# A New Memetic Strategy for the Numerical Treatment of Multi-Objective Optimization Problems

Oliver Schuetze
CINVESTAV-IPN
Computer Science Department
Mexico City 07300, MEXICO
schuetze@cs.cinvestav.mx

Gustavo Sanchez
Simon Bolivar University
Industrial Technology Department
Valle de Sartenejas, Edo. Miranda. VENEZUELA
gsanchez@usb.ve

Carlos A. Coello Coello
CINVESTAV-IPN
Computer Science Department
Mexico City 07300, MEXICO
ccoello@cs.cinvestav.mx

## ABSTRACT

In this paper we propose a novel iterative search procedure for multi-objective optimization problems. The iteration process – though derivative free – utilizes the geometry of the directional cones of such optimization problems, and is capable both of moving toward and along the (local) Pareto set depending on the distance of the current iterate toward this set. Next, we give one possible way of integrating this local search procedure into a given EMO algorithm resulting in a novel memetic strategy. Finally, we present some numerical results on some well-known benchmark problems indicating the strength of both the local search strategy as well as the new hybrid approach.

## Categories and Subject Descriptors

G.1.6 [**Numerical Analysis**]: Optimization

## General Terms

Algorithms, Performance

## Keywords

multi-objective optimization, memetic algorithm, hill climber, Pareto set

## 1. INTRODUCTION

In general, deterministic search methods present fast (local) convergence properties which surpass evolutionary multiobjective (EMO) algorithms properties. On the other hand, many EMOs accomplish the 'global task' exceedingly, that is, they tend to find very quickly a rough approximation of the entire solution set (the *Pareto set*), even for highly nonlinear models. Thus, to improve the overall performance of the search procedure, it can be advantageous to hybridize such two methods leading to a so-called *memetic* EMO (MEMO, see e.g., [18], [15], [4]).

One of the first MEMOs for models on discrete domains was presented in [14] as a 'Multi-Objective Genetic Local Search' (MOGLS) approach. In this algorithm, the authors propose to use the local search method after the usual variation operators are applied. Another historically important MEMO, called M-PAES, was proposed in [16]. Unlike the Ishibuchi approach, M-PAES is based on a pure local search algorithm (PAES), but adding a population with a crossover operator. Two archives are also used: one that maintains the global non-dominated solutions and the other that is used as the comparison set for the local search phase.

The continuous case – i.e., continuous objectives defined on a continuous domain – was first explored in [11], where a neighborhood search was applied to NSGA-II [7]. In their initial work, the authors applied the local search only after NSGA-II had ended. Later works compare this approach with the same local search method being applied after every generation. Evidently, they found that the added computational workload impacted efficiency.

In [13] a gradient-based local algorithm, sequential quadratic programming (SQP), was used in combination with NSGA-II and SPEA to solve the ZDT benchmark suite [26]. The authors stated that if there are no local Pareto fronts, the hybrid EMO has faster convergence toward the true Pareto front than the original one, either in terms of the objective evaluations or in terms of the CPU time consumed (recall that a gradient-based algorithm is utilized and the sole usage of the number of function calls as basis for a comparison can be misleading). Furthermore, they found that the hybridization technique does not decrease the solution diversity.

In this work, we present a novel local search operator, the *Hill Climber with Sidestep* (HCS). This iteration process aims to find a sequence of 'better' solutions (hill climber). If no better solution is found within a certain number of trials – which indicates that the actual iterate is already 'near' to a local solution – the process automatically tries to determine the next iterates *along* the Pareto set (sidestep). Additionally, we show one possible way to integrate the HCS into a given EMO algorithm, and thus, how to construct a novel MEMO. Unlike previous works (see [13] and [23]) the proposed operator does not require any gradient information, but such information can easily be integrated if provided by the model.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
*GECCO'08*, July 12–16, 2008, Atlanta, Georgia, USA.
Copyright 2008 ACM 978-1-60558-131-6/08/07 ...$5.00.

The remainder of this paper is organized as follows. In Section 2, we state some theoretical background. In Section 3, we introduce the HCS, a novel iterative procedure which is of local nature, and propose one possible way to integrate this algorithm into a global search procedure. In Section 4, we show some numerical results, and finally, some conclusions are drawn in Section 5.

## 2. BACKGROUND

In the following we consider continuous multi-objective optimization problems of the form

$$\min_{x \in Q} \{F(x)\}, \qquad \text{(MOP)}$$

where $Q \subset \mathbb{R}^n$ and the function $F$ is defined as the vector of the objective functions

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

and where each $f_i : Q \to \mathbb{R}$ is continuous.
The optimality of a point $x \in Q$ is based on the concept of *dominance* which dates back over a century and was proposed first by Pareto ([24]).

**Def 2.1** *(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 $i \in \{1, \ldots, k\}$. The relation $\leq_p$ 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 $F(x) \leq_p F(y)$ and $F(x) \neq F(y)$, 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*. This set typically – i.e., under mild regularity assumptions – forms a $(k-1)$-dimensional object. The image of the Pareto set is called the *Pareto front*.

## 3. THE MEMETIC STRATEGY

In the following we propose a novel iterative local search procedure, the HCS, and present further on one possible way to integrate this procedure into a given EMO algorithm in order to obtain a MEMO.

### 3.1 A Local Search Procedure: Hill-Climber with Sidestep (HCS)

Some requirements on such a process, which will also be addressed here, are as follows:

(a) The process should work with or without gradient information (whether or not provided by the model).

