EVOLVE 2012
Mexico City, Mexico, August 07-09 2012
ISBN 978-2-87971-112-6 ISSN 2222 - 9434

THE HILL-CLIMBER WITH SIDESTEP FOR CONSTRAINED MULTI-OBJECTIVE OPTIMIZATION PROBLEMS

Gustavo Sánchez[a]

[a]Departmento de Procesos y Sistemas
Universidad Simón Bolívar
Venezuela
gsanchez@usb.ve

Abstract

Many researchers have hybridized Multi-Objective Evolutionary Algorithms (MOEAs) with local search techniques in order to obtain fast and reliable algorithms, capable of computing an accurate representation of the entire Pareto set in one single run. Recently, the Hill Climber with Sidestep (HCS) has been proposed: a point-wise iterative local search method, capable of moving both toward and along the Pareto set, depending on the location of the current iterate. However, one crucial drawback is that HCS was restricted to unconstrained models. In this paper, we investigate four different constraint handling techniques—penalty, barrier, backtracking, and gradient projection—with respect to their potential usage together with HCS. We first study new variants as standalone operators. Second, we integrate them into well-known algorithms: SPEA2 and NSGA-II. Numerical results on several benchmarks indicate the benefit of HCS as a local searcher also when applied to constrained problems.

1 Introduction

In the past decades, solving constrained engineering design problems via evolutionary algorithms has attracted increasing attention. In many of such problems, several objectives have to be considered concurrently leading to a *Multi-Objective Optimization Problem* (MOP). As a general example, two common goals are to maximize the quality of the product as well as 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 a MOP does in general not consist of one single solution but rather of an entire set of solutions. Evolutionary algorithms specialized to this kind of problems, *Mult-Objective Evolutionary Algorithms* (MOEAs), offer the advantage that a finite size representation of the entire solution set can be computed in one run. Further, MOEAs can be applied to a broad class of MOPs since they do not require hard assumptions on the objectives nor on the domain of the problem. Nevertheless, one important drawback of classical MOEAs is their high computational cost due to their relatively slow convergence (see, e.g., [1] for an explanation). To improve the overall performance, it can be advantageous in certain cases to hybridize the MOEA with a local search method, leading to a *memetic MOEA* (see, e.g., [8, 5, 2, 7, 13] and references therein).

Recently, the *Hill Climber with Sidestep* (HCS) has been proposed ([6]). It is a point-wise operator that allows to steer the search process both *towards and along* the Pareto set, depending on the location of the point to which it is being applied. The main features of HCS can be summarized as follows:

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

(b) If a given point x_0 is 'far away' from the Pareto set, the operator try to generate a new point x_1 which is better (i.e., $x_1 \prec x_0$). If x_0 is 'near' to the Pareto set, the operator try to generate a set of points along the Pareto set.

(c) The HCS can be easily 'integrated' into any MOEA (plug and play philosophy).

Hence, the HCS is able to explore a part of the (local) Pareto set starting from one single point. Moreover, it has been shown that it can be efficiently used as a local search operator within a MOEA ([6]). However, the main drawback of the previously proposed operator is that it is restricted to *unconstrained* MOPs which results in a severe handicap— particularly in the context of engineering design problems which are highly constrained.

In this paper, we empirically investigate the HCS with respect to its applicability to constrained MOPs. It is important to stress that the original idea behind HCS is based on the geometry of unconstrained MOPs, since it is based on some observations about descent and diversity cones of such problems (see [1]). Hence, it makes sense to consider first techniques that transform constrained into unconstrained problems such as penalty and barrier methods.

Further, we consider 'repair' strategies that generate feasible points from unfeasible points generated by the global search process. For this, we will consider the backtracking and the gradient projection method.

The remainder of this paper is organized as follows. In Section 2, we state some theoretical background and formulate the constrained MOPs under investigation. In Section 3, we consider different variants of the HCS resulting from each constraint handling technique (penalty, barrier, backtracking, and gradient projection) in order to get an impression of their own dynamic. In Section 4, we present numerical results of SPEA2 ([15]) and NSGA-II ([3]) together with their memetic variants. Finally, we draw conclusions in Section 5.

