# The Gradient Free Directed Search Method as Local Search within Multi-Objective Evolutionary Algorithms

Adriana Lara, Sergio Alvarado, Shaul Salomon, Gideon Avigad, Carlos A. Coello Coello, and Oliver Schütze

**Abstract.** Recently, the Directed Search Method has been proposed as a point-wise iterative search procedure that allows to steer the search, in any direction given in objective space, of a multi-objective optimization problem. While the original version requires the objectives' gradients, we consider here a possible modification that allows to realize the method without gradient information. This makes the novel algorithm in particular interesting for hybridization with set oriented search procedures, such as multi-objective evolutionary algorithms.

In this paper, we propose the DDS, a gradient free Directed Search method, and make a first attempt to demonstrate its benefit, as a local search procedure within a memetic strategy, by integrating the DDS into the well-known algorithm MOEA/D. Numerical results on some benchmark models indicate the advantage of the resulting hybrid.

## 1 Introduction

Many real world problems demand for the concurrent optimization of $k$ objectives leading to a *multi-objective optimization problem* (MOP) [16]. One characteristic of these problems, compared with those where only *one* objective is under consideration, is that the solution set of a MOP (the *Pareto set*) typically forms a

Adriana Lara
Mathematics Department ESFM-IPN, Edif. 9 UPALM, 07300 Mexico City, Mexico
e-mail: `adriana@esfm.ipn.mx`

Sergio Alvarado · Carlos A. Coello Coello · Oliver Schütze
Computer Science Department, CINVESTAV-IPN, Av. IPN 2508,
Col. San Pedro Zacatenco, 07360 Mexico City, Mexico
e-mail: `{ccoello,schuetze}@cs.cinvestav.mx`,
`salvarado@computacion.cs.cinvestav.mx`

Shaul Salomon · Gideon Avigad
ORT Braude School of Engineering, Karmiel, Israel
e-mail: `{shaulsal,gideona}@braude.ac.il`

$(k-1)$-dimensional object. So far, there exist many methods for the computation of the Pareto set of a MOP. Among them, multi-objective evolutionary algorithms (MOEAs) have caught the attraction of many researchers (e.g., [7, 6] and references therein). The major reason for this might be that the population based approach, together with a stochastic component in the search procedure, allows typically for an approximation of the entire (global) Pareto set in one single run of the algorithm. This represents an advantage over most mathematical programming (MP) techniques, which require in addition certain smoothness assumptions on the MOP. On the other hand, it is well-known that MOEAs normally need a large amount of function evaluations, due to their slow convergence rate, in order to generate a suitable finite size approximation of the set of interest ([4]). As a remedy, researchers have proposed *memetic MOEAs*, i.e., hybrids of MOEAs and MP with the aim to get fast and reliable global search procedures (e.g., [9, 11, 10, 22, 14]).
In this paper, we adapt the Directed Search (DS) method [21] for the use within MOEAs. One crucial drawback of the DS is that it requires gradient information which restricts its usability. Here, we propose a modification of the DS that is gradient free. Even more, the computation of the search direction comes without the cost of additional function evaluations if the neighborhood information can be exploited. The latter makes the Discrete Directed Search (DDS) a suitable algorithm, in particular, for the usage within set oriented search techniques. We demonstrate the benefit of the DDS by hybridizing it with MOEA/D ([24]), a state-of-the-art MOEA whose neighborhood definition can be directly used for the DDS.
The remainder of this paper is organized as follows: In Section 2, we give the required background for the understanding of the sequel. In Section 3, we present the DDS, a gradient free Directed Search variant. In Section 4, we propose a way to integrate the DDS into MOEA/D leading to a new memetic algorithm. In Section 5, we present some results, and finally, we draw our conclusions in Section 6.

## 2 Background

In the following we consider unconstrained multi-objective optimization problems (MOPs) which can be stated as follows:

$$\min_{x\in\mathbb{R}^n} F(x), \tag{1}$$

where $F : R \subset \mathbb{R}^n \to \mathbb{R}^k$ is defined as the vector of $k$ objective functions $f_i : R \subset \mathbb{R}^n \to \mathbb{R}$, $i = 1,\ldots,k$. A point $x \in R$ is said to dominate another point $y \in R$, if $f_i(x) \le f_i(y)$ for all $i \in \{1,\ldots,k\}$, and if there exists an index $j \in \{1,\ldots,k\}$ such that $f_j(x) < f_j(y)$. A point $x \in R$ is called optimal, or *Pareto* optimal, with respect to (1), if there is no other point $y \in R$ that dominates $x$. The set of all optimal solutions is called the Pareto set, and the set of images of the optimal solutions is called the the Pareto front.