(b) If a given point $x_0 \in Q$ is 'far away' from the Pareto set, it is desirable that the next iterate $x_1$ is better (i.e., $x_1 \prec x_0$). If $x_0$ is already 'near' to the Pareto set, a search *along* the solution set is of particular interest.

(c) It is desirable that the search procedure can decide *automatically* between the two situations described in (b) and generate a candidate solution according to the given situation.

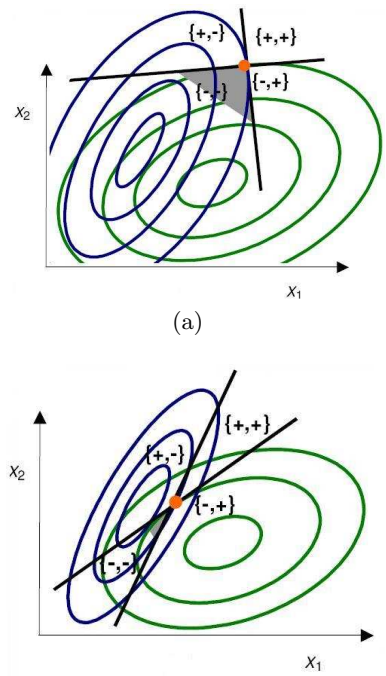


**Figure 1: The descent cone (shaded) for an MOP with 2 parameters and 2 objectives during initial (a) and final (b) stages of convergence. The descent cone shrinks to zero during the final stages of convergence. The figure is taken from [4].**

(d) The procedure should be capable of handling (inequality[1]) constraints.

In [3] a good insight into the structure of multi-objective problems is given by analyzing the geometry of the directional cones at different stages of the optimization process: when a point $x_0$ is 'far away' from any local Pareto optimal solutions, the gradients' objectives are typically aligned and the descent cone is almost equal to the half-spaces associated with each objective. Therefore, for a randomly chosen search direction $\nu$, there is a nearly 50 % chance that this direction is a descent direction (i.e., there exits an $h_0 \in \mathbb{R}_+$ such that $F(x_0 + h_0\nu) <_p F(x_0)$). If on the other side a point $x_0$ is 'close' to the Pareto set, the individual gradients are almost contradictory (compare also to the famous theorem of Kuhn and Tucker [17]), and thus the size of the descent cone is extremely narrow, resulting in a small probability for a randomly chosen vector to be a descent direction (see also Figure 1). The Hill-Climber with Sidestep (HCS) was constructed on the basis of these observations. In the following, the HCS algorithm is described for the case in which no gradient information is available, since that is more common for real-world engineering problems which is the main area of application for EMO algorithms. Further, we describe the

[1]Equality constraints, which are in general much more difficult to satisfy than inequality constraints, may be possible to relax by converting them into suitable inequality constraints with some loss of accuracy. See [5] for a thorough discussion.

unconstrained case. Possible modifications for both differentiable and constrained models are given below.
Given a point $x_0 \in Q$, the next iterate is selected as follows: a further point $x_1$ is chosen randomly from a neighborhood of $x_0$, say $\tilde{x}_1 \in B(x_0, r)$ with

$$B(x_0, r) := \{x \in \mathbb{R}^n : x_{0,i} - r_i \leq x_i \leq x_{0,i} + r_i \ \forall i = 1, .., n\}, \tag{1}$$

where $r \in \mathbb{R}^n_+$ is a given (problem depending) radius. If $\tilde{x}_1 \prec x_0$, then $\nu := \tilde{x}_1 - x_0$ is a descent direction[2] at $x_0$, and along it a 'better' candidate can be searched, for example via line search methods (see below for one possible realization). If $x_0 \prec \tilde{x}_1$ the same procedure can be applied to the opposite direction (i.e., $\nu := x_0 - \tilde{x}_1$) starting with $\tilde{x}_1$. If $x_0$ is 'far away' from any local solution, the chance is quite high that domination occurs (see above). If $x_0$ and $\tilde{x}_1$ are mutually non-dominating, the process will be repeated with further candidates $\tilde{x}_2, \tilde{x}_3, \ldots \in B(x_0, r)$. If only mutually nondominated solutions $(\tilde{x}_i, x_0)$ are found within $N_{nd}$ steps, this indicates, using the above observation, that the point $x_0$ is already near to the (local) Pareto set, and hence it is desirable to search along this set. For this, we suggest to use the accumulated information by taking the average search direction

$$\nu_{acc} = \frac{1}{N_{nd}} \sum_{i=1}^{N_{nd}} \frac{\tilde{x}_i - x_0}{\|\tilde{x}_i - x_0\|}, \tag{2}$$

since with this direction one expects the maximal diversity (or 'sidestep') among the available directions. This direction has previously been proposed as a local guide for a multi-objective particle swarm algorithm in [2]. Note that this is a heuristic that does *not* guarantee that $\nu_{acc}$ indeed points to a diversity cone. In fact, it can happen that this vector points to the descent or ascent cone, though the probability for this is low for points $x_0$ 'near' to a local solution due to the narrowness of these cones. However, in both cases Algorithm 1 acts like a classical hill climber (i.e., searches for better points) which is in the scope of the procedure. Alternatively, to prevent this and to be sure to search in a diversity direction, one can restrict (2) to points $\tilde{x}_i$ that lie in one common descent cone (for instance, for $k = 2$ the cone $\{+, -\}$ is the one where the values of objective $f_1$ are greater than at the given point $x_0$ while the values of $f_2$ are less, see Figure 1). However, note that there exist for $k$ objectives $2^k - 2$ such cones (for instance, for $k = 3$ the cones $\{-, -, +\}$, $\{-, +, -\}$, $\{+, -, -\}$, $\{-, +, +\}$, $\{+, -, +\}$, $\{+, +, -\}$), and it is ad hoc unclear which of them to favor in each situation. A pseudocode of the HCS is given in Algorithm 1, and in the following we go into detail for possible realizations. Table 1 summarizes the design parameters those values have to be chosen when realizing the HCS as it is proposed here.