2 Background

In the following we consider constrained MOPs of the form

$$\begin{aligned} &\min_{x\in\mathbb{R}^n} F(x) \\ &s.t \\ &c_i(x) \geq 0, \quad i \in \{1,2,\ldots,m\} \end{aligned} \tag{1}$$

where F is defined as the vector of objective functions $F : Q \to \mathbb{R}^k$, $F(x) = (f_1(x),\ldots,f_k(x))$, and where each $f_i : Q \to \mathbb{R}$ is continuous. We define Q to be the set of points that satisfy the constraints (feasible set), that is, $Q = \{x | c_i(x) \geq 0, \quad i \in \{1,\ldots,m\}\}$. The active set $A(x) = \{i| \ c_i(x) = 0, \quad i \in \{1,2,\ldots,m\}\}$ at any feasible x is the set of indices corresponding to active inequalities. The optimality of a point $x \in Q$ is based on the concept of *dominance* [12].

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 $y \in Q$ is dominated *by a vector $x \in Q$ (in short: $x \prec y$) with respect to (1) if $F(x) \leq_p F(y)$ and $F(x) \neq F(y)$ (i.e. there exists a $j \in \{1,\ldots,k\}$ such that $f_j(x) < f_j(y)$), 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 Pareto optimal solutions is called the *Pareto set $\mathscr{P}_Q$*. The image $F(\mathscr{P}_Q)$ is called the *Pareto front*. In this study, we will consider the following two MOPs with linear constraints, with two and three objective functions respectively.

$$\text{Problem} \quad P_1 \quad : \quad \min_{x\in\mathbb{R}^n} \begin{pmatrix} (x_1-1)^4 + \sum_{i=2}^{n}(x_i-1)^2 \\ \sum_{i=1}^{n}(x_i+1)^2 \end{pmatrix}$$

$$s.t$$

$$0 \quad \leq \quad x_i \leq 5 \,,\, i \in \{1,2,3,\ldots,n\}$$

and

$$\text{Problem} \quad P_2: \qquad \min_{x\in\mathbb{R}^n} \begin{pmatrix} \|x-e_1\|_2^2 \\ \|x-e_2\|_2^2 \\ \|x-e_3\|_2^2 \end{pmatrix}$$

$$s.t$$

$$-10 \leq x_i \leq 10 \ , i \in \{1,2,3,\dots,n\}$$

$$x_1 + x_2 \leq 1$$

Hereby, e_i, $i = 1,2,3$, denotes the i-th unit vector. If the constraints are left out in problem P_1, then the Pareto set would form a curve connecting the points $a_1 = (-1,\dots,-1)$ and $a_2 = (1,\dots,1)$ ([11]). Hence, by construction, the Pareto set of P_1 can be divided into two parts: one part is contained in the inside of the domain Q, and one part is contained in the boundary ∂Q. Since P_1 is convex, both parts form one connected component. For P_2, the Pareto set consists of the simplex that has the e_i's as nodes, i.e., $\mathscr{P}_Q = S(e_1,e_2,e_3)$. Crucial is that the subset $S(e_1,e_2)$ of the Pareto set is located on the boundary of the domain.

3 HCS as standalone algorithm

In this section we investigate the applicability of the HCS to auxiliary MOPs derived from the original models P_1 and P_2 together with a constraint handling method. Here, we concentrate on the behavior of the HCS as standalone algorithm in order to see the effect of each constraint handling method on the operator dynamic. We denote by HCS1 the gradient-free version of the HCS and by HCS2 the variant using gradient information.

The HCS1 as proposed in [6] is described next. Given a point $x_0 \in Q$, a further point x_1 is chosen randomly from a neighborhood of x_0, say $\tilde{x}_1 \in B(x_0,r)$ where

$$B(x_0,r) := \{x \in \mathbb{R}^n : x_{0,i} - r_i \leq x_i \leq x_{0,i} + r_i \ \forall i = 1,..,n\}, \tag{2}$$

and where $r \in \mathbb{R}^n_+$ is a given (problem depending) radius. If $\tilde{x}_1 \prec x_0$, then $\nu := \tilde{x}_1 - x_0$ can be considered as a descent direction at x_0, and along it a 'better' candidate can be searched, for example via line search methods. 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 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, it is suggested to use the accumulated information by taking the average search direction

$$\nu_{acc} = \frac{1}{N_{nd}} \sum_{i=1}^{N_{nd}} \frac{\tilde{x}_i - x_0}{\|\tilde{x}_i - x_0\|}, \tag{3}$$

since with this direction one expects the maximal diversity (or 'sidestep') among the available directions.

