HCS: A New Local Search Strategy for Memetic Multi-Objective Evolutionary Algorithms

Oliver Schütze[1], Gustavo Sanchez[2],
Adriana Lara[1], and Carlos A. Coello Coello[1⋆]

[1] CINVESTAV-IPN, Computer Science Department, Mexico City, Mexico
{schuetze,lara,ccoello}@cs.cinvestav.mx,
[2] Simon Bolivar University, Caracas, Venezuela
gsanchez@usb.ve

Abstract. In this paper we propose and investigate a new local search strategy for multi-objective memetic algorithms. For this, we suggest a novel iterative search procedure, the HCS, which is designed for the treatment of multi-objective optimization problems, and show further on one possible way to integrate the HCS into a given evolutionary strategy leading to a new memetic (or hybrid) algorithm. The pecularity of the HCS, the *Hill climber with Sidestep*, is that it is intended to be capable both of moving toward and along the (local) Pareto set depending on the distance of the current iterate toward this set. The local search procedure utilizes the geometry of the directional cones of such optimization problems and works with or without gradient information. Finally, we present some numerical results on some well-known benchmark problems indicating the strength of the local search strategy as a standalone algorithm as well as the new hybrid approach, i.e., the HCS embedded into a multi-objective evolutionary algorithm.

Key words: multi-objective optimization, heuristic search, memetic algorithm, hill climber, Pareto set

1 Introduction

In a variety of applications in industry and finance one is faced with the problem that several objectives have to be optimized concurrently leading to a *multi-objective optimization problem* (MOP). 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).

⋆ The fourth author is also affiliated to the UMI-LAFMIA 3175 CNRS

For the computation of the Pareto set of a given MOP there exist several classes of algorithms. There exist, for instance, a variety of mathematical programming techniques such as scalarization methods (see e.g., [31, 16, 8] and references therein) or continuation methods [21] which are in general very efficient in finding single solutions—the most prominent example is probably Newton's method which is used within continuation methods and which has local quadratic convergence [33]—but which may have trouble in finding the entire (global) Pareto set in certain cases. In contrast, there are global methods including multi-objective evolutionary algorithms (MOEAs) [10, 7] or subdivision techniques [44, 14] which accomplish the 'global task' exceedingly but offer in turn (much) slower convergence rates compared to the algorithms mentioned above.

Another class of algorithms are the *memetic* (or hybrid) algorithms, i.e., algorithms which hybridize MOEAs with local search strategies (see Section 2.2 for an overview of existing methods). This is done in order to obtain an algorithm which offers on one side the globality and robustness of the evolutionary approach, but on the other side also an improved overall performance by the inclusion of well directed local search.

The scope of this paper is to present a novel approach which fits into the last category of algorithms. To be more precise, according to the classification made in [46], we suggest an *exploitation-embedded hybrid method*. The core of the method is a local search prodecure, the *Hill Climber with Sidestep* (HCS). This point-wise iteration process aims to find a sequence of 'better' solutions (hill climber). In case the actual iterate is already 'near' to a local solution the process automatically tries to determine the subsequent iterates *along* the Pareto set (sidestep). A preliminary study of this work can be found in [42].

The remainder of this paper is organized as follows: In Section 2, we state some theoretical background and give an overview on existing memetic MOEAs (MEMOEAs). In Section 3, we introduce the idea of the HCS and propose two realizations, and give further on in Section 5 one possible way to integrate this algorithm into a global search procedure. In Section 5, we show some numerical results on both the HCS as a standalone algorithm as well as one resulting MEMOEA. Finally, some conclusions are drawn in Section 5.

2 Background

Here we briefly describe the background required for this paper: we introduce to the notion of multi-objective optimization and give an overview on existing memetic strategies for the numerical treatment of such problems.

2.1 Multi-objective Optimization (MOO)

In a variety of applications in industry and finance a problem arises that several objective functions have to be optimized concurrently leading to *multi-objective*

optimization problems (MOPs). In the following we consider continuous MOPs which are of the following form:

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

where $Q \subset \mathbb{R}^n$ is the domain 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. In this work we will mainly consider the unconstrained case (i.e., $Q = \mathbb{R}^n$) but will give some possible modifications of the algorithms in case Q is defined by inequality constraints such as box constraints.
Central for the treatment of MOPs is the concept of the optimality of a point $x \in Q$ which is not analogue to the scalar objective case ($k = 1$). In the multi-objective case ($k > 1$) the concept of *dominance* is used which dates back over a century and was proposed first by Pareto [35].

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 $i \in \{1, \ldots, k\}$. 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.*

In case all the objectives $f_i, i = 1, \ldots, k$, of the MOP are differentiable the following theorem of Kuhn and Tucker [29] states a necessary condition for Pareto optimality for unconstrained MOPs. For a more general formulation of the theorem we refer e.g. to [31].

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*

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

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, (1) is not a sufficient condition for Pareto optimality. On the other hand, points satisfying (1) are certainly 'Pareto candidates'.

Definition 2. *A point $x \in \mathbb{R}^n$ is called a* Karush–Kuhn–Tucker point[3] *(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 (1) is satisfied.*

[3] Named after the works of Karush [26] and Kuhn & Tucker [29].

The set of all (globally) Pareto optimal solutions is called the *Pareto set*. It has been shown that this set typically—i.e., under mild regularity assumptions—forms a $(k-1)$-dimensional object [21]. The image of the Pareto set is called the *Pareto front*. Since we are involving local search strategies in our work we have to take also locally optimal points into consideration. In the following, let $\mathcal{P}$ be the set of local Pareto points. In case the MOP is differentiable, $\mathcal{P}$ can be considered as the set of KKT–points.

Theorem 2 ([39]). *Let (MOP) be given and $q : \mathbb{R}^n \to \mathbb{R}^n$ be defined by*

$$q(x) = \sum_{i=1}^{k} \hat{\alpha}_i \nabla f_i(x), \tag{2}$$

where $\hat{\alpha}$ is a solution of

$$\min_{\alpha \in \mathbb{R}^k} \left\{ \| \sum_{i=1}^{k} \alpha_i \nabla f_i(x) \|_2^2 ; \alpha_i \geq 0, i = 1, \ldots, k, \sum_{i=1}^{k} \alpha_i = 1 \right\}. \tag{3}$$

Then either $q(x) = 0$ or $-q(x)$ is a descent direction for all objective functions $f_1, \ldots, f_k$ in x.

The theorem states that for every point $x \in Q$ which is not a KKT–point a descent direction (i.e., a direction where all objectives' values can be improved) can be found by solving the quadratic optimization problem (3). In case $q(x) = 0$ the point x is a KKT–point. Thus, a test for optimality has to be performed automatically when computing the descent direction for a given point $x \in Q$.

2.2 Memetic Strategies in MOO

Hybridization of MOEAs with local search algorithms has been investigated for more than twelve years, starting short time after the first MOEAs were proposed [28, 7]. One of the first MEMOEAs for models on discrete domains was presented in [24] as a 'Multi-Objective Genetic Local Search' (MOGLS) approach. The authors proposed to use the local search method after classical variation operators are applied. A randomly drawn scalarizing function is used to assign fitness for parent selection.

Jaszkiewicz [25] proposed an algorithm called the Pareto Memetic Algorithm (PMA). This algorithm uses an unbounded 'current set' of solutions (CS) and from this selects a small 'temporary population' (TP) that comprises the best solutions with respect to a scalarizing function. Then TP is used to generate offspring by crossover. Jaszkiewicz suggests that scalarizing functions are particularly better at encouraging diversity than dominance ranking methods used in most MOEAs.

Another important MEMOEA, called M-PAES, was proposed in [27]. Unlike Ishibuchi's and Jaszkiewicz's approaches, M-PAES does not use scalarizing

functions, but employs instead a Pareto ranking based selection coupled with a grid-type partition of the objective space. Two archives are used: one that maintains the global non-dominated solutions and the other that is used as the comparison set for the local search phase.

In [32], the authors proposed a local search process with a generalized replacement rule. Ordinary two-replacement rules based on the dominance relation are usually employed in a local search for multiobjective optimization. One is to replace a current solution with a solution which dominates it. The other is to replace the solution with a solution which is not dominated by it. The movable area with the first rule is very small when the number of objectives is large. On the other hand, it is too huge to move efficiently with the latter. The authors generalize these extreme rules by counting the number of improved objectives for a given candidate.

In [47, 48], methods are presented which are hybrids of evolutionary search algorithms and multi-agent strategies where the task of the agents is to perform the local search. The continuous case—i.e., continuous objectives defined on a continuous domain—was first explored in [18], where a neighborhood search was applied to NSGA-II [12]. In their initial work, the authors applied the local search only after NSGA-II had ended. To do this, the authors applied a local search using a weighted sum of objectives. The weights were computed for each solution based on its location in the Pareto front such that the direction of improvement is roughly in the direction perpendicular to the Pareto front. 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 [22] a gradient-based local algorithm (Sequential Quadratic Programming (SQP)), was used in combination with NSGA-II and SPEA [54] to solve the ZDT benchmark suite [52]. The authors stated that if there are no local Pareto fronts, the hybrid MOEA has faster convergence toward the true Pareto front than the original one, either in terms of the objective function evaluations or in terms of the CPU time consumed (since a gradient-based algorithm is utilized, the sole usage of the number of function calls as a basis for a comparison can be misleading). Furthermore, they found that the hybridization technique does not decrease the solution diversity.