Recently, a numerical method has been proposed for differentiable MOPs that allows to steer the search from a given point into a desired direction $d \in \mathbb{R}^k$ in objective space ([21]). To be more precise, given a point $x \in \mathbb{R}^n$, a search direction $\nu \in \mathbb{R}^n$ is sought such that

$$\lim_{t \searrow 0} \frac{f_i(x_0 + t\nu) - f_i(x_0)}{t} = d_i, \quad i = 1, \dots, k. \tag{2}$$

Such a direction vector $\nu$ solves the following system of linear equations:

$$J(x_0)\nu = d, \tag{3}$$

where $J(x)$ denotes the Jacobian of $F$ at $x$. Since typically $k << n$, we can assume that the system in Equation (3) is (highly) underdetermined. Among the solutions of Equation (3), the one with the smaller 2-norm can be viewed as the greedy direction for the given context. This solution is given by

$$\nu_+ := J(x)^+ d, \tag{4}$$

where $J(x)^+$ denotes the pseudo inverse of $J(x)$ (we refer e.g. to [17] for an efficient computation of $\nu_+$). If one proceeds the search in direction $d$ in the same manner, this is identical to the numerical solution of the following initial value problem (starting from solution $x_0 \in \mathbb{R}^n$):

$$\begin{aligned} x(0) &= x_0 \in \mathbb{R}^n \\ \dot{x}(t) &= \nu_+(x(t)), \quad t > 0 \end{aligned} \tag{5}$$

If $d$ is a 'descent direction' (i.e., $d_i \leq 0$ for all $i = 1, \dots, k$ and there exists an index $j$ such that $d_j < 0$), a numerical solution of (5) can be viewed as a particular hill climber for MOPs.

The endpoint $x^*$ of the solution curve of (5) does not necessarily have to be a Pareto point, but it is a boundary point in objective space, i.e., $F(x^*) \in \partial F(\mathbb{R}^n)$ which means that the gradients of the objectives in $x^*$ are linear independent (and hence, that $rank(J(x^*)) < k$). This fact can be used to check numerically if a current iterate is near to a boundary point: For the condition number of the Jacobian it holds

$$\kappa_2(J(x)) = \sqrt{\frac{\lambda_{max}(J(x)^T J(x))}{\lambda_{min}(J(x)^T J(x))}} \to \infty \quad \text{for} \quad x \to x^*, \tag{6}$$

where $\lambda_{max}(A)$ and $\lambda_{min}(A)$ denote the largest and the smallest eigenvalue of matrix $A$, respectively. (Roughly speaking, the condition number indicates how 'near' the rows of $J(x)$, i.e., the gradients of the objectives, are to be linearly independent: the higher the value of $\kappa_2(J(x))$, the closer $J(x)$ is to a matrix with rank less than $k$.) Further, one can check the (approximated) endpoint $x^*$ numerically for optimality by checking if $\| \sum_{i=1}^{k} \tilde{\alpha}_i \nabla f_i(x^*) \|_2 \leq tol$, where $tol > 0$ is a given tolerance and $\tilde{\alpha}$ solves the following $k$-dimensional quadratic optimization problem (see [18]):

$$\min_{\alpha}\left\{\left\|\sum_{i=1}^{k}\alpha_i\nabla f_i(x)\right\|_2^2 : \alpha_i \geq 0,\ i=1,\dots,k,\ \sum_{i=1}^{k}\alpha_i = 1\right\} \tag{7}$$

The hill climber described above shares many characteristics with the one described in [3], where also possible choices for $d$ are discussed.

## 3 Gradient Free Directed Search

The key of the DS is to solve Equation (3) in order to find a vector $\nu$ such that the search can be steered in $d$-direction. For this, the most expensive part might be the computation or approximation of the objectives' gradients. Here, we suggest an alternative way to compute such search directions $\nu$ using a finite difference method tailored to the given context. We note that this approach is not equal to the classical finite difference approach used to approximate the gradient (e.g., [17]).

Assume we are given a candidate solution $x \in \mathbb{R}^n$ and $r$ search directions $\nu_i \in \mathbb{R}^n$, $i = 1,\dots,r$. Define the matrix $\mathscr{F}(x) \in \mathbb{R}^{k\times r}$ as follows:

$$\mathscr{F}(x) := (\langle \nabla f_i(x), \nu_j\rangle)_{\ i=1,\dots,k;\ \ j=1,\dots,r}. \tag{8}$$

That is, every entry $m_{ij}$ of $\mathscr{F}$ is defined by the directional derivative of objective $f_i$ in direction $\nu_j$, $m_{ij} = \nabla_{\nu_j} f_i(x)$. Crucial for the subsequent discussion is the following result:

**Proposition 1** *Let $x, \nu_i$, $i = 1,\dots,r \in \mathbb{R}^n$, $\lambda \in \mathbb{R}^r$, and $\nu := \sum_{i=1}^{r}\lambda_i\nu_i$. Then*

$$J(x)\nu = \mathscr{F}(x)\lambda \tag{9}$$

*Proof.* It is

$$\mathscr{F}(x)\lambda = \begin{pmatrix} \langle\nabla f_1(x),\nu_1\rangle & \dots & \langle\nabla f_1(x),\nu_r\rangle \\ \vdots & \vdots & \vdots \\ \langle\nabla f_k(x),\nu_1\rangle & \dots & \langle\nabla f_k(x),\nu_r\rangle \end{pmatrix}\begin{pmatrix}\lambda_1\\ \vdots\\ \lambda_r\end{pmatrix} \tag{10}$$

and

$$J(x)\nu = J(x)(\sum_{i=1}^{r}\lambda_i\nu_i) = \sum_{i=1}^{r}\lambda_i\begin{pmatrix}\nabla f_1(x)^T\\ \vdots\\ \nabla f_k(x)^T\end{pmatrix}\nu_i \tag{11}$$