Now, the HCS2 variant is described. Given a point x_0, the descent direction d is calculated as (see [10])

$$d = -\sum_{i=1}^{k} \alpha_i^* \nabla f_i(x_0) \tag{4}$$

where $\alpha^* \in \mathbb{R}^k$ is solution of the following quadratic optimization problem

$$\begin{aligned} &\min_{\alpha\in\mathbb{R}^k} \left\| \textstyle\sum_{i=1}^k \alpha_i \nabla f_i(x_0) \right\|_2^2 \\ &\text{subject to} \sum_{i=1}^{k} \alpha_i = 1, \alpha_i \geq 0 \end{aligned} \tag{5}$$

If the following condition is satisfied:

$$\left\| \sum_{i=1}^{M} \alpha_i^* \nabla f_i(x_0) \right\|_2^2 \geq \varepsilon_P \tag{6}$$

i.e., if the square of the norm of weighted gradients is larger than a given threshold, the candidate solution can be considered to be away from $\mathscr{P}_Q$, and thus it makes sense to seek for a dominating solution. For this, the descent direction d can be taken together with a suitable step size control.

If the value of the term in (6) is less than ε_P, this indicates that x_0 is already in the vicinity of $\mathscr{P}_Q$. In that case, one can lean elements from multiobjective continuation (see [4]) to perform a search along $\mathscr{P}_Q$, trying to accomplish 'sidesteps'.

3.1 HCS Using Augmented Logarithmic Barrier Function

An important class of methods for constrained optimization seeks solutions by replacing the original problem by a sequence of unconstrained sub-problems [9]. As first method considered here, we choose the augmented logarithmic barrier that transforms an objective f_i into

$$\tilde{f}_i^{(LB)}(x) := f_i(x) + \mu \sum_{i=1}^{m} log(c_i(x)), \tag{7}$$

where $\mu \in \mathbb{R}_+$ is the barrier parameter. For these auxiliary objectives it holds

- they approach infinity when the iterate x_i approaches the boundary of the feasible set ∂Q;
- they are as smooth as the original objective f_i inside Q.

As an example, the problem P_1 is tranformed into the following auxiliary MOP (note that all objectives are penalized by the same quantity.):

$$\text{Problem} \quad P_3 : \min_{\mathbf{x} \in \mathbb{R}^n} \begin{pmatrix} (x_1-1)^4 + \sum_{i=2}^{n}(x_i-1)^2 - \mu \sum_{i=1}^{n} \log(x_i) - \mu \sum_{i=1}^{n} \log(5-x_i) \\ \sum_{i=1}^{n}(x_i-1)^2 - \mu \sum_{i=1}^{n} \log(x_i) - \mu \sum_{i=1}^{n} \log(5-x_i) \end{pmatrix}$$

Since HCS1 and HCS2 as described in the previous section have no orientation in the search along the Pareto set, for bi-objective models we propose to generate a trajectory of points as follows: the HSC is applied to a first point x_0 trying to find a sequence of descent direction. When the first side-step is performed, this indicates that the current iteration is already near to the (local) Pareto set, and the current point is stored in x_p.

In further side-steps, candidates will only be accepted if the first objective gets decreased. If no improvements can be achieved according to f_1 within a given number of sidesteps, the HCS 'jumps' back to x_p, and a similar process is started but aiming for improvements according to f_2.

For $n = 2$, figures 1 and 2 show one trajectory generated by HCS for $\mu = 1$ and $\mu = 0.1$ respectively. In both cases the same initial point has been chosen. Note that for $\mu = 0.1$ HCS achieves a better approximation as for the first case ($\mu = 1$).

It could also be interesting to investigate what happens if we increase the number of decision variables. For $n = 30$, figures 3 and 4 show the resulting trajectories for $\mu = 0.1$ and $\mu = 0.01$. Again, for lower values of μ we get better approximations.

If gradient information is at hand, one can use HCS2 which yields more accurate iteration steps both toward and along the Pareto set. This can be observed in Figures 5 and 6.