***Computation of*** $t_k$**.** The situation is that we are given two points $x_0, x_1 \in \mathbb{R}^n$ such that $x_1 \prec x_0$. That is, there exists a subsequence $\{i_1, \ldots, i_l\} \subset \{1, \ldots, k\}$ with

$$f_{i_j}(x_1) < f_{i_j}(x_0), \quad j = 1, \ldots, l,$$

and thus, $\nu := x_1 - x_0$ is a descent direction for all $f_{i_j}$'s at the point $x_0$. For this case there exist various strategies

[2]In the sense that there exists a $\bar{t} \in \mathbb{R}_+$ such that $f_i(x_0 + \bar{t}\nu) < f_i(x_0)$, $i = 1, \ldots, k$, but not in the 'classical' sense, i.e., in case $f_i$ is differentiable $\nabla f_i(x_0)^T \nu < 0$ is not guaranteed.

**Algorithm 1** Multi-Objective Hill Climber with Sidestep

**Require:** starting point $x_0 \in Q$, radius $r \in \mathbb{R}^n_+$
**Ensure:** sequence $\{x_i\}_{i \in \mathbb{N}}$ of candidate solutions
$a := (0, \ldots, 0) \in \mathbb{R}^n$
$nondom := 0$
**for** $k = 1, 2, \ldots$ **do**
  set $x^1_k := x^b_{k-1}$ and choose $x^2_k \in B(x^1_k, r)$ at random
  **if** $x^1_k \prec x^2_k$ **then**
    $\nu_k := x^2_k - x^1_k$
    compute $t_k \in \mathbb{R}_+$ and set $\tilde{x}^n_k := x^2_k + t_k \nu_k$.
    choose $x^b_k \in \{\tilde{x}^b_k, x^1_k\}$ such that $f(x^b_k) = \min(f(\tilde{x}^n_k), f(x^1_k))$
    $nondom := 0, \quad a := (0, \ldots, 0)$
  **else if** $x^2_k \prec x^1_k$ **then**
    proceed analogous to case "$x^1_k \prec x^2_k$" with
    $\nu_k := x^1_k - x^2_k$ and $\tilde{x}^n_k := x^1_k + t_k \nu_k$.
  **else**
    $a := a + \frac{1}{N_{nd}} \frac{x^2_k - x^1_k}{\|x^2_k - x^1_k\|}$
    $nondom := nondom + 1$
    **if** $nondom = N_{nd}$ **then**
      compute $\tilde{t}_k \in \mathbb{R}_+$ and set $\tilde{x}^n_k := x^1_k + \tilde{t}_k a$.
      $nondom := 0, \quad a := (0, \ldots, 0)$
    **end if**
  **end if**
**end for**

to perform the line search (see e.g., [9]). We propose to proceed in analogy to [22], where a step size control for scalar optimization problems has been developed, as follows:
for $x_0, x_1$ and $f_{i_j}$, $j = 1, \ldots, l$ (for simplicity denoted by $f$) as above define

$$f_\nu : \mathbb{R} \to \mathbb{R}, \quad f_\nu(t) = f(x_0 + t\nu) \tag{3}$$

Choose $e \in (1, 2]$ (the same value for all $l$ cases) and compute $f_\nu(e)$. If $f_\nu(e) < f_\nu(1)$ then accept $t^*_{i_j}$ as step size for objective $f = f_{i_j}$. If the above condition does not hold we have collected enough information to approximate $f_\nu$ by a quadratic polynomial $p(t) = at^2 + bt + c$ with coefficients $a, b, c \in \mathbb{R}$. Using the interpolation conditions

$$p(0) = f_\nu(0), \quad p(1) = f_\nu(1), \quad p(e) = f_\nu(e), \tag{4}$$

we obtain all the coefficients of $p$. Since $p(1) < p(0)$ and $p(e) \geq p(1)$ and since $p$ is a quadratic polynomial the function contains exactly one minimum at

$$t^*_{i_j} = \frac{-b}{2a} = 2\frac{e^2(f_\nu(1) - f_\nu(0)) - f_\nu(e) + f_\nu(0)}{e(f_\nu(1) - f_\nu(0)) - f_\nu(e) + f_\nu(0)} \in (0, e). \tag{5}$$

Finally, the question arises how this information obtained by scalarization can be put together to select a step size strategy for the given multi-objective problem. The 'safest' step size control is certainly to take the smallest value of the $t^*_{i_j}$'s. In order not to get stuck due to small step sizes and to introduce a stochastic component into the search strategy we propose to choose a step size within the range which is given by the $t^*_{i_j}$'s, i.e.

$$x_{new} = x_0 + t^*_k \nu, \tag{6}$$

where $t^*_k \in [\min_{i=1,\ldots,l} t^*_{i_j}, \max_{i=1,\ldots,l} t^*_{i_j}]$ is taken at random.