Hence, for the $l$-th component of both products it holds

$$(\mathscr{F}(x)\lambda)_l = \sum_{i=1}^{r}\lambda_i\langle\nabla f_l(x),\nu_i\rangle = (J(x)\nu)_l, \tag{12}$$

and the desired identity follows. $\square$

Hence, in search for a direction $\nu$, one can instead of Equation (3) try to solve the following equation:

$$\mathscr{F}(x)\lambda = d, \tag{13}$$

and set

$$\nu := \sum_{i=1}^{r} \lambda_i \nu_i. \tag{14}$$

**Remark 1** *Assume that we are given a candidate solution $x_0 \in \mathbb{R}^n$ and further $r$ points $x_i$, $i = 1,\ldots,r$, in the neighborhood of $x_0$ together with their function values $F(x_i)$, $i = 0,\ldots,r$. Defining*

$$\nu_j := \frac{x_j - x_0}{\|x_j - x_0\|_2}, \quad t_j := \|x_j - x_0\|_2, \quad j = 1,\ldots,r, \tag{15}$$

*one can approximate the entries of $\mathscr{F}$ by finite differences as follows:*

$$\begin{aligned} m_{ij} = \langle \nabla f_i(x_0), \nu_j \rangle &= \lim_{t \searrow 0} \frac{f_i(x_0 + t\nu_j) - f_i(x_0)}{t} \\ &\approx \frac{f_i(x_j) - f_i(x_0)}{\|x_j - x_0\|_2}, \quad i = 1,\ldots,k,\ j = 1,\ldots,r. \end{aligned} \tag{16}$$

*Analog to the well-known forward differences to approximate the gradient, one can show that the computational error is given by*

$$\langle \nabla f_i(x_0), \nu_j \rangle = \frac{f_i(x_j) - f_i(x_0)}{\|x_j - x_0\|_2} + O(\|x_j - x_0\|_2). \tag{17}$$

*Note that, by this, the search direction can be computed without any additional function evaluations.*

Since it is ad hoc not clear if Equation (13) has a solution, and even if it is solvable, how the condition of the problem is (in terms of $\kappa_2(\mathscr{F})$), we have to investigate the choice of $r$ and the $\nu_i$'s. For this, it is advantageous to write $\mathscr{F}(x)$ as follows:

$$\mathscr{F}(x) = J(x)V, \tag{18}$$

where $V := (\nu_1,\ldots,\nu_r) \in \mathbb{R}^{n\times r}$ is the matrix consisting of the search directions $\nu_i$. If $rank(J(x)) = k$ (which is given for a non-boundary point $x$), it is known from linear algebra that

$$rank(J(x)) = k \quad \Rightarrow \quad rank(\mathscr{F}(x)) = rank(V). \tag{19}$$

If on the other hand $x$ is a boundary point (and hence, $rank(J(x)) < k$), then it follows by the rank theorem of matrix multiplication that also $rank(\mathscr{F}(x)) < k$ regardless of the choice of $V$ (i.e., regardless of the number $r$ and the choice of the search directions $\nu_i$.).

This indicates that the condition number of $\mathscr{F}(x)$ can be used to check numerically if a current iterate is already near to an endpoint of (5). Equation (19) indicates that the $\nu_i$'s should be chosen such that they are linearly independent. If in addition the search directions are orthogonal to each other, a straightforward calculation shows that

$$V \text{ orthogonal} \quad \Rightarrow \quad \kappa_2(\mathscr{F}(x)) = \kappa_2(J(x)). \tag{20}$$

In that case, the condition number $\kappa_2(\mathscr{F}(x))$ can indeed be used as a stopping criterion, analog to the original method described in Section 2. That is, one can stop the iteration if for a current iterate $x_i$ it holds

$$\kappa_2(\mathscr{F}(x_i)) > tol_\kappa, \tag{21}$$

where $tol_\kappa >> 1$ is a large number.

**Example 1** *Consider the following bi-objective model ([12]):*

$$\begin{aligned} F &: \mathbb{R}^n \to \mathbb{R}^2 \\ f_i(x) &= \|x - a_i\|_2^2, \quad i = 1,2, \end{aligned} \tag{22}$$

*where $a_1 = (1,\ldots,1)^T, a_2 = (-1,\ldots,-1)^T \in \mathbb{R}^n$. The Pareto set is given by the line segment between $a_1$ and $a_2$, i.e.,*

$$\mathscr{P} = \{x \in \mathbb{R}^n \; : \; x_i = 2\alpha - 1, i = 1,\ldots,k, \alpha \in [0,1]\} \tag{23}$$

*Let $r = 2$ and $\nu_1 := e_i$ and $\nu_2 := e_j$, $i \neq j$, where $e_i$ denotes the i-th canonical vector. Then, it is*

$$\mathscr{F}(x) = \begin{pmatrix} x_i - 1 & x_j - 1 \\ x_i + 1 & x_j + 1 \end{pmatrix} \tag{24}$$