In [1] a new local search technique was introduced and hybridized with MOGA [17]. The new hybrid algorithm was tested against a set of optimization problems of varying complexity. Three important questions were addressed: Where shall a local search process be hybridized with a genetic algorithm? Which individuals should be fine-tuned? And when shall the local refinement be applied? To the best of the authors' knowledge, these remain as open questions in the field of hybrid evolutionary algorithms.

In [37], the authors proposed a hybrid technique that combines the robustness of MOGA-II [36] with the accuracy and speed of NBI-NLPQLP, an accurate and fast converging algorithm based on a classical gradient method. The methodology consists of starting with a preliminary robust MOGA-II run, then isolating each

single portion of the Pareto curve as an independent problem, each of which is treated with an independent accurate NBI-NLPQLP run.

In [51] the proposed local search process employs quadratic approximations for all objective functions. The samples gathered by the algorithm along the evolutionary process are used to fit these quadratic approximations around the point selected for local search. After that, a locally improved solution is estimated from the quadratic associated problem. The hybridization of the procedure is demonstrated with SPEA 2 [53].

A succesful hybrid approach was proposed in [23]. The authors proposed the algorithm MO-CMA-ES, a multi-objective CMA-ES [19], which combines the strategy parameter adaptation of evolutionary strategies with a multi-objective selection based on non-dominated sorting. The MO-CMA-ES is independent of the chosen coordinate system and its behavior does not change if the search space is translated, rotated, and/or rescaled. The authors claim that MO-CMA-ES significantly outperforms NSGA-II on all but one of the considered test problems: the NSGA-II is faster only on the ZDT4 problem where the optima form a regular axis-parallel grid, because NSGA-II heavily exploits this kind of separability.

In [50], a novel evolutionary algorithm (EA) for constrained optimization problems is presented: the so-called hybrid constrained optimization EA (HCOEA). The algorithm combines multi-objective optimization with global and local search processes. In performing the global search, a niching genetic algorithm based on tournament selection is used. Meanwhile, the best infeasible individual replacement scheme is used as a local search operator for the purpose of guiding the population toward the feasible region of the search space. During the evolutionary process, the global search model effectively promotes high population diversity, and the local search model remarkably accelerates the convergence speed. HCOEA was tested on 13 benchmark functions, and the experimental results suggest that it is more robust and efficient than other state-of-the-art algorithms in terms of the selected performance metrics.

Finally, in [20, 43], hybrids can be found were heuristic methods are coupled with multi-objective continuation methods.

3 HCS: The Hill Climber with Sidestep

In the following, we propose a novel iterative local search procedure, the HCS, which is designed for a use within a memetic strategy.

Before we can come to the design of such a strategy, we have to ask ourselves what are the requirements for an iterative search procedure $\Phi : \mathbb{R}^n \to \mathbb{R}^n$ with

$$x_{l+1} = \Phi(x_l), \tag{4}$$

where $x_0 \in \mathbb{R}^n$ is a given initial solution and $\{x_i\}_{i \in \mathbb{N}_0}$ is the resulting sequence of iterates. Note that we are dealing with a point-wise iteration—i.e., input and output of Φ are a single points of the domain—, and not with a population based strategy. We are of the opinion that such a 'whish list' on Φ for the treatment of MOPs includes the following tasks:

(a) Φ should generate an improvement of the current iterate x_l if this one is not already 'close' to the $\mathcal{P}$, i.e., a point x_{l+1} with $x_{l+1} \prec x_l$.
(b) In case the current iterate x_l is already 'close' to $\mathcal{P}$, a search *along* $\mathcal{P}$ would be desired.
(c) The switch between the situations described in (a) and (b) should be done *automatically* according to the position of the current iterate x_l.
(d) The process should work with or without gradient information (whether or not provided by the model).
(e) The process should be capable of handling constraints of the MOP.

In (a) the 'classical' task of a hill climber as known for single-objective optimization problems [15, 34, 38, 30] is described. Item (b) contains a pecularity of multi-objective optimization, namely that there is—using the climbing metaphor—no single mountain top but rather an entire ridge of mountain tops which forms $\mathcal{P}$ (respectively a set of ridges in case $\mathcal{P}$ is diconnected). The generation of such a point x_{l+1} can be regarded as a 'sidestep' relative to the current iterate x_l in the upward movement of the hill climber. Important for the efficiency of Φ within a memetic strategy is item (c), i.e., the capability to decide if case (a) or (b) is more appropriate.
In the following we describe two variants of such a function Φ which aims to fulfil the above wish list: one version of the HCS which is gradient-free, and another version which involves gradient information.

3.1 HCS without Using Gradient Information

First, we describe the HCS algorithm for the case in which no gradient information is available, since that seems to be more relevant for common real-world engineering problems which is the main area of application for MOEAs. We concentrate here on the unconstrained case and possible modifications of the algorithm for the treatment of MOPs with inequality constraints are given below.

The method we describe here is based on the geometry of multi-objective optimization which has been studied in [6]. This work gives a good insight into the structure of such problems by analyzing the geometry of the directional cones of canditate solutions at different stages of the optimization process: when a point x_0 is 'far away' from any local Pareto optimal solution, 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 ν, there is a nearly 50 % chance that this direction is a descent direction at x_0 (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 [29] which holds for points on $\mathcal{P}$), 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. The two scenarios are depicted in Figure 1 for the bi-objective case. Hereby, $\{-,-\}$ and $\{+,+\}$ denote the descent and ascent cone, respectively. The symbol $\{-,+\}$ indicates that in this direction an improvement

according to f_1 can be achieved while the values of f_2 will increase (this is analogous for $\{+,-\}$).

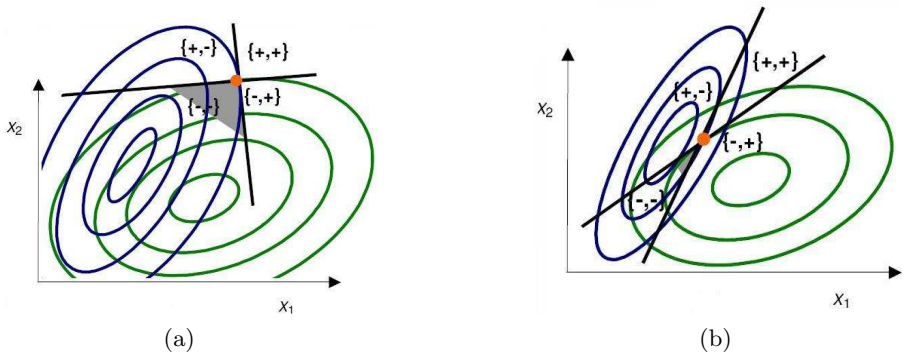


Fig. 1. The descent cone (shaded) for an MOP with 2 parameters and 2 objectives during initial (a) and final (b) stages of convergence. The descent cone shrinks to zero during the final stages of convergence. The figure is taken from [4].

The gradient-free HCS is constructed on the basis of these observations. Given a point $x_0 \in Q$, the next iterate x_1 is selected as follows: a further point $\tilde{x}_1$ is chosen randomly from a neighborhood of x_0, say $\tilde{x}_1 \in B(x_0, r)$ with

$$B(x_0, r) := \{x \in \mathbb{R}^n : x_{0,i} - r_i \leq x_i \leq x_{0,i} + r_i \; \forall i = 1, .., n\}, \tag{5}$$

where $r \in \mathbb{R}^n_+$ is a given (problem dependent) radius. If $\tilde{x}_1 \prec x_0$, then $\nu := \tilde{x}_1 - x_0$ is a descent direction[4] 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., $\nu := x_0 - \tilde{x}_1$) and starting with $\tilde{x}_1$. If x_0 is 'far away' from any local solution, the chance is by the above discussion quite high that domination occurs, either $\tilde{x}_1 \prec x_0$ or $x_0 \prec \tilde{x}_1$. 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. This is because even if a descent direction would be available further improvements would very likely be negligible, and, hence, it is advisable to seek for further regions of the Pareto set. To perform such a sidestep it would be desirable to use the accumulated information obtained by the unsuccessful trials. Fundamental for the algorithm

[4] 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.

we present here is the fact that the 'unsuccessful' search directions $\nu_{i,1} := \tilde{x}_i - x_0$ and $\nu_{i,2} := x_0 - \tilde{x}_i = -\nu_{i,1}$ are located in the diversity cones. Further, there exists the following relation of $\nu_{i,1}$ and $\nu_{i,2}$: if $\nu_{i,1}$ is, for example, in the cone $\{+,-\}$, then $\nu_{i,2}$ is the opposite cone $\{-,+\}$ (this is for bi-objective MOPs, the general k-objective case is analogue).