*Computation of $\tilde{t}_k$.* We are given a point $x_0 \in \mathbb{R}^n$ and the search direction $a = \sum_{i=1}^{N_{nd}} (\tilde{x}_i - x_0)/\|\tilde{x}_i - x_0\|$ with $\tilde{x}_i \in B(x_0, r)$, $i = 1, \ldots, N_{nd}$, and such that $(x_0, \tilde{x}_i), i = 1, \ldots, N_{nd}$, are mutually nondominating. For this situation, we propose to proceed analogously to [21], where a step size strategy for multi-objective continuation methods is suggested: given a target value $\epsilon \in \mathbb{R}_+$ – e.g., the minimal value which makes two solutions distinguishable from a practical point of view –, the task is to compute a new candidate $x_{new} = x_0 + \tilde{t}a$ such that

$$\|F(x_0) - F(x_{new})\|_\infty \approx \epsilon \tag{7}$$

In case $F$ is Lipschitz continuous there exists an $L \geq 0$ such that

$$\|F(x) - F(y)\| \leq L\|x - y\|, \quad \forall x, y \in Q. \tag{8}$$

This constant can be estimated around $x_0$ by

$$L_{x_0} := \|DF(x_0)\|_\infty = \max_{i=1,\ldots,k} \|\nabla f_i(x_0)\|_1,$$

where $DF(x_0)$ denotes the Hessian of $F$ at $x_0$ and $\nabla f_i(x_0)$ the gradient of the $i$-th objective at $x_0$. In case $F$ is not differentiable the accumulated information can be used to compute the estimation

$$\tilde{L}_{x_0} := \max_{i=1,\ldots,N_{nd}} \frac{\|F(x_0) - F(\tilde{x}_i)\|_\infty}{\|x_0 - \tilde{x}_i\|_\infty}.$$

Combining (7), (8) and using the estimation $L_{x_0}$ leads to the step size control

$$x_{new} = x_0 + \frac{\epsilon}{L_{x_0}} \frac{a}{\|a\|_\infty}. \tag{9}$$

*Handling constraints.* In the course of the computation it can occur that iterates are generated which are not inside the feasible domain $Q$. That is, we are faced with the situation that $x_0 \in Q$ and $x_1 := x_0 + h_0\nu \notin Q$, where $\nu$ is the search direction. In that case we propose to proceed analogously to the well-known bisection method for root finding in order to track back from the current iterate $x_1$ to the feasible set:
let $in_0 := x_0 \in Q$ and $out_0 := x_1 \notin Q$ and $m_0 := in_0 + 0.5(out_0 - in_0) = x_0 + \frac{h_0}{2}\nu$. If $m_0 \in Q$ set $in_1 := m_0$, else $out_1 := m_0$. Proceeding in an analogous way, one obtains a sequence $\{in_i\}_{i \in \mathbb{N}}$ of feasible points which converges linearly to the boundary $\partial Q$ of the feasible set. One can, for example, stop this process with an $i_0 \in \mathbb{N}$ such that $\|out_{i_0} - in_{i_0}\|_\infty \leq tol$, obtaining a point $in_{i_0}$ with maximal distance $tol$ to $\partial Q$. See Algorithm 2 for one possible realization. Note that by this procedure no function evaluation has to be spent (in contrast, for instance, to penalization methods).

*Using gradient information.* In case the MOP is sufficiently smooth, some tools from mathematical programming can be integrated directly into the HCS in order to increase its performance (significantly). For instance, if all objectives' gradients are available descent directions can be computed directly ([10], [1]) for a given point $x \in Q$. This could e.g. replace the random search as proposed in line 4 of Algorithm 1.
Also a movement along the Pareto set can be performed much more efficiently by using *predictor-corrector* (PC) methods ([12], [20]). The direction $a$ in line 14 of Algorithm 1 can be viewed as a possible predictor for further efficient points along the solution manifold, and the hill climber plays the role of the correction mechanism. However, in case classical multi-objective PC methods are chosen the resulting better performance in terms of convergence does not come for free: in this case even the second derivatives of the objectives have to be available or approximated.

**Algorithm 2** Backtracking to Feasible Region

**Require:** $x_0 \in Q$, $x_1 = x_0 + h_0\nu \notin Q$, $tol \in \mathbb{R}_+$
**Ensure:** $\tilde{x} \in \overline{x_0 x_1} \cap Q$ with $\inf_{b \in \partial Q} \|b - \tilde{x}\| < tol$

```
in_0 := x_0
out_0 := x_1
i := 0
while ||out_i − in_i|| ≥ tol do
    m_i := in_i + (1/2)(out_i − in_i)
    if m_i ∈ Q then
        in_{i+1} := m_i
        out_{i+1} := out_i
    else
        in_{i+1} := in_i
        out_{i+1} := m_i
    end if
    i := i + 1
end while
return x̃ := in_i
```

**Table 1: Design parameters that are required for the realization of the HCS algorithm.**

| Parameter | Description |
|---|---|
| r | Radius for neighborhood search (Alg. 1) |
| $N_{nd}$ | Number of trials for the hill climber before the sidestep is performed (Alg. 1) |
| $\epsilon$ | Desired distance (in image space) for the sidestep (7) |
| tol | Tolerance value used for the backtracking in Alg. 2 |

## 3.2 Integration of HCS into the EMO Algorithm

Here we present one way to integrate the HCS into an EMO algorithm. There exist certainly several ways to do this. In this work we present a first attempt whereas the HCS acts like a typical mutation operator within a given EMO algorithm. The advantage of this approach is that by doing so the local search procedure can be integrated into any existing EMO algorithm with little effort.
For this particular study, we have decided to use SPEA2 [27] as the basis for our memetic algorithm, together with the SBX crossover operator [6] and a variable-wise mutation operator. A pseudocode can be found in Algorithm 3.