*It is* $\det(\mathscr{F}(x)) = 1/(2(x_i - x_j))$, *and hence,*

$$\det(\mathscr{F}(x)) = 0 \quad \Leftrightarrow \quad x_i = x_j, \tag{25}$$

*by which it follows that it is rank$(\mathscr{F}(x)) = 2$ for all $x \in \mathbb{R}^n \backslash B$, where $B := \{x \in \mathbb{R}^n \; : \; x_i = x_j\}$ (note that $\mathscr{P} \subset B$). Since $B$ is a zero set in $\mathbb{R}^n$, the probability is one that for a randomly chosen point $x \in \mathbb{R}^n$ the matrix $\mathscr{F}(x)$ has full rank, and hence, that Equation (13) has a unique solution. To be more precise, it is $\nu = \lambda_1 e_i + \lambda_2 e_j$, where*

$$\begin{aligned} \lambda = \mathscr{F}^{-1}(x)d &= \frac{1}{det(\mathscr{F}(x))} \begin{pmatrix} x_{j+1} & -x_j + 1 \\ -x_i - 1 & x_j - 1 \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} \\ &= \frac{1}{2(x_i - x_j)} \begin{pmatrix} x_j(d_1 - d_2) + d_1 + d_2 \\ x_i(d_2 - d_1) - d_1 - d_2 \end{pmatrix}. \end{aligned} \tag{26}$$

*Note that this holds regardless of the number $n \geq 2$ of the parameter dimension.*

The above considerations show that already for $r = k$ search directions $v_i$, $i = 1, \ldots, r$, one can find a descent direction $\tilde{v}$ by solving Equation (13). However, by construction it is $v \in span\{v_1, \ldots, v_k\}$ which means that only a $k$-dimensional subspace of the $\mathbb{R}^n$ is explored in one step. One would expect that the more search directions $v_i$ are taken into account, the better the choice of $\tilde{v}$ is. This is indeed the case: For $r > k$, we suggest to choose analog to (4)

$$v_+^{(r)} := \sum_{i=1}^{r} \lambda_i v_i, \quad \text{where} \quad \lambda = \mathscr{F}(x_0)^+ d \tag{27}$$

The following discussion gives a relation between $v_+^{(r)}$ and $v_+$ for non-boundary points $x$ for the case that the $v_i$'s are orthonormal: It is

$$v_+ = J^+(x)d = J(x)^T(J(x)J(x)^T)^{-1}d \tag{28}$$

and

$$\begin{aligned}\lambda = \mathscr{F}(x)^+ d &= V^T J(x)^T (J(x)\underbrace{VV^T}_{I} J(x)^T)^{-1} d \\ &= V^T \underbrace{J(x)^T (J(x)J(x)^T)^{-1} d}_{v_+} = V^T v_+\end{aligned} \tag{29}$$

and hence

$$v_+^{(r)} = \sum_{i=1}^{r} \lambda_i v_i = \sum_{i=1}^{r} \langle v_i, v_+ \rangle v_i \tag{30}$$

For instance, when choosing $v_i = e_{j_i}$, Equation (30) gets simplified:

$$v_+^{(r)} = \sum_{i=1}^{r} v_{+,j_i} e_{j_i}, \tag{31}$$

i.e., $v_+^{(r)}$ has only $r$ entries which are identical to the corresponding entries of $v_+$. In both cases $v_+^{(r)}$ gets closer to $v_+$ with increasing number $r$ and for $r = n$ it is $v_+^{(r)} = v_+$.

**Remark 2** *We would like to stress that this approach is intended for* multi-*objective optimization problems (i.e., $k > 1$). For the special (and important) case of scalar optimization (i.e., $k = 1$), the present approach is of very limited value as the following discussion shows: For $r = k = 1$, Equation (13) reads as*

$$\langle \nabla f(x), v_1 \rangle \lambda = d \tag{32}$$

*Concrete values for the desired direction $d$ in image space are hard to find. If it is e.g. desired to obtain improvements of the objective $f$, one may choose (after normalization) $d = -1$. Assuming that $\langle \nabla f(x), v_1 \rangle \neq 0$, Equation (32) leads then (again after normalization) to*

$$\lambda = \begin{cases} -1 & \textit{if } \langle \nabla f(x), v_1 \rangle > 0 \\ 1 & \textit{if } \langle \nabla f(x), v_1 \rangle < 0 \end{cases} \tag{33}$$

*and thus to $v \in \{v_1, -v_1\}$. However, this does not bring any new insight: It is well-known that the descent cone of $f$ at $x$ is given by*

$$C(x) := \{v \in \mathbb{R}^n \ : \ \langle \nabla f(x), v \rangle < 0\}, \tag{34}$$

*and hence, it is under the above assumption on $v_1$ either $v_1 \in C(x)$ or $-v_1 \in C(x)$.*

Finally, we state the Discrete Directed Search (DDS) which is simply a line search along the search direction $v_+^{(r)}$ (see Algorithm 1). For an efficient step size control we refer to [15].

**Algorithm 1** Discrete Directed Search (DDS)

```
Require: Initial solutions $x_0, x_1, \ldots x_r \in \mathbb{R}^n$
Ensure: New candidate solution $x_{new}$
1: compute $v_+^{(r)}$ as in Eq. (27).
2: compute $t \in \mathbb{R}_+$
3: $x_{new} := x_0 + t v_+^{(r)}$
```

## 4 Integration of DDS into MOEA/D

Here we show the potential of the DDS as local search engine within the state-of-the-art method MOEA/D [24]. The philosophy behind this MOEA consists of employing a decomposition approach, to convert the problem of approximating the Pareto front into a certain number of scalar optimization problems (SOPs). We stress that MOEA/D is indeed particularly attractive to be combined with the DDS procedure. Two important reasons for this are (a) MOEA/D has an implicit neighborhood structure, imposed by the particular decomposed problems, and (b) there is a weight vector associated to each subproblem, and to each individual.