Based on these observations we propose the following search directions. First we address the bi-objective case. If, for example, a search along $\{-,+\}$ after N_{nd} unsuccessful trials is sought, we propose to use the following one which uses the previous information:

$$\nu_{acc} = \frac{1}{N_{nd}} \sum_{i=1}^{N_{nd}} s_i \frac{\tilde{x}_i - x_0}{\|\tilde{x}_i - x_0\|}, \tag{6}$$

where

$$s_i = \begin{cases} 1 & \text{if } f_1(\tilde{x}_i) < f_1(x_0) \\ \text{-1} & \text{else} \end{cases} \tag{7}$$

By construction, ν_{acc} is in $\{-,+\}$, and by the averaging of the search directions we aim to obtain a direction which is 'perpendicular' to the (small) descent cone. Note that in this case ν_{acc} is indeed a 'sidestep' to the upward movement of the hill climbing process as desired, but this search direction does not necessarily have to point along the Pareto set (see next subsection for a better guided search). A similar strategy for the search can be done for a general number k of objectives, however, leading to a larger variety for the search direction. For instance, for $k = 3$, there are six diversity cones which can be grouped by reflection as follows:

$$\begin{array}{lll} \{+,-,-\} & \text{and} & \{-,+,+\}, \\ \{+,-,+\} & \text{and} & \{-,+,-\}, \\ \{+,+,-\} & \text{and} & \{-,+,+\}. \end{array} \tag{8}$$

That is, for $k = 3$ there are three different groups of cones in which search directions can be divided. For a general k there are a total of $2^{k-1} - 1$ different groups making it less likely to find a perpendicular direction due to averaging within N_{nd} trials and within one of these cones. Alternatively to (6) on can e.g. use the accumulated information by taking the average search direction over *all* search directions as follows:

$$\nu_{acc} = \frac{1}{N_{nd}} \sum_{i=1}^{N_{nd}} \frac{\tilde{x}_i - x_0}{\|\tilde{x}_i - x_0\|}, \tag{9}$$

This direction has previously been proposed as a local guide for a multi-objective particle swarm algorithm in [5]. Note that this is a heuristic that does *not* guarantee that ν_{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., it searches for better points—which is still in the scope of the procedure (though the improvements may not be significant due to the vicinity of the current iterate to $\mathcal{P}$).

A pseudocode of the HCS for the bi-objective case which uses the strategies described above and the sidestep heuristic (6) is given in Algorithm 1. In the following we go into detail for possible realizations of the line search and the handling of the constraints.

Algorithm 1 HCS1 (without using gradient information)

Require: starting point $x_0 \in Q$, radius $r \in \mathbb{R}^n_+$, number N_{nd} of trials, MOP with $k = 2$
Ensure: sequence $\{x_l\}_{l\in\mathbb{N}}$ of candidate solutions

1: $a := (0, \ldots, 0) \in \mathbb{R}^n$
2: $nondom := 0$
3: **for** $l = 1, 2, \ldots$ **do**
4: set $x_l^1 := x_{l-1}^b$ and choose $x_l^2 \in B(x_l^1, r)$ at random
5: choose $i_0 \in \{1, 2\}$ at random
6: **if** $x_l^1 \prec x_l^2$ **then**
7: $\nu_l := x_l^2 - x_l^1$
8: compute $t_l \in \mathbb{R}_+$ and set $\tilde{x}_l^n := x_l^2 + t_l\nu_l$.
9: choose $x_l^b \in \{\tilde{x}_l^b, x_l^1\}$ such that $f(x_l^b) = \min(f(\tilde{x}_l^n), f(x_l^1))$
10: $nondom := 0, \quad a := (0, \ldots, 0)$
11: **else if** $x_l^2 \prec x_l^1$ **then**
12: proceed analogous to case "$x_l^1 \prec x_l^2$" with
13: $\nu_l := x_l^1 - x_l^2$ and $\tilde{x}_l^n := x_l^1 + t_l\nu_l$.
14: **else**
15: **if** $f_{i_0}(x_l^2) < f_{i_0}(x_l^1)$ **then**
16: $s_l := 1$
17: **else**
18: $s_l := -1$
19: **end if**
20: $a := a + \frac{s_l}{N_{nd}} \frac{x_l^2 - x_l^1}{\|x_l^2 - x_l^1\|}$
21: $nondom := nondom + 1$
22: **if** $nondom = N_{nd}$ **then**
23: compute $\tilde{t}_l \in \mathbb{R}_+$ and set $\tilde{x}_l^n := x_l^1 + \tilde{t}_l a$.
24: $nondom := 0, \quad a := (0, \ldots, 0)$
25: **end if**
26: **end if**
27: **end for**

Sidestep direction The direction for the sidestep is determined by the value of i_0 (see line 5 and lines 15-20 of Algorithm 1). For simplicity, in Algorithm 1 the value of i_0 is chosen at random. In order to introduce an orientation to the search, the following modifications can be done in the bi-objective case: in the

beginning, i_0 is fixed to 1 for the following iteration steps. When the sidestep (line 23 of Algorithm 1) has been performed N_s times during the run of an algorithm, this indicates that the current iteration is already near to the (local) Pareto set, and this vector is stored in x_p. If in the following 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. That is, i_0 is set to -1 for the following steps. A possible stopping criterion, hence, could be to stop the process when no improvements can be achieved according to f_2 within another N_i sidesteps along $\{+,-\}$ (this has in fact been chosen as the stopping criterion in Section 5.1).

Computation of t_l The situation is that we are given two points, say $x_0, x_1 \in \mathbb{R}^n$, 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 to perform the line search (see e.g., [15]). We propose to proceed in analogy to [45], 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

$$f_\nu : \mathbb{R} \to \mathbb{R}, \quad f_\nu(t) = f(x_0 + t\nu) \tag{10}$$

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_ν by a quadratic polynomial $p(t) = at^2 + bt + c$ with coefficients $a, b, c \in \mathbb{R}$. Using the interpolation conditions

$$p(0) = f_\nu(0), \quad p(1) = f_\nu(1), \quad p(e) = f_\nu(e), \tag{11}$$

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

$$t^*_{i_j} = \frac{-b}{2a} = 2\frac{e^2(f_\nu(1) - f_\nu(0)) - f_\nu(e) + f_\nu(0)}{e(f_\nu(1) - f_\nu(0)) - f_\nu(e) + f_\nu(0)} \in (0, e). \tag{12}$$

The idea to approximate f_ν locally by a quadratic polynomial was first proposed by Armijo [3].

Finally, the question that arises is 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.

$$x_{new} = x_0 + t^*_k \nu, \tag{13}$$

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 $\tilde{t}_l$ We are given a point $x_0 \in \mathbb{R}^n$ and the search direction $a = \sum_{i=1}^{N_{nd}} s_i(\tilde{x}_i - x_0)/\|\tilde{x}_i - x_0\|$ (or alternatively direction (6)) 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 [41], where a step size strategy for multi-objective continuation methods is suggested: given a target value $\epsilon_y \in \mathbb{R}_+$—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

$$\|F(x_0) - F(x_{new})\|_\infty \approx \epsilon_y \tag{14}$$

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

$$\|F(x) - F(y)\| \leq L\|x - y\|, \quad \forall x, y \in Q. \tag{15}$$

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 the derivatives of F are not given (which is considered in this section) the accumulated information can be used to compute the estimation

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

since the $\tilde{x}_i$'s are near to x_0. Combining (14), (15) and using the estimation L_{x_0} leads to the step size control

$$x_{new} = x_0 + \frac{\epsilon_y}{L_{x_0}} \frac{a}{\|a\|_\infty}. \tag{16}$$

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 ν 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 ∂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 ∂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).

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 $\|out_i - in_i\| \geq tol$ **do**
  $m_i := in_i + \frac{1}{2}(out_i - in_i)$
  if $m_i \in Q$ **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 $\tilde{x} := in_i$

Design parameters We agree that a realization of Algorithm 1 may include a variety of design parameters which may be difficult to tune and adapt to a particular problem. However, if the suggestions made in this paper are taken merely the values for four design parameters have to be chosen (see Table 1): the parameter r defines the neigborhood search of the procedure. Since this neighorbood search is used to find a search direction which is afterwards coupled with a step size control, the value of r is not that important, but should be 'small' to guarantee a local search. N_{nd} is the value which determines the number of directions which have to be averaged in order to choose the sidestep direction. In general, a larger value of N_{nd} leads to a 'better' sidestep (in the sense that the search is performed orthogonal to the upward movement), but will in turn increase the cost of the search. We have experienced that a low value N_{nd}, say 5 to 10, already gives satisfactory results, the 'accuracy' of the search does not seem to influence the performance of the HCS (unless the second derivatives of the objectives are available, see below). The value of ϵ_y is problem dependent but can be given quite easily in a real world application (see discussion above Equation (14)). Finally, the tolerance *tol* has to be adjusted for constrained MOPs. The choice of this value is also problem dependent and has to be chosen in every algorithm dealing with constraints.