If the HCS is used as local search operator within a given (global) search strategy, it seems to be wise just to compute a few iterates which are used to build up the next population. For the computations in the subsequent section we have used the following strategy: if a point $x_0 \in \overline{P}_{k+1}$ is

**Algorithm 3** SPEA2 with Local Search

1: Generate initial population $P_0 \subset Q$ and set $A_0 := \emptyset$, $\bar{P}_0 := \emptyset$.
2: **for** $k = 0, 1, \ldots, N_{maxiter}$ **do**
3: $\overline{P}_{k+1} :=$ nondominated solutions of $P_k \cup A_k$
4: Set $A_{k+1} :=$ nondominated solutions of $\overline{P}_{k+1}$
5: Calculate fitness values of individuals in $\overline{P}_{k+1}$
6: Perform tournament selection in $\overline{P}_{k+1}$ to fill the mating pool
7: Apply crossover, mutation and the local search operators (HCS) to the mating pool.
8: Denote the resulting population by $P_{k+1}$.
9: **end for**

selected for local search, either

$$x_1 := x_0 + t_0 \nu_0 \tag{10}$$

is added to $P_{k+1}$ in case a dominated point could be found (lines 7 and 8 respectively line 12 of Algorithm 1), or

$$x_1 := x_0 + \tilde{t}_0 \nu_0 \quad \text{and} \quad x_2 := x_0 - \tilde{t}_0 \nu_0, \tag{11}$$

i.e., a search along the directions $\nu_0$ and $-\nu_0$, are both added to $P_{k+1}$ in case a sidestep is performed (line 17 of Alg. 1).

# 4. NUMERICAL RESULTS

Here we present some numerical results for the HCS as well as for a MEMO which results when the HCS is integrated into SPEA2 in order to demonstrate the strength of both the standalone algorithm and the memetic strategy.

## 4.1 Results of the HCS

Since the HCS as described in Algorithm 1 has no orientation in the search along the Pareto set, we have modified it for bi-objective models in the following way in order to demonstrate its potential:
the HSC is started as described in Algorithm 1. When the sidestep (line 17 of Algorithm 1) is performed $N_s$ times, this indicates that the current iteration is already near to the (local) Pareto set, and this vector is stored in $x_p$. In further sidesteps, candidates will only be accepted if the first objective gets decreased: if for a point $\tilde{x}_k^n := x_k^1 + \tilde{t}_k a$ it holds that $f_1(\tilde{x}_k^n) > f_1(x_k^1)$, the signum of $\tilde{t}_k$ is changed, and thus, $\tilde{x}_k^n := x_k^1 - \tilde{t}_k a$ is chosen as next iterate. If no improvements can be achieved according to $f_1$ within a given number $N_i$ of sidesteps, the HCS 'jumps' back to $x_p$, and a similar process is started but aiming for improvements according to $f_2$.
However, since this orientation is not needed within the use of an EMO algorithm because in that case only few iterates are beeing computed from a given starting point, these modifications are only done within this subsection.
In the following we will test the HCS on a convex model (i.e., a model which does not contain local minima where the local search can get stuck) and will investigate the unconstrained and the constrained case. Then we will consider a nonlinear and constrained model (ZDT1).
Consider the MOP CONVEX (see Table 2). The Pareto set of this model is located within $[-1, 1]^n$. First, we turn our attention to the unconstrained case: Figure 2 shows a result obtained by the modified HCS with dimension $n = 10$ and domain $Q = [-5, 5]^{10}$. Since the Pareto set is located within $Q$, no constraint handling techniques had to be applied in order to generate the sequence. Here, a total of

**Table 2: Objective functions used within this work with $\tilde{k} = n - k + 1$.**

| |
|---|
| **CONVEX** |
| $f_1(x) = (x_1 - 1)^4 + \sum_{i=2}^n (x_i - 1)^2$ |
| $f_2(x) = \sum_{i=1}^n (x_i + 1)^2$ |
| **ZDT1** |
| $f_1(x) = x_1$ |
| $f_2(x) = g(x)(1 - \sqrt{x_1/g(x)})$ |
| $g(x) = 1 + 9(\sum_{i=1}^n x_i)/(n-1)$ |
| $0 \le x_i \le 1$ |
| **DTLZ1** |
| $f_1(x) = \frac{1}{2} x_1 x_2 \ldots x_{k-1}(1 + g(x))$ |
| $f_2(x) = \frac{1}{2} x_1 x_2 \ldots (1 - x_{k-1})(1 + g(x))$ |
| $\vdots$ |
| $f_{k-1}(x) = \frac{1}{2} x_1 (1 - x_2)(1 + g(x))$ |
| $f_k(x) = \frac{1}{2}(1 - x_1)(1 + g(x))$ |
| $g(x) = 100 \left[ \tilde{k} + \sum_{i=k}^n (x_i - \frac{1}{2})^2 - \cos(20\pi(x_i - \frac{1}{2})) \right]$ |
| **DTLZ2** |
| $f_1(x) = \cos(\frac{x_1 \pi}{2}) \cos(\frac{x_2 \pi}{2}) \ldots \cos(\frac{x_{k-1} \pi}{2})(1 + g(x))$ |
| $f_2(x) = \cos(\frac{x_1 \pi}{2}) \cos(\frac{x_2 \pi}{2}) \ldots \sin(\frac{x_{k-1} \pi}{2})(1 + g(x))$ |
| $\vdots$ |
| $f_{k-1}(x) = \cos(\frac{x_1 \pi}{2}) \sin(\frac{x_2 \pi}{2})(1 + g(x))$ |
| $f_k(x) = \sin(\frac{x_1 \pi}{2})(1 + g(x))$ |
| $g(x) = 100 \left[ \tilde{k} + \sum_{i=k}^n (x_i - \frac{1}{2})^2 \right]$ |
| **DTLZ3** |
| $f_1(x) = \cos(\frac{x_1 \pi}{2}) \cos(\frac{x_2 \pi}{2}) \ldots \cos(\frac{x_{k-1} \pi}{2})(1 + g(x))$ |
| $f_2(x) = \cos(\frac{x_1 \pi}{2}) \cos(\frac{x_2 \pi}{2}) \ldots \sin(\frac{x_{k-1} \pi}{2})(1 + g(x))$ |
| $f_{k-1}(x) = \cos(\frac{x_1 \pi}{2}) \sin(\frac{x_2 \pi}{2})(1 + g(x))$ |
| $f_k(x) = \sin(\frac{x_1 \pi}{2})(1 + g(x))$ |
| $g(x) = 100 \left[ \tilde{k} + \sum_{i=k}^n (x_i - \frac{1}{2})^2 - \cos(\alpha\pi(x_i - \frac{1}{2})) \right]$ |
| $\alpha = 20$ |
| **DTLZ3*** |
| same as DTLZ3 but with $\alpha = 2$ |