In this sense, DDS can take advantage of (a) to avoid the computation of the neighbors used to estimate the search direction $v_+^{(r)}$; also, no extra function evaluations are necessary, which makes the computation of the search direction an effortless procedure—in terms of function evaluations. In other words, given a point $x \in \mathbb{R}^n$, if some neighbors of $x_1, \ldots, x_r$ are already evaluated, the computation can be done without any additional function evaluations. In general, memetic MOEAs which use gradient-based information have already proven their efficacy on several MOPs [14, 3], but the cost of estimating the first order information has been always an issue.

From a practical point of view, the reason (b) allows us to automatically identify which is the individual with the best fitness associated with each subproblem; and establishing, in this manner, a relationship with the corresponding weight vector for the movement performed by DDS. Furthermore, DDS also takes advantage that MOEA/D already has a computed reference point for the decomposed problems. In this sense, once the individual $p$ is chosen to be affected by the local search, the values for $d_i$ in the DDS are already set for $p$ (according with its corresponding decomposed MOEA/D subproblem).

How to apply the local search is one of the main issues when designing memetic algorithms. Two important parameters have been identified [8, 13] as crucial when controlling the local search application on memetic MOEAs. They are:

(i) The frequency $k_{ls}$ for application of the local search along the total amount of generations.
(ii) The number of elements $h_{ls}$, from the population, to which the local search is applied each generation.

Algorithm 2 describes the coupling of DDS and MOEA/D. The notation regarding MOEA/D procedures and parameters is consistent with the one presented in [24]. The SOP regarding the decomposition was, in this case, taken by the Tchebycheff approach as:

$$\text{minimize } g^{te}(x|\lambda, z^*) = \max_{1\leq i\leq m} \{\lambda_i |f_i(x) - z_i^*\} \tag{35}$$

where $z^*$, such that $z_i^* = min\{f_i(x)|x \in P_0\}$, is the reference point; and the direction $d_i$ for the application of DDS to the individual $x_i$ is set as $d_i = \lambda_i - z^*$.

**Algorithm 2** MOEA/D/DDS

```
 1: Set the weight vectors $\lambda_i$ and the neighborhoods $B(i) = \{i_1,\dots,i_T\}$ for each
    decomposed problem ($\lambda_{i_1},\dots,\lambda_{i_T}$ are the $T$ closest weight vectors to $\lambda_i$).
 2: Initialize an initial population $P_0 = \{x_1,\dots x_N\}$.
 3: Initialize the reference point $z^*$, $EP = \emptyset$, $gen = 1$.
 4: repeat
 5:     for $i = 1,\dots,N$ do
 6:         Select two indexes $k,l$ from $B(i)$ and generate, using genetic operators, a new
            solution $y$ from $x_k$ and $x_l$.
 7:         Apply the subproblem improvement heuristic for each $y$ in order to get $y'$
            (Eq. 35).
 8:         if $mod(gen,k_{ls}) == 0$ and $mod(i,h_{ls}) == 0$ then
 9:             Apply DDS to $y'$, in order to get $y''$.
10:             Set $y' \leftarrow y''$.
11:         end if
12:         Update the reference point $z^*$.
13:         Remove from $EP$ all the vectors dominated by $y'$ and add it if no vectors in $EP$
            dominate $y'$.
14:     end for
15:     $gen = gen + 1$.
16: until Stopping criteria is satisfied
17: report $EP$.
```

## 5 Numerical Results

In this section we show some results of the MOEA/D/DDS for the computation of Pareto fronts as well as of the DS in the context of a particular control problem.

### *5.1 Comparison MOEA/D and MOEA/D/DDS*

Since we have chosen MOEA/D as base MOEA, it seems reasonable to test over the CEC09 benchmark [27]. For this, we adapted the available code from a specific version of MOEA/D [25], which was tested for performance with remarkable results over this particular test suite [26]. Differences of this code and the MOEA/D original version are that this modification allows the computational effort to be distributed among the subproblems based on an utility function $\pi_i$ defined for each subproblem.

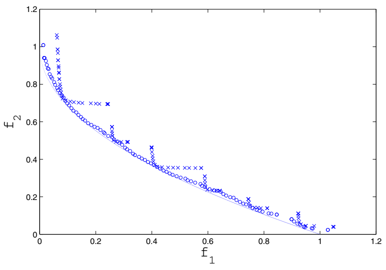
The main parameters for MOEA/D were set according to Table 1, and for the DDS we have chosen $r = 5$. We stop the computations after 30,000 function evaluations, which represents the 10% of the budget originally allowed by the competition. Figure 1 presents plots that show that the Pareto front has been reached, by the MOEA/D/DDS using this reduced budget. Finally, the parameters related to the control for application of the local search are presented in Table 2. As performance indicators to compare the results of the different algorithms we have chosen to take the Generational Distance (GD, see [23]), the Inverted Generational Distance (IGD, see [5]), the averaged Hausdorff Distance $\Delta_1$ (see [19, 20]) which is in fact the maximum of the GD and the IGD value, and the Hypervolume indicator (HV, see [28]). From Figure 1 and Table 3 it becomes clear that the new hybrid is outperforming its base MOEA in three out of four cases. For UF2, the indicator values of MOEA/D are slightly better, however, there is no clear winner.