3.2 HCS Using Gradient Information

In this section we discuss possible modifications which can be made to increase the performance of the HCS in case the MOP is sufficiently smooth. It will turn out that the resulting algorithm is more efficient (see Section 5), but in turn, more information of the model is required.
Here we describe one possible realization of the HCS using the descent direction

Table 1. Design parameters that are required for the realization of the gradient-free 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)
ϵ_y	Desired distance (in image space) for the sidestep (7)
tol	Tolerance value used for the backtracking in Alg. 2

presented in Theorem 2 for the hill climber and some elements from multi-objective continuation for the sidestep:

Given a point $x \in \mathbb{R}^n$ the quadratic optimization problem (3) can be solved leading to the vector $\hat{\alpha}$. In case

$$\| \sum_{i=1}^{k} \hat{\alpha}_i \nabla f_i(x) \|_2^2 \geq \epsilon_{\mathcal{P}}, \tag{17}$$

i.e., if the square of the norm of the weighted gradients is larger than a given threshold $\epsilon_{\mathcal{P}} \in \mathbb{R}_+$, the candidate solution x can be considered to be 'away' from $\mathcal{P}$, and thus, it makes sense to seek for a dominating solution. For this, the descent direction (2) can be taken together with a suitable step size control. For the latter the step size control described above can be taken, or—probably better—a step size control which uses gradient information as e.g. described in [15] or the one presented in [14]. If the value of the term in (17) is less than $\epsilon_{\mathcal{P}}$, this indicates that x is already in the vicinity of $\mathcal{P}$. In that case one can lean elements from (multi-objective) continuation [21, 2] to perform a search along $\mathcal{P}$. To do this, we assume for simplicity that we are given a KKT–point $\hat{x}$ and the according weight $\hat{\alpha}$ obtained by (3). Then the point $(\hat{x}, \hat{\alpha}) \in \mathbb{R}^{n+k}$ is obviously contained in the zero set of the auxiliary function $\tilde{F} : \mathbb{R}^{n+k} \to \mathbb{R}^{n+1}$ of the given MOP which is defined as follows:

$$\tilde{F}(x, \alpha) = \begin{pmatrix} \sum_{i=1}^{k} \alpha_i \nabla f_i(x) \\ \sum_{i=1}^{k} \alpha_i - 1 \end{pmatrix}. \tag{18}$$

In [21] it has been shown that the zero set $\tilde{F}^{-1}(0)$ can be linearized around $\hat{x}$ by using a QU-factorization of $\tilde{F}'(\hat{x}, \hat{\alpha})^T$, i.e., the transposed of the Jacobian matrix of $\tilde{F}$ at $(\hat{x}, \hat{\alpha})$. To be more precise, given a factorization

$$\tilde{F}'(\hat{x}, \hat{\alpha})^T = QU \in \mathbb{R}^{(n+k)\times(n+k)}, \tag{19}$$

where $Q = (Q_N, Q_K) \in \mathbb{R}^{(n+k)\times(n+k)}$ is orthogonal with $Q_N \in \mathbb{R}^{(n+k)\times(n+1)}$ and $Q_K \in \mathbb{R}^{(n+k)\times(k-1)}$, the column vectors of Q_K form—under some mild

regularity assumptions on $\tilde{F}^{-1}(0)$ at $(\hat{x}, \hat{\alpha})$, see [21]—an orthonormal basis of the tangent space of $\tilde{F}^{-1}(0)$. Hence, it can be expected that each column vector $q_i \in Q_K$, $i = 1, \ldots, k-1$, points (locally) along $\mathcal{P}$ and is thus well suited for a sidestep direction. The step size control can in this case taken exactly as proposed in Equation (16) since the setting for that case was the same. In fact, since the search direction q_i is indeed pointing along $\mathcal{P}$, the results will be more accurate than for an averaged direction such as (6) or (9).
Algorithm 3 presents a procedure which is based on the above discussion. Note that this is one possible realization and that there exist certainly other possible ways leading, however, to similar results. For instance, alternatively to the descent direction used in Algorithm 3 the ones proposed in [16] and [4] can be taken. Further, the vicinity test (17) can be changed, though alternative conditions will most likely also be based on Theorem 1. Finally, the movement along $\mathcal{P}$ can be realized by predictor-corrector methods [21, 2] which consist, roughly speaking, of a repeated application of a predictor step obtained by a linearization of $\tilde{F}^{-1}(0)$ as in (19) and a corrector step which is done via a Gauss-Newton method.
Note that the HCS is proposed for the unconstrained case. While an extension to the contrained case for the hill climber is possible (see, e.g., [16] for possible modifications) this does not hold for the movement along the Pareto set (i.e., the sidestep). Though it is possible to extend system (18) by equality constraints (e.g., by introducing slack variables to transform the inequality constraints into equality constraints) this could lead to effiency problems in the numerical treatment [21]. Hence, we restrict ourselves here to the unconstrained case.

As it will be shown in Section 5 the performance of the gradient based HCS in terms of convergence is better than its gradient-free version, but this improvement does not come for free: for the descent direction all objectives' gradients have to be available (or approximated), and to perform the linearization of $\mathcal{P}$ even all second derivatives are required.

Algorithm 3 HCS2 (Using Gradient Information)

Require: starting point $x_0 \in Q$
Ensure: sequence $\{x_l\}_{l \in \mathbb{N}}$ of candidate solutions
for $l = 0, 1, 2, \ldots$ **do**
  compute the solution $\hat{\alpha}$ of (3) for x_l.
  if $\| \sum_{i=1}^{k} \hat{\alpha}_i \nabla f_i(x_l) \|_2^2 \geq \epsilon_{\mathcal{P}}$ **then**
    $\nu_l := -q(x_l)$
    compute $t_l \in \mathbb{R}_+$ and set $x_{l+1} := x_l + t_l \nu_l$
  else
    compute $\tilde{F}'(\hat{x}, \hat{\alpha})^T = (Q_N, Q_K)U$ as in (19)
    choose a column vector $\tilde{q} \in Q_K$ at random
    compute $\tilde{t}_l \in \mathbb{R}_+$ and set $x_{l+1} := x_l + \tilde{t}_l \tilde{q}$.
  end if
end for

Design parameters Analogue to the gradient-free version of the HCS, the values of some design parameters have to be chosen for the realization of Algorithm 3. ϵ_y and *tol* are as discussed above, and N_{nd} and r are not needed due to the accuracy of the gradient based search. A new parameter, compared to the gradient-free version of the HCS, is the threshold $\epsilon_{\mathcal{P}}$ for the vicinity test of a given candidate solution to $\mathcal{P}$. This value is certainly problem dependent, but it can be made 'small' due to the convergence properties of the hill climber (e.g., [16]).

Table 2. Design parameters that are required for the realization of the HCS algorithm which involves gradient information.

Parameter	**Description**
ϵ_y	Desired distance (in image space) for the sidestep (7)
tol	Tolerance value used for the backtracking in Alg. 2
$\epsilon_{\mathcal{P}}$	Threshold for the vicinity test (17)

4 A Memetic Multi-objective Strategy Using HCS

Having stated the local search procedure HCS, the question which arises is how it can be integrated efficiently into a given MOEA which we address here.

In Algorithms 1 and 3 the HCS is presented as a standalone algorithm generating an infinite sequence of candidate solutions, which is certainly not applicable when coupling it with a MOEA. To support the local search of the latter algorithm, it is rather advisable to stop the iteration after a few, probably even after one, iteration (see line 3 of Alg. 1 and line 1 of Alg. 3). The modified HCS can thus be simplified written as

$$P_{HCS} = HCS\,(x_0), \tag{20}$$

where x_0 is a given point (e.g., coming from the current population of the MOEA) and P_{HCS} the output set. Here we describe one possible realization for the case just one iteration is done (a generalization to the performance of more iteration steps of the HCS is analogue). For x_0 either P_{HCS} is set to $P_{HCS} = \{x_1\}$, where

$$x_1 := x_0 + t_0\nu_0 \tag{21}$$

in case a dominated point could be found (lines 7 and 8 respectively line 13 of Algorithm 1, or line 5 of Algorithm 3), or $P_{HCS} = \{x_1, x_2\}$, where

$$x_1 := x_0 + \tilde{t}_0\nu_0 \quad \text{and} \quad x_2 := x_0 - \tilde{t}_0\nu_0, \tag{22}$$

i.e., a search along the sidestep directions ν_0 and $-\nu_0$, are both added to P_{HCS} in case a sidestep is performed (line 20 of Algorithm 1 or line 9 of Algorithm 3). This is done because the computation of the sidestep direction is relatively expensive for both variants of the HCS and since the orientation of the search is chosen at random. Thus, a search in both directions seems to be advisable. Given a probability p_{hcs} for the application of the procedure (20) on an individual of a population, the operator can be defined set-wise as

$$P_{HCS} = HCS\,(P, p_{HCS}), \tag{23}$$

where P denotes a given population. By doing so, the HCS can be interpreted as a particular mutation operator. Algorithm 4 states a generic EA following the notation of [9], where p_c and p_m denote the probability for crossover and mutation, repectively. Obviously, the operator (23) fits into this framework which indicates that it can be integrated basically into any MOEA with little effort.