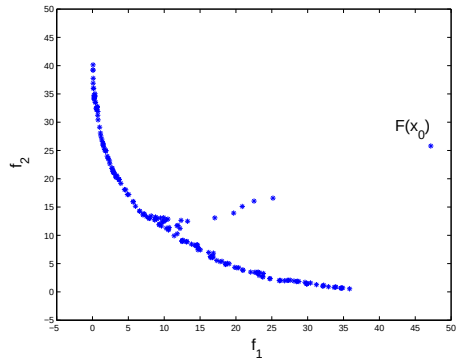


**Figure 2: Numerical result of HCS for MOP CONVEX with $Q = [-5, 5]^{10}$ in objective space (unconstrained case).**

2453 function calls had to be spent in order to get this result which actually represents a good approximation of the Pareto front.

Next, we consider the constrained case. For the result in Figure 3 we have chosen $n = 2$ and $Q = [0.5, 1.5] \times [1, 2]$ for the same problem, and thus – since the model is convex – the Pareto set is contained in the boundary of $Q$. The figures show that also in this case the HCS is capable of approaching the solution set, and moving along it further on. However, a total of 4211 function calls had to be spent in this setting, that is, much more than in the unconstrained case (note that the dimension of the model is much lower in the latter case).

Finally, we consider the problem ZDT1 (Table 2), which is a highly nonlinear model. Figure 4 shows a result in image space for two different initial solutions $x_0, z_0 \in Q = [0, 1]^{10}$. As anticipated, the results differ significantly since the HCS is a local strategy and ZDT1 contains many local Pareto fronts. However, the procedure is also in this case able to explore a part of the local Pareto front which is located near to the image of the initial solution.

## 4.2 Results of the EMO Algorithm

Here we make a comparison of the classical SPEA2 and the modification of this algorithm which is enhanced by the local iteration process (denoted by SPEA2HCS) in order to demonstrate the advantage of the memetic strategy. Table 3 displays the parameters which have been used for both SPEA2 and SPEA2HCS. Since at the beginning of the algorithm's execution, typically a global search is more effective than a local one, we have started the HCS after 75 % of the total number of generations (i.e., we have set $P_{LS} = 0$ in the first generations, see Table 3).

In order to evaluate the performance of the algorithms we have used the following three indicators (see [25] and [19]):

Generational Distance : $GD = \frac{1}{n}\sqrt{\sum_{i=1}^{n} \delta_i^2}$

Efficient Set Space : $ESS = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(d_i - \overline{d})^2}$

Maximal Distance : $MD = \max\limits_{\substack{i,j=1,\ldots,n \\ i \neq j}} d_{ij}$

Hereby, $\delta_i$ denotes the minimal Euclidean distance from

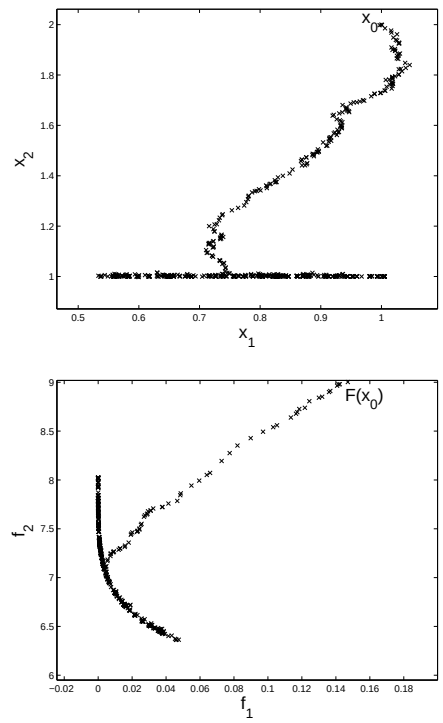


**Figure 3: Numerical result of HCS for MOP CONVEX with $Q = [0.5, 1.5] \times [1, 2]$ in objective space (constrained case).**

the image $F(x_i)$ of a solution $x_i, i = 1, \ldots, n$, to the true Pareto front, and

$$d_i := \min_{\substack{j=1,\ldots,n \\ i \neq j}} d_{ij} \quad \text{and} \quad \overline{d} := \frac{1}{n}\sum_{i=1}^{n} d_i, \tag{12}$$

where $d_{ij}$ is the Euclidean distance between $F(x_i)$ and $F(x_j)$. In the multi-objective optimization framework, there are in general three goals ([26]): (i) the distance of the resulting nondominated set to the Pareto-optimal front should be minimized, (ii) a uniform distribution of the solutions found is desirable, and (iii) the extent of the obtained nondominated front should be maximized. We have chosen the three indicators with the aim to measure the achievement of each of these goals, respectively.