**Table 1** Parameters setting for MOEA/D in this experiments

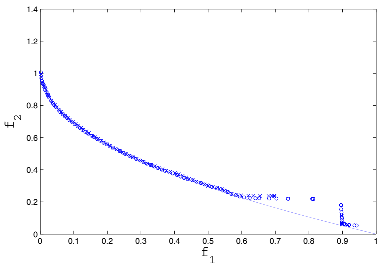
| Identifier | Value | Description |
|---|---|---|
| N | 600 | The number of subproblems considered |
| T | 0.1 N | Size of the neighborhood |
| $P_m$ | $1/n$ | Mutation rate |
| $EP$ | 100 | Number of final solutions (external population) |

**Table 2** Parameters setting for the memetic part

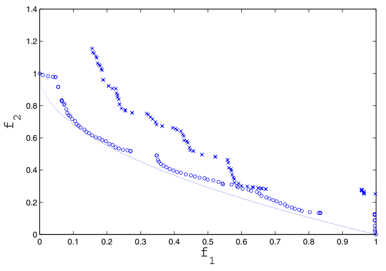
| Identifier | Value | Description |
|---|---|---|
| $k_{ls}$ | (0.15) tg | Local search application frequency; tg is the total number of generations. |
| $h_{ls}$ | (0.1) N | Percentage of the population over which the local search is applied. |



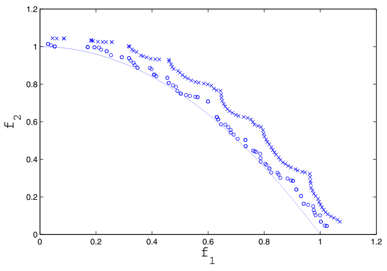
(a) Result UF1



(b) Result UF2



(c) Result UF3



(d) Result UF4

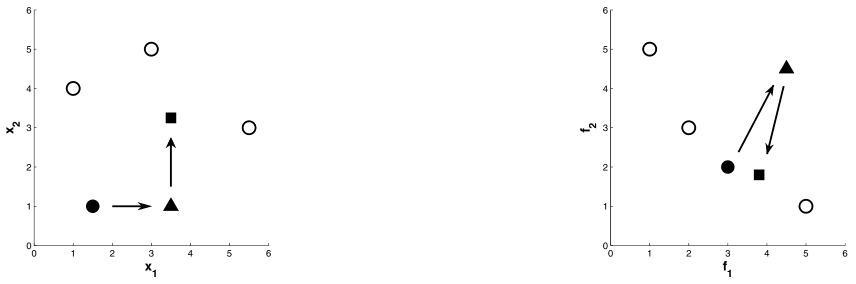
**Fig. 1** Numerical results for MOEA/D (crosses) and MOEA/D/DDS (circles) on the benchmark models UF1 to UF4 (compare also to Table 3). The true Pareto fronts are indicated by the dotted lines.

**Table 3** Indicator values obtained by MOEA/D and MOEA/D/DDS on the benchmark models UF1 to UF4. The budget for the function evaluations was set to 30,000. The information was gathered by 10 independent runs.

| | **Indicators** | | | | |
|---|---|---|---|---|---|
| **Problems** | | *GD* | *IGD* | $\Delta_1$ | HV |
| UF1 | MOEA/D | 0.0696046570 | 0.0709107497 | 0.0736649773 | 0.9431136545 |
| | MOEA/D/DDS | **0.0415884479** | **0.0400041235** | **0.0422788829** | **0.9619027643** |
| UF2 | MOEA/D | **0.0261457333** | **0.0195839746** | **0.0261457333** | **0.9757865917** |
| | MOEA/D/DDS | 0.0323564751 | 0.0151025484 | 0.0323564751 | 0.9645998451 |
| UF3 | MOEA/D | 0.1459679411 | 0.1307335754 | 0.1520909424 | 0.8604675543 |
| | MOEA/D/DDS | **0.0552610854** | **0.0537723289** | **0.0616385221** | **0.9608503276** |
| UF4 | MOEA/D | 0.0823081769 | 0.0871159952 | 0.0871159952 | 0.9206159284 |
| | MOEA/D/DDS | **0.0472797997** | **0.0472797997** | **0.0478409975** | **0.9498433991** |

## 5.2 A Control Problem

The skill of the DS is to steer the search into any direction in objective space. This can be used for Pareto front computations as seen above, however, may also have other applications as the following discussion shows. In [1], robustness of optimal solutions to MOPs subject to physical deterioration has been addressed. The problem posed in that study involves the need to steer the decorated performances (due to undesired changes in some design parameters) as close as possible to the original performances. In order to elucidate this demand for robustness, consider a two parameter bi-objective design space (i.e., $n = k = 2$). Assume Figure 2 shows four optimal solutions, and that the performance vector designated by the bold circle is the decision makers selected solution (denote by $x^*$). Now suppose that due to wear, one of the design parameters associated with that solution, changes (say $x_1$). This will cause the performances to deteriorate (see the triangle in both panels). Now suppose that there is a way to actively change the remaining parameter $x_2$ by actively controlling its value. If this is done properly, the performances might be improved to new performance vectors (designated in the figures by squares). The way the deteriorated performances are steered (controlled) as close as possible to the original location, has been termed in [1] as '*control in objective space*'. In [1] the control



(a) Parameter space (b) Image space

**Fig. 2** Hypothetical two dimensional bi-objective design problem

problem has been defined as a regulative control problem, and a proportional controller has been used to update the optimal solution in time. Note, however, that since optimality is defined in objective space, the DS (or DDS) can be used to accomplish this task: The direction $d_i$ is simply the difference of performance of the desired solution and the performance of the actual performance at the (deteriorated) point $x_i$, i.e., $d_i = F(x^*) - F(x_i)$.