Algorithm 4 $P := Generic\_EA\,(p_c, p_m)$

```
 1: initialize P
 2: f := Evaluate (P)
 3: P := Select (P, f)
 4: l := 1
 5: while not_stopping_criterion do
 6:     P := Crossover (P, f, p_c)
 7:     P := Mutation (P, p_m)
 8:     f := Evaluate (P)
 9:     P := Select (P, f)
10:     t := t + 1
11: end while
```

As one example, the HCS can be integrated into the state-of-the-art MOEA SPEA2 [53], which we have used for our computations presented in the next section. To be more precise, we have used SPEA2 as the basis for our memetic algorithm, together with the SBX crossover operator [11] and a variable-wise mutation operator. A pseudocode can be found in Algorithm 5, for purpose of distinction we call this algorithm SPEA2–HCS in the sequel.

5 Results and Discussions

Here we present and discuss some numerical results for the HCS as well as for SPEA2-HCS in order to demonstrate the strength of both the standalone algorithm and the memetic strategy. The MOPs we have used here are listed in Table 3. All computations have been done using the programming language MATLAB[5].

[5] `https://www.mathworks.com`

Table 3. The MOPs under investigation in this work. Hereby, $\tilde{k} = n - k + 1$.

CONV1
$f_1(x) = (x_1 - 1)^4 + \sum_{i=2}^{n}(x_i - 1)^2$
$f_2(x) = \sum_{i=1}^{n}(x_i + 1)^2$
CONV2
$f_1(x) = \sum_{i=1}^{n}(x_i - 1)^2$
$f_2(x) = \sum_{i=1}^{n}(x_i + 1)^2$
$f_3(x) = \sum_{i=1}^{n}(x_i + (-1)^{n+1})^6$
ZDT4
$f_1(x) = x_1$
$f_2(x) = g(x)(1 - \sqrt{f_1/g(x)})$
$g(x) = 1 + 10(n-1) + \sum_{i=2}^{n}(x_i^2 - 10cos(4\pi x_i))$
$0 \leq x_1 \leq 1, \quad -5 \leq x_i \leq 5,\ i = 2, \ldots, n$
DTLZ1
$f_1(x) = \frac{1}{2}x_1x_2 \ldots x_{k-1}(1 + g(x))$
$f_2(x) = \frac{1}{2}x_1x_2 \ldots (1 - x_{k-1})(1 + g(x))$
$\vdots$
$f_{k-1}(x) = \frac{1}{2}x_1(1 - x_2)(1 + g(x))$
$f_k(x) = \frac{1}{2}(1 - x_1)(1 + g(x))$
$g(x) = 100\left[\tilde{k} + \sum_{i=k}^{n}(x_i - \frac{1}{2})^2 - \cos(20\pi(x_i - \frac{1}{2}))\right]$
$0 \leq x_i \leq 1,\ i = 1, \ldots, n$
DTLZ2
$f_1(x) = \cos(\frac{x_1\pi}{2})\cos(\frac{x_2\pi}{2}) \ldots \cos(\frac{x_{k-1}\pi}{2})(1 + g(x))$
$f_2(x) = \cos(\frac{x_1\pi}{2})\cos(\frac{x_2\pi}{2}) \ldots \sin(\frac{x_{k-1}\pi}{2})(1 + g(x))$
$\vdots$
$f_{k-1}(x) = \cos(\frac{x_1\pi}{2})\sin(\frac{x_2\pi}{2})(1 + g(x))$
$f_k(x) = \sin(\frac{x_1\pi}{2})(1 + g(x))$
$g(x) = \sum_{i=k}^{n}(x_i - \frac{1}{2})^2$
$0 \leq x_i \leq 1,\ i = 1, \ldots, n$
DTLZ3
$f_1(x) = \cos(\frac{x_1\pi}{2})\cos(\frac{x_2\pi}{2}) \ldots \cos(\frac{x_{k-1}\pi}{2})(1 + g(x))$
$f_2(x) = \cos(\frac{x_1\pi}{2})\cos(\frac{x_2\pi}{2}) \ldots \sin(\frac{x_{k-1}\pi}{2})(1 + g(x))$
$f_{k-1}(x) = \cos(\frac{x_1\pi}{2})\sin(\frac{x_2\pi}{2})(1 + g(x))$
$f_k(x) = \sin(\frac{x_1\pi}{2})(1 + g(x))$
$g(x) = 100\left[\tilde{k} + \sum_{i=k}^{n}(x_i - \frac{1}{2})^2 - \cos(\alpha\pi(x_i - \frac{1}{2}))\right]$
$\alpha = 20$
$0 \leq x_i \leq 1,\ i = 1, \ldots, n$
DTLZ3*
same as DTLZ3 but with $\alpha = 2$

Algorithm 5 SPEA2–HCS

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**

5.1 HCS as Standalone Algorithm

Since the two variants of the HCS as described in Algorithm 1 (which we will denote by HCS1 in this section) and in Algorithm 3 (denoted by HCS2) have 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 (see also discussion in Section 3.1): the HCS—i.e., both variants—is started as described above. If the current iterate x_p enough to $\mathcal{P}$ such that the sidestep procedure can start (taking $N_s = 5$), first improvements according to f_1 are sought (leading to a 'left up' movement from $F(x_p)$ along the Pareto front). If no improvements according to f_1 can be obtained, an analogue 'right down' movement is performed starting again from x_p. This is intended so 'screen' the entire connected component of $\mathcal{P}$ which is near to x_p.
However, since this orientation is not needed within the use of a MOEA because in that case only few iterates are being computed from a given starting point, these modifications are only done within this subsection.

In the following we will test HCS1 and HCS2 on a convex model (i.e., a model which does not contain local minima where the local search can get stuck) and we will investigate both the unconstrained and the constrained case. Then we will consider a multi-modal and constrained model (ZDT4).
Consider the MOP CONV1. The Pareto set of this model which is equal to $\mathcal{P}$ is located within $[-1, 1]^n$. First, we turn our attention to the unconstrained case: Figure 2 shows two results obtained by the modified algorithms HCS1 and HCS2 with dimension $n = 10$ and domain $Q = [-5, 5]^{10}$. In both cases the same starting point x_0 has been chosen. Since $\mathcal{P}$ is located within Q, no constraint handling techniques had to be applied in order to generate the sequence. For HCS1 a total of 1099 function calls had to be spent in order to get this result. For HCS2, 107 function calls, 60 evaluations of the Jacobian and 192 evaluations of the Hessian were required. It is obvious that due to the different requirements of the algorithms a quantitative comparison is hardly possible. On the other hand, Figure 2 shows some qualitative differences as anticipated from the design of the different algorithms: HCS2 converges faster (in this case four iterates were needed to reach $\mathcal{P}$ while HCS1 needed 23 iterations) and the nondominated

front is better distributed compared to the results obtained by the gradient-free version HCS1. However, both results are satisfying since both nondominated fronts represent a good approximation of the Pareto front with reasonable effort.

Next we consider the constrained case. Figure 3 shows a numerical result from the HCS1 where we have used dimension $n = 2$ and for the domain $Q = [0.5, 1.5] \times [1, 2]$. The Pareto set is given by $P_Q = [0.5, 1] \times \{1\}$ and thus included in the boundary of Q. The figures show that also in this case the HCS1 is capable of approaching the solution set, and moving along it further on. However, a total of 997 function calls had to be spent in this setting, that is, more in comparison to the unconstrained case (note that the dimension of the model is much lower in the latter case).

Finally, we consider the problem ZDT4, which is a highly nonlinear and multimodal model. Figure 4 shows two results in image space for two different initial solutions $x_0, z_0 \in Q = [0, 1] \times [-5, 5]^9$ and for the two variants of the HCS. As anticipated, the results for both algorithms and starting points differ significantly since the HCS is a local strategy and ZDT4 contains many local Pareto fronts. However, both procedures also in this case are able to explore a part of the local Pareto front which is located 'near' to the image of the initial solution.

5.2 SPEA2 Coupled with HCS

Here we make a comparison of the classical SPEA2 algorithm and SPEA2–HCS as presented in the previous section in order to demonstrate the possible benefit of the memetic strategy. Since we have two variants of the HCS we therefore have two MEMOEAs namely SPEA2–HCS1 where Algorithm 1 is used in combination with SPEA2 and SPEA2–HCS2. Since we are dealing in this section with MOPs where the Pareto set is located at the boundary of the domain, we have used a modification of Algorithm 3 which acts just as a hill climber. That is, the search along the Pareto set is not performed (the value $\epsilon_{\mathcal{P}}$ is set to 0).

