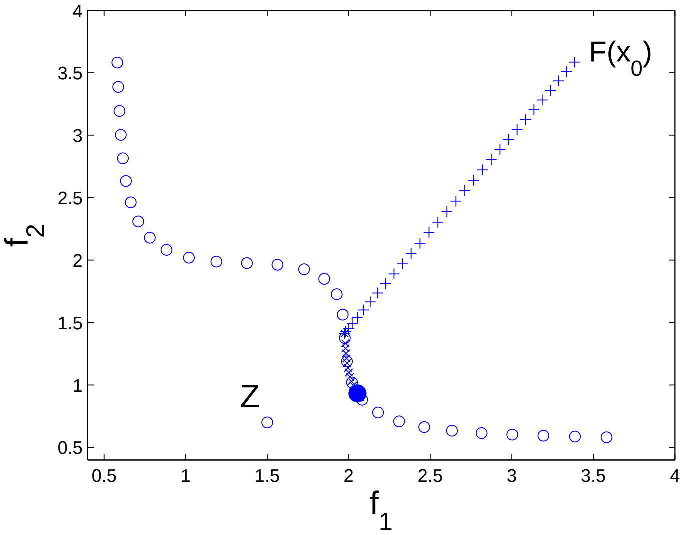


Fig. 7. Numerical result for MOP (43): a combination of the Directed Search descent method (indicated by the stars) and a modified continuation procedure (crosses) leads to the solution of the goal programming problem (filled circle).

7 Conclusions and Future Work

We have presented the Directed Search Method for multi-objective optimization which allows to steer the search into any given direction d in objective space. Based on this idea we have presented a new class of descent methods and a novel continuation procedure. The descent methods depend next to the initial point x_0 on the choice of d. For this, we have discussed several possibilities, among others

the greedy search directions for the Weighted Sum approach and for Goal Programming. Though these directions are locally optimal, the resulting approaches do, however, not guarantee that at least a locally optimal point is reached. The essential novelty in the alternative continuation method we have presented is the choice of the predictor direction. Instead of linearizing the Pareto *set* we have used a linearization of the Pareto *front* to obtain a new predictor solution. By this, no 2nd gradient information is required as in 'classical' continuation methods which makes the new strategy a competitive alternative in particular for the treatment of higher dimensional problems. Finally, we have illustrated behavior of both methods on some benchmark problems.
For the future, there are some interesting topics which can be addressed to advance the present work. For instance, further choices of the search direction d for the descent method would be interesting together with the investigation of their convergence properties. Another interesting issue is the application of the Directed Search Method to other problems. For instance, as a local searcher inside a memetic strategy, or on a high-dimensional real-world engineering problem.

References

1. E. L. Allgower and K. Georg. *Numerical Continuation Methods.* Springer, 1990.
2. L. Armijo. Minimization of functions having Lipschitz-continuous first partial derivatives. *Pacific Journal of Mathematics*, 16:1–3, 1966.
3. P. A. N. Bosman and E. D. de Jong. Exploiting Gradient Information in Numerical Multi-Objective Evolutionary Optimization. In Hans-Georg Beyer et al., editor, *2005 Genetic and Evolutionary Computation Conference (GECCO'2005)*, volume 1, pages 755–762, New York, USA, June 2005. ACM Press.
4. R. Bramley and B. Winnicka. Solving linear inequalities in a least squares sense. *SIAM Journal on Scientific Computing*, 17(1):275–286, 1996.
5. A. Charnes and W. W. Cooper. *Management models and industrial applications of linear programming.* Wiley, New York, 1961.
6. C. A. Coello Coello and A. D. Christiansen. Multiobjective optimization of trusses using genetic algorithms. *Computers and Structures*, 75(6):647–660, May 2000.
7. C. A. Coello Coello and G. B. Lamont, editors. *Applications of Multi-Objective Evolutionary Algorithms.* World Scientific, Singapore, 2004. ISBN 981-256-106-4.
8. C. A. Coello Coello, G. B. Lamont, and D. A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems.* Springer, New York, second edition, September 2007. ISBN 978-0-387-33254-3.
9. I. Das and J. Dennis. *Normal-boundary intersection*: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal of Optimization*, 8:631–657, 1998.
10. K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms.* John Wiley & Sons, Chichester, UK, 2001. ISBN 0-471-87339-X.
11. M. Dellnitz, O. Schütze, and T. Hestermeyer. Covering Pareto sets by multilevel subdivision techniques. *Journal of Optimization Theory and Applications*, 124:113–155, 2005.
12. P. Deuflhard and F. Borneman. *Scientific Computing with Ordinary Differential Equations.* Texts in Applied Mathematics 42. Springer New York, 2002.