To illustrate the performance of the suggested control scheme, we choose the quadratic bi-objective problem

$$\begin{aligned} F &: \mathbb{R}^{15} \to \mathbb{R}^2 \\ F(x) &= (\|x-a_1\|_2^2, \|x-a_2\|_2^2), \end{aligned} \tag{36}$$

where $a_1 = (1,\dots,1)$ and $a_2 = (-1,\dots,-1)$. For the decision space we choose $n = 15$ whereof three parameters deteriorate ($x_1, x_2$, and $x_3$) and the rest can be controlled. The results for the handling of the DDS controller with the above problem are depicted in Figure 3 (a). The solid line represents the initial Pareto front. The circle is the performance of the initial design. The performance after deterioration occurs is marked with triangles. Since some design variables have deteriorated, the Pareto front has changes in time. The deteriorated Pareto front in every time step is marked with a dashed line. The final state of the DDS controlled performance is marked with a black square, while the trajectory is marked with smaller gray squares. Note that the trajectory is going along the Pareto optimal front, and stops when the error is minimal. Figure 3 (b) depicts the performance of the deteriorated product with and without the DDS controller. The uncontrolled performance is described with triangles and the controlled one with squares.

We note that this result has been obtained by using the classical DS, however, from this we conclude that the DDS might be an alternative choice for models where no gradient information is at hand. We leave this for future research.

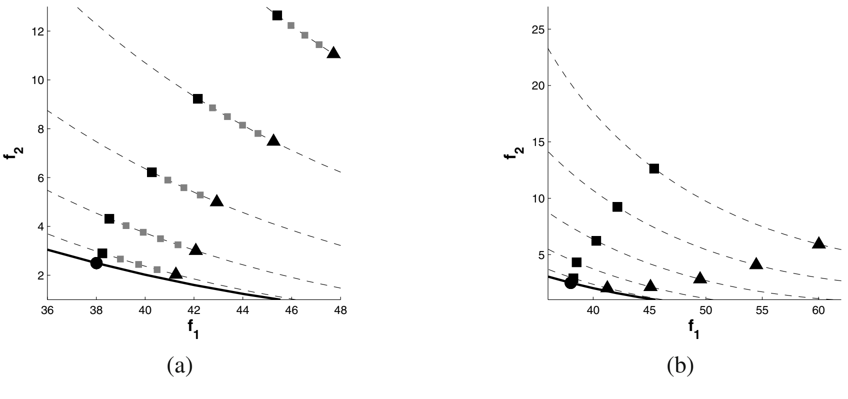


**Fig. 3** Result of the DS approach to the control problem

## 6 Conclusions

In this paper, we have modified the Directed Search Method, a point-wise iterative search procedure that allows to steer the search into any direction given in objective space of a given MOP. The resulting algorithm, DDS, allows to perform similar iterations as its original, however, without using gradient information but by exploiting the neighborhood information in order to find a suitable search direction. The latter makes the new algorithm in particular interesting for set oriented search procedures such as MOEAs. Here, we have made a first attempt to demonstrate this by integrating the DDS into MOEA/D. Comparisons on some benchmark functions have shown the benefit of such a hybridization.

For future work, the development of more efficient memetic strategies as the one proposed in this paper is an interesting topic which will call for a more sophisticated interplay of local and global search. Also, the adaption of the DDS to higher dimensional problems seems to be very interesting. Note that the choice of the number of test points near a solution $x_0$ that have to be chosen in order to find a search direction merely depends on the number of objectives involved in the MOP, and not on the dimension of the parameter space. Finally, we intend to utilize the DS/DDS in other applications, e.g., in the context of changing market demands as described in [2].

**Acknowledgements.** This research was supported by a Marie Curie International Research Staff Exchange Scheme Fellowship within the 7th European Community Framework Program. The first author acknowledges support from IPN project no. 20121478, and the last author acknowledges support from CONACyT project no. 128554.

## References

1. Avigad, G., Eisenstadt, E.: Robustness of Multi-objective Optimal Solutions to Physical Deterioration through Active Control. In: Deb, K., Bhattacharya, A., Chakraborti, N., Chakroborty, P., Das, S., Dutta, J., Gupta, S.K., Jain, A., Aggarwal, V., Branke, J., Louis, S.J., Tan, K.C. (eds.) SEAL 2010. LNCS, vol. 6457, pp. 394–403. Springer, Heidelberg (2010)
2. Avigad, G., Eisenstadt, E., Schütze, O.: Handling changes of performance-requirements in multi objective problems. Journal of Engineering Design (to appear, 2012)
3. Bosman, P.A.N., de Jong, E.D.: Exploiting gradient information in numerical multi-objective evolutionary optimization. In: Beyer, H.-G., et al. (eds.) 2005 Genetic and Evolutionary Computation Conference (GECCO 2005), vol. 1, pp. 755–762. ACM Press, New York (2005)
4. Brown, M., Smith, R.E.: Directed multi-objective optimisation. International Journal of Computers, Systems and Signals 6(1), 3–17 (2005)
5. Coello Coello, C.A., Cruz Cortés, N.: Solving Multiobjective Optimization Problems using an Artificial Immune System. Genetic Programming and Evolvable Machines 6(2), 163–190 (2005)
6. Coello Coello, C.A., Lamont, G.B., Van Veldhuizen, D.A.: Evolutionary Algorithms for Solving Multi-Objective Problems, 2nd edn. Springer, New York (2007)
7. Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons, Chichester (2001) ISBN 0-471-87339-X