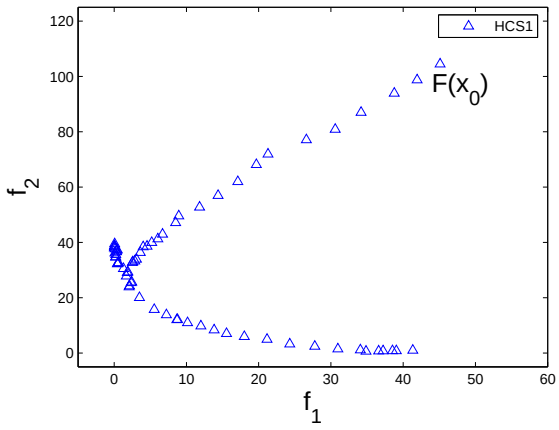
In order to evaluate the performance of the algorithms we have used the following three indicators (see [49] and [40]):

$$\begin{aligned}
&\text{Generational Distance} \quad : \quad GD = \tfrac{1}{n}\sqrt{\textstyle\sum_{i=1}^{n} \delta_i^2} \\
&\text{Efficient Set Space} \quad : \quad ESS = \sqrt{\tfrac{1}{n-1}\textstyle\sum_{i=1}^{n}(d_i - \overline{d})^2} \\
&\text{Maximal Distance} \quad : \quad MD = \max_{\substack{i,j=1,\ldots,n \\ i \neq j}} d_{ij}
\end{aligned}$$

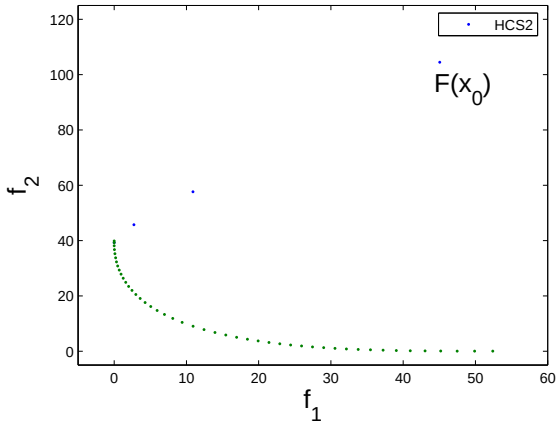
Hereby, δ_i denotes the minimal Euclidean distance from the image $F(x_i)$ of a solution $x_i, i = 1, \ldots, n$, to the true Pareto front, and

$$d_i := \min_{\substack{j=1,\ldots,n \\ i \neq j}} d_{ij} \quad \text{and} \quad \overline{d} := \frac{1}{n}\sum_{i=1}^{n} d_i, \tag{24}$$

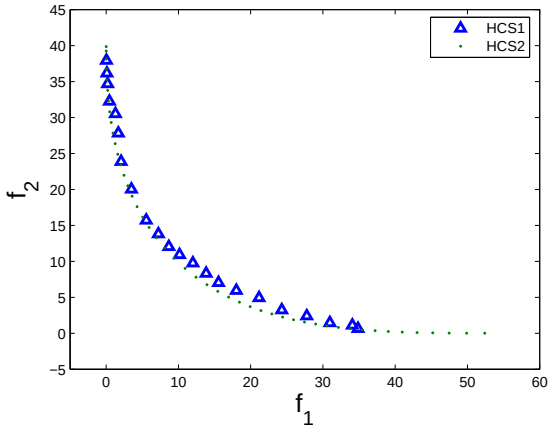
where d_{ij} is the Euclidean distance between $F(x_i)$ and $F(x_j)$. In the multiobjective optimization framework, there are in general three goals [52]: (i) the



(a) Solution HCS1

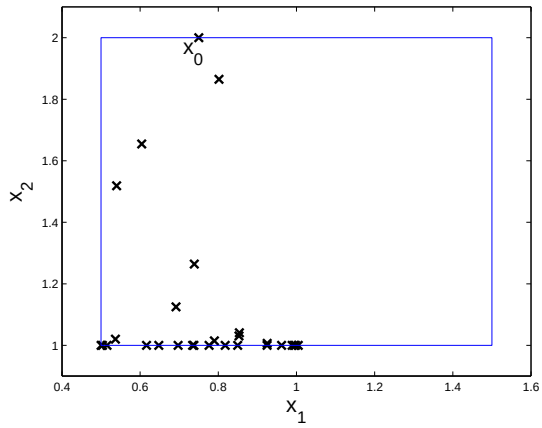


(b) Solution HCS2

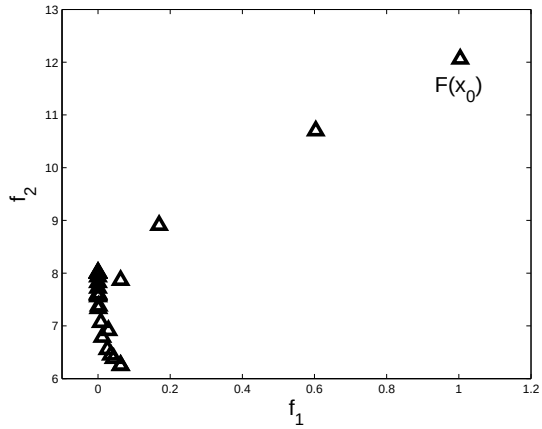


(c) Comparison Nondominated Fronts

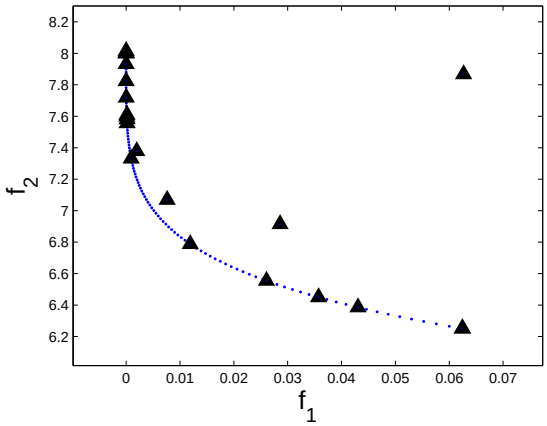
Fig. 2. Numerical result of HCS for MOP CONV1 with $Q = [-5, 5]^{10}$ in objective space (unconstrained case).



(a) Parameter Space

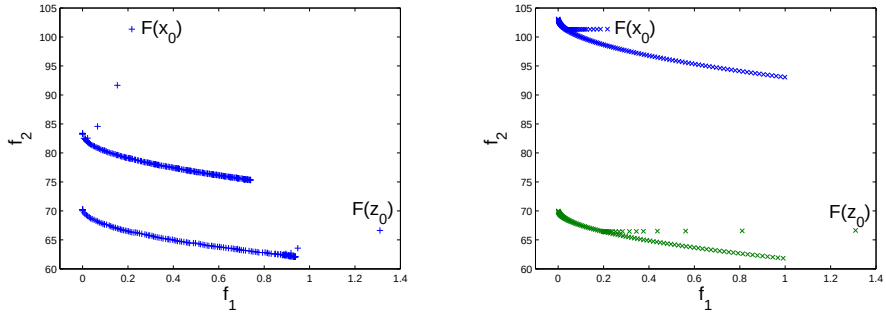


(b) Image Space



(c) Zoom Image Space and Pareto Front

Fig. 3. Numerical result of HCS1 for MOP CONV1 with $Q = [0.5, 1.5] \times [1, 2]$ (constrained case).



(a) Result HCS1 (b) Result HCS2

Fig. 4. Numerical result of HCS1 and HCS2 for MOP ZDT4 in objective space for two initial solutions x_0 and z_0.

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.

Crucial for the performance of the memetic strategies is certainly the choice of the probability p_{HCS} for the application of HCS. In general, there are two factors which influence the 'right' choice of p_{HCS}: the number of local Pareto fronts of a given MOP (i.e., roughly speaking, the problem complexity) and the total number of function evaluations available. In case the number of local Pareto fronts is high, a high probability for the use of HCS may lead to premature convergence. In addition, since the HCS requires relatively many function calls a high value of p_{HCS} may lead to the fact that the available budget of function calls is spent too fast, i.e., before a satisfying approximation is obtained. On the other hand, a low value of p_{HCS} may make the operator ineffective. Figure 5 shows one example for the evolution of the GD indicator which measures the convergence corresponding to different values of p_{HCS} for the DTLZ3 problem. Hereby we have run SPEA2 for 50 generations and have used the final population as the initial population for SPEA2–HCS2, which was then run for another 50 generations and with different values of p_{HCS}. Note that in all cases there is a noticeable decay in the value of GD in the beginning (iteration one to ten), but 'stabilizes' later on at different values. The worst results in this example are obtained for $p_{HCS} = 1$ and $p_{HCS} = 0.02$ which conforms with the above discussion since DTLZ3 is a multi-modal MOP. Here we have chosen the trade-off value $p_{HCS} = 0.2$ based on this and other computations, however, the development of an adaptive strategy is a task for future investigation. Table 4 displays all the parameters which have been used for both SPEA2 and SPEA2–HCS. Since at the beginning of the algorithm's execution, typically a global search is more ef-

fective than a local one, we have started the HCS after 75 % of the total number of generations (i.e., we have set $p_{HCS} = 0$ in the first generations).