Table 4 shows one comparison for some DTLZ test functions ([8], see also Table 2), where we have chosen $n = 7$ for the dimension of the parameter space and $k = 3$ objectives. The numerical results show that in almost all cases SPEA2HCS achieves better values than SPEA2 for all three indicators. For the generational distance, which is an indicator for convergence, the reduction is often near 50%.

The comparison of the results for DTLZ3 (with parameter $\alpha = 20$) and DTLZ3* (same as DTLZ3 but with $\alpha = 2$) gives some insight into the nature of memetic algorithms such as SPEA2HCS. The parameter $\alpha$ controls the number $N_l$ of strictly local Pareto fronts and is thus an indicator for the complexity of the model. $N_l$ is approximately equal to $\left(\frac{\alpha}{2}\right)^{n-k+1}$. That is, for $\alpha = 20$ there exist a total of 100.000 local Pareto fronts while for $\alpha = 2$ there exists merely one.

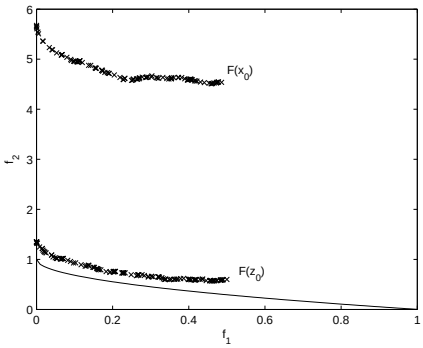


**Figure 4: Numerical result of HCS for MOP ZDT1 with $Q = [0,1]^{10}$ in objective space. Sequences for two initial solutions, $x_0$ and $z_0$, are shown as well as the true Pareto front.**

**Table 3: Parameters for SPEA2: $P_{size}$ and $A_{size}$ denote the population size and the maximal cardinality of the archive, and $P_{cross}, P_{mut}, P_{LS}$ denote the probabilities for crossover, mutation and line search respectively.**

| Parameter | Value |
|---|---|
| $P_{size}$ | 100 |
| $A_{size}$ | 100 |
| $P_{cross}$ | 0.8 |
| $P_{mut}$ | 0.01 |
| $P_{LS}$ | 0.05 |

The results for SPEA2HCS compared to SPEA2 are – as anticipated – much better for the 'easier' model DTLZ3* since in that case the probability is very high that the HCS converges to a global solution starting from a randomly chosen point, and the local search can thus contribute to increase the overall performance. If the model gets more complicated the improvements achieved by the local search procedure decreases, and global search operators get more important. Thus, it comes as no surprise that the improvements achived by the memetic strategy for DTLZ3 are less significant than for DTLZ3*.
It has to be noted that all the results presented here come from 3-objective models. This is due to the fact that for all bi-objective models that we have tested no remarkable improvements have been achieved which indicates that SPEA2 (as well as other state-of-the-art EMOs) is already very efficient on the well-known bi-objective benchmark suite – and certainly on other models as well. However, since the performance of most EMO algorithms decreases significantly with increasing number of objectives (starting with $k \geq 3$), the usage of memetic algorithms – such as the one proposed here – seems to be advantageous for the numerical treatment of MOPs which contain more than two objectives.

## 5. CONCLUSIONS AND FUTURE WORK

In this paper we have presented the Hill Climber with Sidestep (HCS), a derivative free local iterative search procedure. This iteration process aims either to move toward or along the (local) Pareto set depending on the location of the current iterate. Further, a novel memetic strategy has been proposed by integrating the HCS into a given EMO. Finally, we have demonstrated the applicability and the strength of the HCS alone and within SPEA2 as a local search operator on several benchmark optimization problems. The results indicate that the local search procedure can be advantageous in particular in the cases where a classical EMO gets 'stuck'. However, due to the nature of local search, the advantage of memetic strategies such as the one presented in this paper decreases with increasing complexity of the underlying model. Thus, overwhelming results compared to 'classical' EMOs cannot be expected in general.
The particular integration of the HCS into a given EMO we have presented has the advantage that in general every EMO can be taken and enhanced to become memetic. However, we expect that a more sophisticated interplay of the global and local search operators will lead to more efficient algorithms, and exactly this will be part of our future research.

*Acknowledgements.* The authors would like to thank the anonymous reviewers for their valuable comments which greatly helped them to improve the contents of this paper. The third author gratefully acknowledges support from the CONACyT project no. 45683-Y.

## 6. REFERENCES

[1] P. A. N. Bosman and E. D. de Jong. Exploiting gradient information in numerical multi-objective evolutionary optimization. In *Genetic and Evolutionary Computation Conference*, ACM, 2005.