13. G. Eichfelder. *Adaptive Scalarization Methods in Multiobjective Optimization.* Springer, Berlin Heidelberg, 2008. ISBN 978-3-540-79157-7.
14. J. Fliege. Gap-free computation of Pareto-points by quadratic scalarizations. *Mathematical Methods of Operations Research*, 59:69–89, 2004.
15. J. Fliege and B. Fux Svaiter. Steepest descent methods for multicriteria optimization. *Mathematical Methods of Operations Research*, 51(3):479–494, 2000.
16. E. A. Portilla Flores, E. Mezura Montes, J. Alvarez Gallegos, C. A. Coello Coello, and C. A. Cruz Villar. Integration of structure and control using an evolutionary approach: an application to the optimal concurrent design of a cvt. *International Journal for Numerical Methods in Engineering*, 71(8):883–901, 2007.
17. A. Griewank. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation.* Number 19 in Frontiers in Appl. Math. SIAM, Philadelphia, PA, 2000.
18. M. E. Henderson. Multiple parameter continuation: Computing implicitly defined k-manifolds. *International Journal of Bifurcation and Chaos*, 12:451–476, 2002.
19. C. Hillermeier. *Nonlinear Multiobjective Optimization - A Generalized Homotopy Approach.* Birkhäuser, 2001.
20. H. Ishibuchi, T. Yoshida, and T. Murata. Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. *IEEE Transactions on Evolutionary Computation*, 7(2):204–223, 2003.
21. J. Jahn. Multiobjective search algorithm with subdivision technique. *Computational Optimization and Applications*, 35(2):161–175, 2006.
22. W. E. Karush. *Minima of functions of several variables with inequalities as side conditions.* PhD thesis, University of Chicago, 1939.
23. K. Klamroth, J. Tind, and M. Wiecek. Unbiased approximation in multicriteria optimization. *Mathematical Methods of Operations Research*, 56:413–437, 2002.
24. J. Knowles and D. Corne. Memetic Algorithms for Multiobjective Optimization: Issues, Methods and Prospects. In William E. Hart, N. Krasnogor, and J.E. Smith, editors, *Recent Advances in Memetic Algorithms*, pages 313–352. Springer. Studies in Fuzziness and Soft Computing, Vol. 166, 2005.
25. H. Kuhn and A. Tucker. Nonlinear programming. In J. Neumann, editor, *Proceeding of the 2nd Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492, 1951.
26. A. Kumar, P. B. Nair, A. J. Keane, and S. Shahpar. Robust design using Bayesian Monte Carlo. *International Journal for Numerical Methods in Engineering*, 73(11):1497–1517, 2008.
27. A. Lara, G. Sanchez, C. A. Coello Coello, and O. Schütze. HCS: A new local search strategy for memetic multiobjective evolutionary algorithms. IEEE Transactions on Evolutionary Computation (to appear).
28. K. Miettinen. *Nonlinear Multiobjective Optimization.* Kluwer Academic Publishers, Boston, Massachusetts, 1999.
29. J. Nocedal and S. Wright. *Numerical Optimization.* Springer Series in Operations Research and Financial Engineering. Springer, 2006.
30. V. Pareto. *Manual of Political Economy.* The MacMillan Press, 1971 (original edition in French in 1927).
31. S. Schäffler, R. Schultz, and K. Weinzierl. A stochastic method for the solution of unconstrained vector optimization problems. *Journal of Optimization Theory and Applications*, 114(1):209–222, 2002.
32. O. Schuetze, L. Jourdan, T. Legrand, E.-G. Talbi, and J.-L. Wojkiewicz. New analysis of the optimization of electromagnetic shielding properties using conducting

polymers and a multi-objective approach. *Polymers for Advanced Technologies*, 19(7):762–769, July 2008.

33. O. Schütze, C. A. Coello Coello, S. Mostaghim, E.-G. Talbi, and M. Dellnitz. Hybridizing Evolutionary Strategies with Continuation Methods for Solving Multi-Objective Problems. *Engineering Optimization*, 40(5):383–402, May 2008.
34. O. Schütze, A. Dell'Aere, and M. Dellnitz. On continuation methods for the numerical treatment of multi-objective optimization problems. In Jürgen Branke, Kalyanmoy Deb, Kaisa Miettinen, and Ralph E. Steuer, editors, *Practical Approaches to Multi-Objective Optimization*, number 04461 in Dagstuhl Seminar Proceedings. Internationales Begegnungs- und Forschungszentrum (IBFI), Schloss Dagstuhl, Germany, 2005. <http://drops.dagstuhl.de/opus/volltexte/2005/349>.
35. O. Schütze, M. Laumanns, C. A. Coello Coello, M. Dellnitz, and E.-G. Talbi. Convergence of stochastic search algorithms to finite size Pareto set approximations. *Journal of Global Optimization*, 41(4):559–577, 2008.
36. O. Schütze, M. Laumanns, E. Tantar, C. A. Coello Coello, and E.-G. Talbi. Convergence of stochastic search algorithms to gap-free Pareto front approximations. In *Genetic and Evolutionary Computation Conference (GECCO-2007)*, pages 892–901, 2007.
37. O. Schütze, M. Vasile, O. Junge, M. Dellnitz, and D. Izzo. Designing optimal low thrust gravity assist trajectories using space pruning and a multi-objective approach. *Engineering Optimization*, 41(2):155–181, 2009.
38. A. Sinha, Y.-P. Chen, and D. E. Goldberg. Designing Efficient Genetic and Evolutionary Algorithm Hybrids. In William E. Hart, N. Krasnogor, and J.E. Smith, editors, *Recent Advances in Memetic Algorithms*, pages 259–288. Springer. Studies in Fuzziness and Soft Computing, Vol. 166, 2005.
39. Z. Tang, J. Périaux, G. Bugeda, and E. Oñate. Lift maximization with uncertainties for the optimization of high-lift devices. *International Journal for Numerical Methods in Fluids*, 2009. (in press) DOI: 10.1002/fld.2130.
40. K. Witting and M. Hessel von Molo. Private communication, 2006.