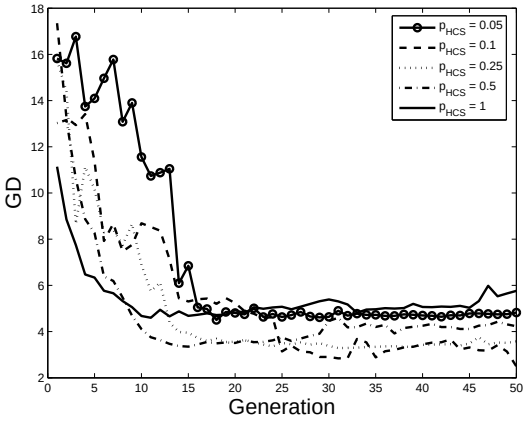


Fig. 5. Results of SPEA2–HCS2 on DTLZ3 using different values of p_{HCS}.

Table 4. Parameters for SPEA2 and SPEA2–HCS: N_{pop} and N_a denote the population size and the maximal cardinality of the archive, and p_c, p_m, p_{HCS} denote the probabilities for crossover, mutation and local search (HCS), respectively.

Parameter	**Value**
N_{pop}	100
N_a	100
p_c	0.8
p_m	0.01
p_{HCS}	0.2

Table 5 shows a comparison for some DTLZ test functions (see [13] and Table 3), 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 SPEA2–HCS (both variants) achieves better values than SPEA2 for all three indicators. For the generational distance, which is an indicator for convergence, the reduction is often near to 50%. Note that in addition the CPU time was significantly less with the SPEA2–HCS algorithms in all tests: since the number of function calls was fixed and since the local search operator required a certain

amount of it, the number of generations was much less for SPEA2–HCS, and thus, less time had to be spent for the update process of the archives. Another interesting observation is that the performance of SPEA2–HCS1 and SPEA2–HCS2 is nearly equal (since the models are given in analytical form, we have for simplicity counted one derivative call as one function call). This indicates that the gradient information which allows for a more exact search (locally) does not give an advantage over HCS1. The reason might be that in a combination of the HCS with a MOEA —which is by nature not exact—the exactness of the local search does not play an influential role.
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 SPEA2–HCS. The parameter α 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. 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*.
Finally, we consider MOP CONV2. Table 5 shows a result for dimension $n = 10$ and for a budget of 10,000 function calls. Since CONV2 is a smooth convex model the local search increases the overall performance significantly. Best values are obtained for SPEA2–HCS2, i.e., the memetic strategy which involves gradient information.
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 MOEAs) 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 MOEAs 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.

6 Conclusions and Future Work

We have proposed a novel point-wise iterative search procedure, the Hill Climber with Sidestep (HCS), which is designed for the local search of a given multi-objective optimization problem. The HCS is intended to be capable to moving both toward and along the set of (local) Pareto points. We have proposed two variants of the HCS, a gradient-free version (HCS1) and one which involves

Table 5. Numerical results on the DTLZ test functions and CONV2. For each DTLZ problem the number of function calls was fixed to 100,000 and to 10,000 for CONV2. The resulting mean value and standard deviation (in brackets) are presented for 30 test runs for each algorithm. We have chosen the dimension $n = 7$ for the DTLZ problems and $n = 10$ for CONV2. T denotes the CPU time.

	Indicators				
Problems		GD	ESS	MD	T
$DTLZ1$	SPEA2	8.20(3.33)	8.06(4.31)	291.18(96.46)	$3.83(1.33) \times 10^3$
	SPEA2–HCS1	4.66(1.33)	5.90(5.17)	**71.22**(38.49)	$2.46(0.42) \times 10^3$
	SPEA2–HCS2	**3.67**(1.64)	**1.99**(2.62)	107.78(59.16)	$\mathbf{1.82(0.31)} \times \mathbf{10^3}$
$DTLZ2$	SPEA2	0.0298(0.0094)	0.0501(0.0146)	2.4125(0.2907)	$1.40(0.85) \times 10^4$
	SPEA2–HCS1	**0.0044**(0.0022)	0.0306(0.0103)	**1.5691**(0.1018)	$1.21(0.15) \times 10^4$
	SPEA2–HCS2	0.0095(0.0054)	**0.0292**(0.0167)	1.7892(0.2024)	$\mathbf{8.70(0.39)} \times \mathbf{10^3}$
$DTLZ3$	SPEA2	11.23(7.93)	8.45(6.76)	300.47(185.25)	$3.09(0.43) \times 10^3$
	SPEA2–HCS1	**6.05**(3.69)	14.93(12.85)	**197.07**(99.78)	$1.84(0.25) \times 10^3$
	SPEA2–HCS2	10.71(5.58)	**6.98**(6.01)	338.98(141.27)	$\mathbf{1.12(0.26)} \times \mathbf{10^3}$
$DTLZ3^*$	SPEA2	18.37(15.02)	9.28(10.20)	427.86(318.84)	$3.55(0.12) \times 10^3$
	SPEA2–HCS1	**0.19**(0.30)	**0.93**(2.29)	**13.98**(24.08)	$2.65(0.42) \times 10^3$
	SPEA2–HCS2	0.47(1.39)	1.25(3.69)	27.04(61.54)	$\mathbf{2.08(0.18)} \times \mathbf{10^3}$
$CONV2$	SPEA2	58.37(55.57)	8.89(8.03)	434.59(327.30)	304.20(4.26)
	SPEA2–HCS1	28.43(21.61)	6.56(7.53)	271.00(241.44)	296.54(10.25)
	SPEA2–HCS2	**5.83**(4.49)	**5.25**(4.18)	**142.80**(77.10)	**198.43**(3.33)

gradient information (HCS2). Both algorithms are able to handle constraints of the model to some extend. Further, we have shown one possible way to integrate the local search procedure into a given MOEA leading to a memetic strategy. Finally, we have shown the efficieny of both the HCS as a standalone algorithm and a resulting memetic strategy on some examples. The results indicate that the HCS can be advantageous in particular in the cases where a classical MOEA gets 'stuck' (e.g., in the many-objective case). However, due to the nature of local search, the advantage of memetic strategies including the one presented in this paper decreases with increasing complexity of the underlying model. Thus, significant improvements of the memetic strategy compared to 'classical' MOEAs can be achieved in certain cases, but this cannot be expected in general.

For future work, there are some interesting topics which can be addressed to advance the present work. For instance, it would be desirable that HCS2 is also able to move along the Pareto set efficiently if this set is contained in the boundary of the domain which would allow for a more general use of the algorithm. Furthermore, the local strategy involves some design parameters which have so far to be adjusted by the user, but for which, an adaptive strategy would clearly be helpful. Another point is the integration of the HCS into a given MOEA. The particular coupling we have presented has the advantage that in general every MOEA can be taken and enhanced with little effort to become memetic. However, we expect that a more sophisticated interplay of the HCS and the MOEA

(e.g., by involving the information of the current population into the HCS) will increase the efficiency of the memetic strategy.

Acknowledgements

The fourth author gratefully acknowledges support from the CONACyT project no. 45683-Y.

References

1. Salem F. Adra, Ian Griffin, and Peter J. Fleming. Hybrid Multiobjective Genetic Algorithm with a New Adaptive Local Search Process. In Hans-Georg Beyer et al., editor, *2005 Genetic and Evolutionary Computation Conference (GECCO'2005)*, volume 1, pages 1009–1010, New York, USA, June 2005. ACM Press.
2. E. L. Allgower and K. Georg. *Numerical Continuation Methods*. Springer, 1990.
3. L. Armijo. Minimization of functions having Lipschitz-continuous first partial derivatives. *Pacific Journal of Mathematics*, 16:1–3, 1966.
4. Peter A.N. Bosman and Edwin 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.
5. Jürgen Branke and Sanaz Mostaghim. About Selecting the Personal Best in Multi-Objective Particle Swarm Optimization. In Thomas Philip Runarsson, Hans-Georg Beyer, Edmund Burke, Juan J. Merelo-Guervós, L. Darrell Whitley, and Xin Yao, editors, *Parallel Problem Solving from Nature - PPSN IX, 9th International Conference*, pages 523–532. Springer. Lecture Notes in Computer Science Vol. 4193, Reykjavik, Iceland, September 2006.
6. M. Brown and R. E. Smith. Directed Multi-Objective Optimisation. *International Journal of Computers, Systems and Signals*, 6(1):3–17, 2005.
7. Carlos A. Coello Coello, Gary B. Lamont, and David A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, New York, second edition, September 2007. ISBN 978-0-387-33254-3.
8. 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.
9. Leandro N. de Castro. *Fundamentals of Natural Computing*. Chapman and Hall/CRC, 2006.
10. Kalyanmoy Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Chichester, UK, 2001. ISBN 0-471-87339-X.
11. Kalyanmoy Deb and Ram Bhushan Agrawal. Simulated Binary Crossover for Continuous Search Space. *Complex Systems*, 9:115–148, 1995.
12. Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, April 2002.
13. Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable Test Problems for Evolutionary Multiobjective Optimization. In Ajith Abraham, Lakhmi Jain, and Robert Goldberg, editors, *Evolutionary Multiobjective Optimization. Theoretical Advances and Applications*, pages 105–145. Springer, USA, 2005.