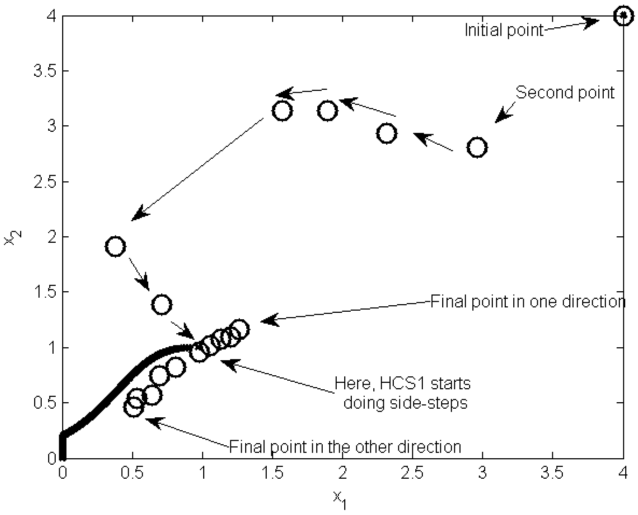


Figure 1: Sample trajectory (in variable space) generated by HCS1 on problem P_3 for $n = 2$ and barrier constant $\mu = 1$.The black line indicates the Pareto set of the original problem P_1.

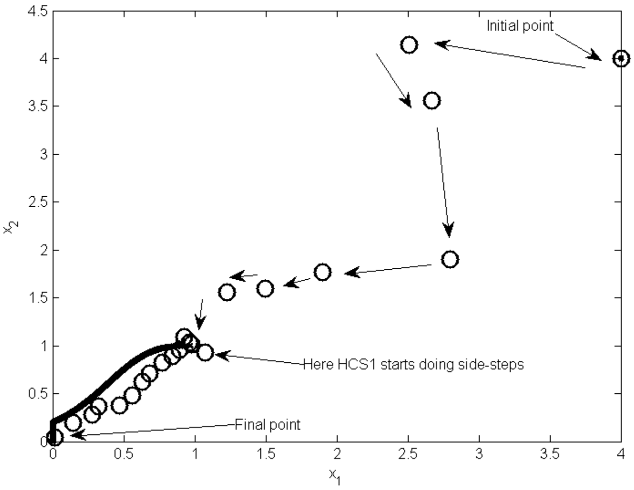


Figure 2: Sample trajectory (in variable space) generated by HCS1 on problem P_3 for $n = 2$ and barrier constant $\mu = 0.1$. The black line indicates the Pareto set of the original problem P_1.

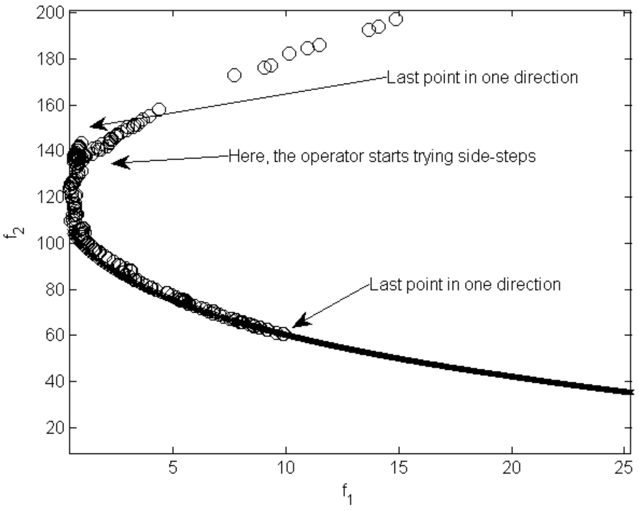


Figure 3: Sample trajectory (in objective space) generated by HCS1 on problem P_3 for $n = 30$ and barrier constant $\mu = 0.1$. The black line indicates the Pareto front of the original problem P_1.

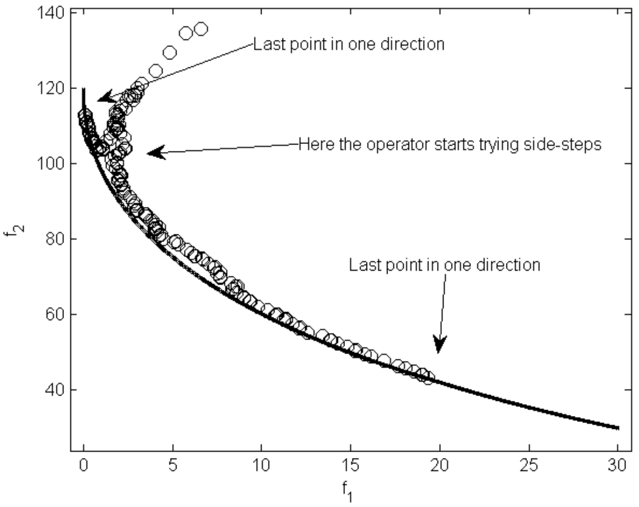


Figure 4: Sample trajectory (in objective space) generated by HCS1 on problem P_3 for $n = 30$ and barrier constant $\mu = 0.01$. The black line indicates the Pareto front of the original problem P_1.