[2] J. Branke and S. Mostaghim. About selecting the personal best in multi-objective particle swarm optimization. In *Parallel Problem Solving from Nature - PPSN IX*, pages 523–532, Springer Berlin / Heidelberg, 2006.

[3] M. Brown and R. E. Smith. Directed multi-objective optimization. *International Journal on Computers, Systems, and Signals*, 6(1):3–17, 2005.

[4] C. A. Coello Coello, D. A. Van Veldhuizen, and G. B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, New York, May 2002. ISBN 0-3064-6762-3.

[5] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley, 2001.

[6] K. Deb and R. B. Agrawal. Simulated binary crossover for continuous search space. *Complex Systems*, 9:115–148, 1995.

[7] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimisation: NSGA-II. In *PPSN VI: Proceedings of the 6th International Conference on Parallel Problem Solving from Nature*.

[8] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable multiobjective optimization test problems. *In Congress on Evolutionary Computation, volume 1, Piscataway, New Jersey*, pages 825 –830, 2002.

[9] J.E. Dennis and R.B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, 1983.

**Table 4: Numerical result for SPEA2 and SPEA2HCS on the DTLZ test functions (see Table 2). For each problem the number of function calls was fixed to 100,000, and the two algorithms were run for 30 times. The resulting mean value and standard deviation (in brackets) are presented.**

| | Indicators | | | |
|---|---|---|---|---|
| **Problems** | | $GD$ | $ESS$ | $MD$ |
| $DTLZ1$ | without LS | 8.2042(3.3349) | 8.0671(4.3136) | 291.1858(96.4631) |
| | with LS | 4.6663(1.3341) | 5.9063(5.1751) | 71.2207(38.4968) |
| $DTLZ2$ | without LS | 0.0298(0.0094) | 0.0501(0.0146) | 2.4125(0.2907) |
| | with LS | 0.0044(0.0022) | 0.0306(0.0103) | 1.5691(0.1018) |
| $DTLZ3$ | without LS | 11.2395(7.9325) | 8.4543(6.7640) | 300.4770(185.2523) |
| | with LS | 6.0506(3.6930) | 14.9354(12.8529) | 197.0717(99.7800) |
| $DTLZ3^*$ | without LS | 18.3752(15.0230) | 9.2857(10.2005) | 427.8683(318.8405) |
| | with LS | 0.1942(0.3043) | 0.9345(2.2947) | 13.9865(24.0805) |


[10] J. Fliege and B. F. Svaiter. Steepest descent methods for multicriteria optimization. *Mathematical Methods of Operations Research*, 51(3):479–494, 2000.

[11] T. Goel and K. Deb. Hybrid methods for multi-objective evolutionary algorithms. In *Proceedings of the 4th Asia-Pacific conference on Simulated Evolution and Learning*, pages 188–192, 2002.

[12] C. Hillermeier. *Nonlinear Multiobjective Optimization - A Generalized Homotopy Approach*. Birkhäuser, 2001.

[13] X. Hu, Z. Huang, and Z. Wang. Hybridization of the multi-objective evolutionary algorithms and the gradient-based algorithms. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 870 – 877., 2003.

[14] H. Ishibuchi and T. Murata. Multi-objective genetic local search algorithm. In *Proc. of 3rd IEEE Int. Conf. on Evolutionary Computation, Nagoya, Japan*, pages 119–124, 1996.

[15] J. D. Knowles and D. W. Corne. *Recent Advances in Memetic Algorithms*, volume 166, chapter Memetic Algorithms for Multi-Ojective Optimization: Issues, Methods and Prospects, pages 313–352. Springer. Studies on Fuzzines and Soft Computing, 2005.

[16] J.D. Knowles and D.W. Corne. M-PAES: a memetic algorithm for multiobjective optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 325 – 332, 2000.

[17] 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.

[18] P. Moscato. On evolution, search, optimization, genetic algorithms and martial arts: Toward memetic algorithms. *Technical Report C3P Report 826, Caltech Concurrent Computation Program*, 1989.

[19] J.R. Schott. Fault tolerant design using single and multicriteria genetic algorithm optimization. *Master's Thesis, Department of Aeronautics and Astronautics, MIT. Cambridge, MA. USA*, 1995.

[20] O. Schütze, A. Dell'Aere, and M. Dellnitz. On continuation methods for the numerical treatment of multi-objective optimization problems. In J. Branke, K. Deb, K. Miettinen, and R. 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>.

[21] 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. Proceedings of the Genetic and Evolutionary Computation Conference, 2007.

[22] O. Schütze, E.-G. Talbi, C. A. Coello Coello, L. V. Santana-Quintero, and G. Toscano Pulido. A memetic PSO algorithm for scalar optimization problems. Proceedings of IEEE Swarm Intelligence Symposium (SIS2007), 2007.

[23] D. Sharma, A. Kumar, K. Deb, and K. Sindhya. Hybridization of SBX based NSGA-II and sequential quadratic programming for solving multi-objective optimization problems. In *Proceedings of the IEEE Congress on Evolutionary Computation*.

[24] V. Pareto. *Manual of Political Economy*. The MacMillan Press, 1971 (original edition in French in 1927).

[25] D.A. Van Veldhuizen and G.B. Lamont. Evolutionary computation and convergence to a Pareto front. In *Late Breaking papers at the Genetic Programming 1998 Conference*.

[26] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195, 2000.

[27] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength Pareto evolutionary algorithm. In K. Giannakoglou, D. Tsahalis, J. Periaux, P. Papailou, and T. Fogarty, editors, *EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, 2002.