14. M. Dellnitz, O. Schütze, and T. Hestermeyer. Covering Pareto sets by multilevel subdivision techniques. *Journal of Optimization Theory and Applications*, 124:113–155, 2005.
15. J.E. Dennis and R.B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, 1983.
16. Jörg Fliege and Benar Fux Svaiter. Steepest descent methods for multicriteria optimization. *Mathematical Methods of Operations Research*, 51(3):479–494, 2000.
17. Carlos M. Fonseca and Peter J. Fleming. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In Stephanie Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423, San Mateo, California, 1993. University of Illinois at Urbana-Champaign, Morgan Kauffman Publishers.
18. Tushar Goel and Kalyanmoy Deb. Hybrid Methods for Multi-Objective Evolutionary Algorithms. In Lipo Wang, Kay Chen Tan, Takeshi Furuhashi, Jong-Hwan Kim, and Xin Yao, editors, *Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning (SEAL'02)*, volume 1, pages 188–192, Orchid Country Club, Singapore, November 2002. Nanyang Technical University.
19. Nikolaus Hansen and Andreas Ostermeier. Completely Derandomized Self-adaptation in Evolution Strategies. *Evolutionary Computation*, 9(2):159–195, Summer 2001.
20. Ken Harada, Jun Sakuma, and Shigenobu Kobayashi. Uniform Sampling of Local Pareto-Optimal Solution Curves by Pareto Path Following and its Applications in Multi-objective GA. In Dirk Thierens, editor, *2007 Genetic and Evolutionary Computation Conference (GECCO'2007)*, volume 1, pages 813–820, London, UK, July 2007. ACM Press.
21. C. Hillermeier. *Nonlinear Multiobjective Optimization - A Generalized Homotopy Approach*. Birkhäuser, 2001.
22. Xiaolin Hu, Zhangcan Huang, and Zhongfan Wang. Hybridization of the Multi-Objective Evolutionary Algorithms and the Gradient-based Algorithms. In *Proceedings of the 2003 Congress on Evolutionary Computation (CEC'2003)*, volume 2, pages 870–877, Canberra, Australia, December 2003. IEEE Press.
23. Christian Igel, Nikolaus Hansen, and Stefan Roth. Covariance Matrix Adaptation for Multi-objective Optimization. *Evolutionary Computation*, 15(1):1–28, Spring 2007.
24. Hisao Ishibuchi and Tadahiko Murata. Multi-Objective Genetic Local Search Algorithm. In Toshio Fukuda and Takeshi Furuhashi, editors, *Proceedings of the 1996 International Conference on Evolutionary Computation*, pages 119–124, Nagoya, Japan, 1996. IEEE.
25. Andrzej Jaszkiewicz. 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, April 2003.
26. W. E. Karush. *Minima of functions of several variables with inequalities as side conditions*. PhD thesis, University of Chicago, 1939.
27. Joshua Knowles and David Corne. M-PAES: A Memetic Algorithm for Multiobjective Optimization. In *2000 Congress on Evolutionary Computation*, volume 1, pages 325–332, Piscataway, New Jersey, July 2000. IEEE Service Center.
28. Joshua Knowles and David 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.

29. 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.
30. M. Lozano, F. Herrera, N. Krasnogor, and D. Molina. Real-coded memetic algorithms with crossover hill-climbing. *Evolutionary Computation*, 12(3):273–302, 2004.
31. Kaisa Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston, Massachusetts, 1999.
32. Tadahiko Murata, Shiori Kaige, and Hisao Ishibuchi. Generalization of Dominance Relation-Based Replacement Rules for Memetic EMO Algorithms. In Erick Cantú-Paz et al., editor, *Genetic and Evolutionary Computation—GECCO 2003. Proceedings, Part I*, pages 1234–1245. Springer. Lecture Notes in Computer Science Vol. 2723, July 2003.
33. J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer, New York, 1999.
34. Una-May O'Reilly and Franz Oppacher. Hybridized crossover-based search techniques for program discovery. In *Proceedings of the 1995 World Conference on Evolutionary Computation*, volume 2, pages 573–578, Perth, Australia, 1995. IEEE Press.
35. Vilfredo Pareto. *Manual of Political Economy*. The MacMillan Press, 1971 (original edition in French in 1927).
36. Silvia Poles, Enrico Rigoni, and Tea Robič. MOGA-II Performance on Noisy Optimization Problems. In Bogdan Filipič and Jurij Šilc, editors, *Bioinspired Optimization Methods and Their Applications. Proceedings of the International Conference on Bioinspired Optimization Methods and their Applications, BIOMA 2004*, pages 51–62. Jožef Stefan Institute, Ljubljana, Slovenia, October 2004.
37. E. Rigoni and S. Poles. NBI and MOGA-II, two complementary algorithms for Multi-Objective optimization. In *Dagstuhl Seminar Proceedings 04461. Practical Approaches to Multi-Objective Optimization*, pages 1–22, 2005.
38. Hiroshi Satoh, Masayuki Yamamura, and Shigenobu Kobayashi. Minimal generation gap model for gas considering both exploration and exploitation. In *Iizuka '96*, volume 2, pages 494–497, Fukuoka, Japan, 1996.
39. 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.
40. Jason R. Schott. Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization. Master's thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, Massachusetts, May 1995.
41. Oliver Schuetze, Marco Laumanns, Emilia Tantar, Carlos A. Coello Coello, and El ghazali Talbi. Convergence of Stochastic Search Algorithms to Gap-Free Pareto Front Approximations. In Dirk Thierens, editor, *2007 Genetic and Evolutionary Computation Conference (GECCO'2007)*, volume 1, pages 892–899, London, UK, July 2007. ACM Press.
42. Oliver Schuetze, Gustavo Sanchez, and Carlos A. Coello Coello. A new memetic strategy for the numerical treatment of multi-objective optimization problems. In *2008 Genetic and Evolutionary Computation Conference (GECCO'2008)*, pages 705–712. ACM Press, Atlanta, USA, July 2008. ISBN 978-1-60558-131-6.
43. Oliver Schütze, Carlos A. Coello Coello, Sanaz Mostaghim, El-Ghazali Talbi, and Michael Dellnitz. Hybridizing Evolutionary Strategies with Continuation Methods

for Solving Multi-Objective Problems. *Engineering Optimization*, 40(5):383–402, May 2008.

44. Oliver Schütze, Sanaz Mostaghim, Michael Dellnitz, and Jürgen Teich. Covering Pareto Sets by Multilevel Evolutionary Subdivision Techniques. In Carlos M. Fonseca, Peter J. Fleming, Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele, editors, *Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003*, pages 118–132, Faro, Portugal, April 2003. Springer. Lecture Notes in Computer Science. Volume 2632.
45. Oliver Schütze, El-Ghazali Talbi, Carlos Coello Coello, Luis Vicente Santana-Quintero, and Gregorio Toscano Pulido. A Memetic PSO Algorithm for Scalar Optimization Problems. In *Proceedings of the 2007 IEEE Swarm Intelligence Symposium (SIS 2007)*, pages 128–134, Honolulu, Hawaii, USA, April 2007. IEEE Press.
46. Abhishek Sinha, Ying ping Chen, and David 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.
47. Massimiliano Vasile. A Behavior-based Meta-Heuristic for Robust Global Trajectory Optimization. In *2007 IEEE Congress on Evolutionary Computation (CEC'2007)*, pages 494–497, Singapore, 2007. IEEE Press.
48. Massimiliano Vasile and Marco Locatelli. A hybrid multiagent approach for global trajectory optimization. *Journal of Global Optimization*, 2008. (in press).
49. David A. Van Veldhuizen. *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.
50. Yong Wang, Zixing Cai, Guanqi Guo, and Yuren Zhou. Multiobjective optimization and hybrid evolutionary algorithm to solve constrained optimization problems. *IEEE Transactions on Systems, Man and Cybernetics Part B–Cybernetics*, 37(3):560–575, June 2007.
51. Elizabeth F. Wanner, Frederico G. Guimaraes, Ricardo H.C. Takahashi, and Peter J. Fleming. A Quadratic Approximation-Based Local Search Procedure for Multiobjective Genetic Algorithms. In *2006 IEEE Congress on Evolutionary Computation (CEC'2006)*, pages 3361–3368, Vancouver, BC, Canada, July 2006. IEEE.
52. Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195, Summer 2000.
53. Eckart Zitzler, Marco Laumanns, and Lothar 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*, pages 95–100, Athens, Greece, 2002.
54. Eckart Zitzler and Lothar Thiele. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, November 1999.