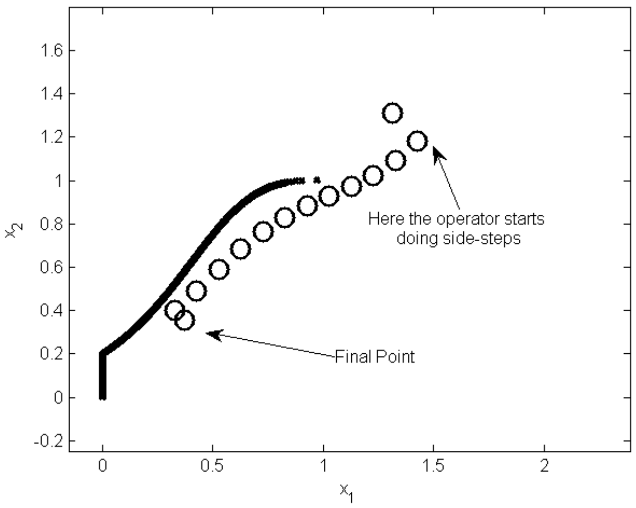


Figure 5: Sample trajectory (in variable space) generated by HCS2 on problem P_3 for $n = 2$ and $\mu = 1$

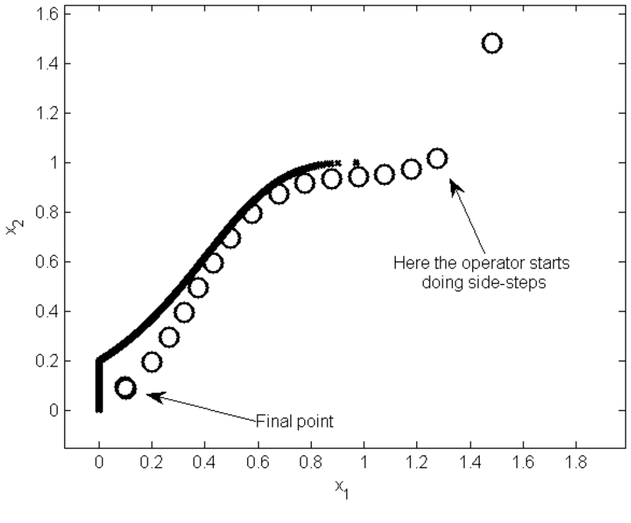


Figure 6: Sample trajectory (in variable space) generated by HCS2 on problem P_3 for $n = 2$ and $\mu = 0.1$

3.2 Augmented Constant-Penalty Function

Next, we consider the augmented constant-penalty method that transforms an objective f_i into

$$\tilde{f}_i^{(CP)}(x) := f_i(x) + M(x), \tag{8}$$

where

$$M(x) = \begin{cases} 0 \text{ if } x \in Q \\ 10.000 \text{ if } x \notin Q \end{cases}$$

As for problem P_1, the resulting unconstrained MOP reads as:

$$\text{Problem} \quad P_4 : \min_{\mathbf{x} \in \mathbb{R}^n} \begin{pmatrix} (x_1 - 1)^4 + \sum_{i=2}^{n} (x_i - 1)^2 + M(x) \\ \sum_{i=1}^{n} (x_i - 1)^2 + M(x) \end{pmatrix}$$

Figures 7 and 8 show some sample trajectories for $n = 2$ and $n = 30$. In general, when no gradient information is available, the HCS seems to produce better results using an augmented constant-penalty function: the generated points are closer to true solutions and cover a greater section.

3.3 Backtracking to Feasible Region

Next to the methods that transform a constrained objective into an unconstrained one, we consider in the following two repair mechanisms: a backtracking procedure that allows to find a feasible point in the current line search, and the gradient projection method.

In order to track back from the current iterate x_1 to the feasible set, we propose to proceed analogously to the well-known bisection method for root finding (see Algorithm 1 for one possible realization): let $in_0 := x_0 \in Q$ and $out_0 := x_1 \notin Q$ and $m_0 := in_0 + 0.5(out_0 - in_0) = x_0 + \frac{h_0}{2} v$.