8. Ishibuchi, T.Y.H., Murata, T.: Balance between Genetic Search and Local Search in Hybrid Evolutionary Multi-Criterion Optimization Algorithms. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2002), pp. 1301–1308. Morgan Kaufmann Publishers, San Francisco (2002)
9. Ishibuchi, H., Murata, T.: Multi-objective genetic local search algorithm. In: Proc. of 3rd IEEE Int. Conf. on Evolutionary Computation, Nagoya, Japan, pp. 119–124 (1996)
10. Jaszkiewicz, A.: Do multiple-objective metaheuristics deliver on their promises? a computational experiment on the set-covering problem. IEEE Transactions on Evolutionary Computation 7(2), 133–143 (2003)
11. J.: D Knowles and D.W Corne. M-PAES: a memetic algorithm for multiobjective optimization. In: Proceedings of the IEEE Congress on Evolutionary Computation, Piscataway, New Jersey, pp. 325–332 (2000)
12. Köppen, M., Yoshida, K.: Many-Objective Particle Swarm Optimization by Gradual Leader Selection. In: Beliczynski, B., Dzielinski, A., Iwanowski, M., Ribeiro, B. (eds.) ICANNGA 2007. LNCS, vol. 4431, pp. 323–331. Springer, Heidelberg (2007)
13. Lara, A., Coello Coello, C.A., Schütze, O.: A painless gradient-assisted multi-objective memetic mechanism for solving continuous bi-objective optimization problems. In: 2010 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8. IEEE, IEEE Press (2010)
14. Lara, A., Sanchez, G., Coello Coello, C.A., Schütze, O.: HCS: A new local search strategy for memetic multiobjective evolutionary algorithms. IEEE Transactions on Evolutionary Computation 14(1), 112–132 (2010)
15. Mejia, E., Schütze, O.: A predictor corrector method for the computation of boundary points of a multi-objective optimization problem. In: International Conference on Electrical Engineering, Computing Science and Automati Control (CCE 2010), pp. 1–6 (2007)
16. Miettinen, K.M.: Nonlinear Multiobjective Optimization. Springer (1999)
17. Nocedal, J., Wright, S.: Numerical Optimization. Springer Series in Operations Research and Financial Engineering. Springer (2006)
18. Schäffler, S., Schultz, R., Weinzierl, K.: A stochastic method for the solution of unconstrained vector optimization problems. Journal of Optimization Theory and Applications 114(1), 209–222 (2002)
19. Schuetze, O., Equivel, X., Lara, A., Coello Coello, C.A.: Some comments on GD and IGD and relations to the Hausdorff distance. In: GECCO 2010: Proceedings of the 12th Annual Conference Comp. on Genetic and Evolutionary Computation, pp. 1971–1974. ACM, New York (2010)
20. Schütze, O., Esquivel, X., Lara, A., Coello Coello, C.A.: Using the averaged Hausdorff distance as a performance measure in evolutionary multi-objective optimization. IEEE Transactions on Evolutionary Computation (2012), doi:10.1109/TEVC.2011.2161872
21. Schütze, O., Lara, A., Coello Coello, C.A.: The directed search method for unconstrained multi-objective optimization problems. In: Proceedings of the EVOLVE – A Bridge Between Probability, Set Oriented Numerics, and Evolutionary Computation (2011)
22. Vasile, M.: A behavior-based meta-heuristic for robust global trajectory optimization. In: IEEE Congress on Evolutionary Computing, vol. 2, pp. 494–497 (2007)
23. Van Veldhuizen, D.A.: Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations. PhD thesis, Department of Electrical and Computer Engineering. Graduate School of Engineering. Air Force Institute of Technology, Wright-Patterson AFB, Ohio (May 1999)
24. Zhang, Q., Li, H.: MOEA/D: A multi-objective evolutionary algorithm based on decomposition. multi-objective evolutionary algorithm based on decomposition 11(6), 712–731 (2007)

25. Zhang, Q., Liu, W., Li, H.: The performance of a new version of moea/d on cec09 unconstrained mop test instances. In: IEEE Congress on Evolutionary Computation, CEC 2009, pp. 203–208. IEEE (2009)
26. Zhang, Q., Suganthan, P.N.: Final report on CEC09 MOEA competition. In: Congress on Evolutionary Computation, CEC 2009 (2009)
27. Zhang, Q., Zhou, A., Zhao, S., Suganthan, P.N., Liu, W., Tiwari, S.: Multiobjective optimization test instances for the cec 2009 special session and competition. University of Essex, Technical Report (2008)
28. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE Transactions on Evolutionary Computation 3(4), 257–271 (1999)