Algorithm 1 Backtracking to Feasible Region

```
Require: $x_0 \in Q, x_1 = x_0 + h_0 v \notin Q, tol \in \mathbb{R}_+$
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 $\tilde{x} := in_i$
```

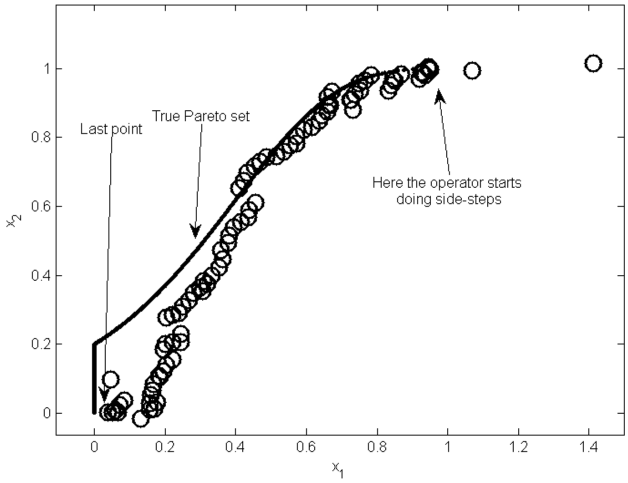


Figure 7: Sample trajectory (in variable space) generated by HCS1 on problem P_4 for $n = 2$ parameters

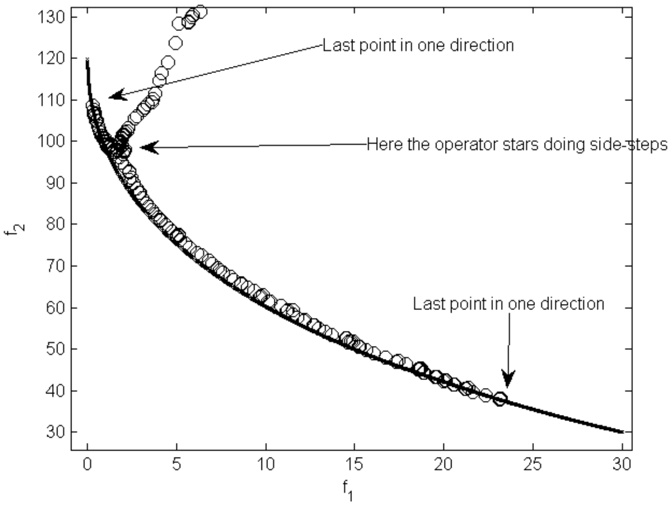


Figure 8: Sample trajectory (in objective space) generated by HCS1 on problem P_4 for $n = 30$ parameters

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 index $i_0 \in \mathbb{N}$ such that $\|out_{i_0} - in_{i_0}\|_\infty \leq tol$, obtaining a point in_{i_0} with maximal distance tol to ∂Q. Note that by this procedure no additional function evaluation has to be spent.

Figures 9 and 10 show sample trajectories generated by the above backtracking method. This method seems to produce the same quality of approximations with respect to the two previous sections.

3.4 Gradient projection method

When constraints are simple in form—e.g., when Q is given by box constraints—the gradient projection method is designed to make rapid changes to the active set. We follow [9] and define the projection of an arbitrary point x onto the feasible region as follows: The i-th component is given by

$$P(x,l,u)_i = \begin{cases} l_i \text{ if } x_i < l_i \\ x_i \text{ if } l_i \leq x_i \leq u_i \\ u_i \text{ if } x_i > l_i \end{cases}$$

The linear path $x(t)$ starting at the reference point x^0 and obtained by projecting the direction d at x^0 onto the feasible region is thus given by

$$x(t) = P(x^0 - td, l, u)$$

The values of t for which each component reaches its bound along the chosen direction $-d$ are given by:

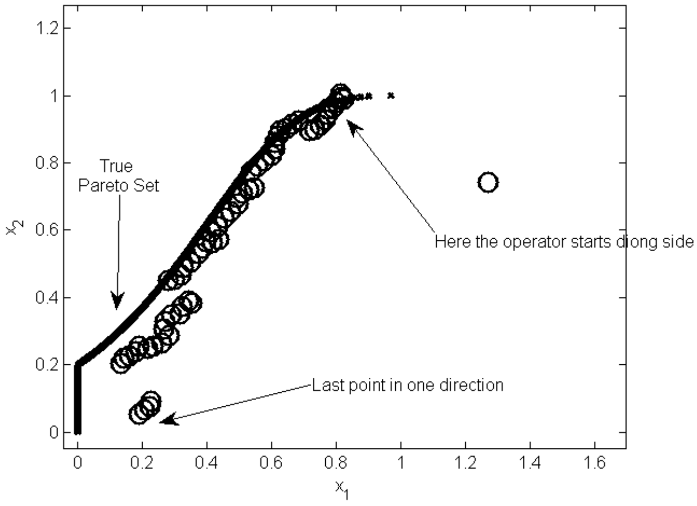
$$\bar{t}_i = \begin{cases} \frac{x_i^0 - u_i}{d_i} \text{ if } d_i < 0 \\ \frac{x_i^0 - l_i}{d_i} \text{ if } d_i > 0 \\ \infty \text{ otherwise} \end{cases}$$

Thus, the components of $x(t)$ are

$$x(t) = \begin{cases} x_i^0 - td_i \text{ if } t < \bar{t}_i \\ x_i^0 - \bar{t}_i d_i \text{ otherwise} \end{cases}$$

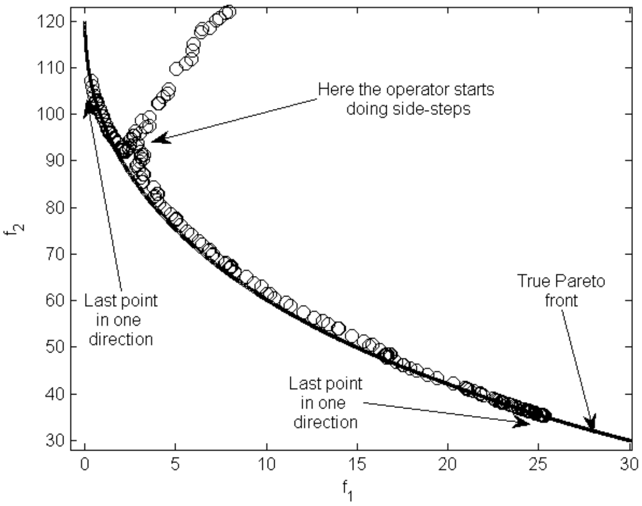
Figure 11 shows a sample trajectory for problem P_1 generated by the gradient projection method for $n = 2$. For problem P_2 (three objectives), Figure 12 also shows the trajectory generated by gradient projection when $n = 2$. In this case, we do not have only box constraints, so we have to use a different projection matrix (see [9] for more details).

Note that in figure 12 the first point is (-1,-1) which is feasible but does not belong to Pareto set. We have a 'descent step' until (-0.34,-0.34) and there the operator starts doing side-steps. Given that in this case we have more than 2 'diversity directions' (6 diversity cones in fact) the next point is randomly selected from the non-dominated generated in the previous iteration. A interesting fact worth to observe in this figure is that the operator can follow the active restriction line. Note also that the HCS has trouble to find its path when it encounters a vertex or a region where the Pareto set changes abruptly. To close this section, Table 1 displays the number of function evaluations that has been allowed to obtain each figure.



1

Figure 9: Sample trajectory (in variable space) generated by HCS1 on problem P_1 using the backtracking method for $n = 2$ decision variables



1

Figure 10: Sample trajectory (in objective space) generated by HCS1 on problem P_1 using the backtracking method for $n = 30$ decision variables

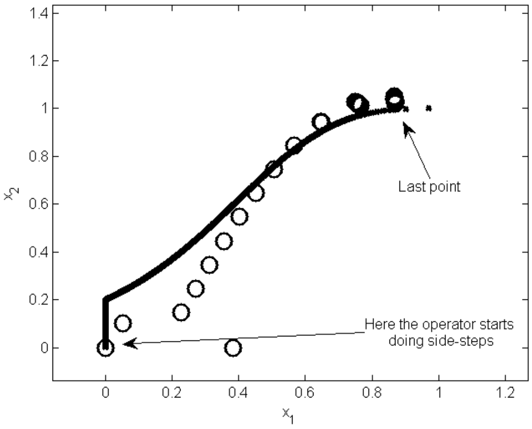


Figure 11: Trajectory generated by HCS2 using the gradient projection method. $n = 2$

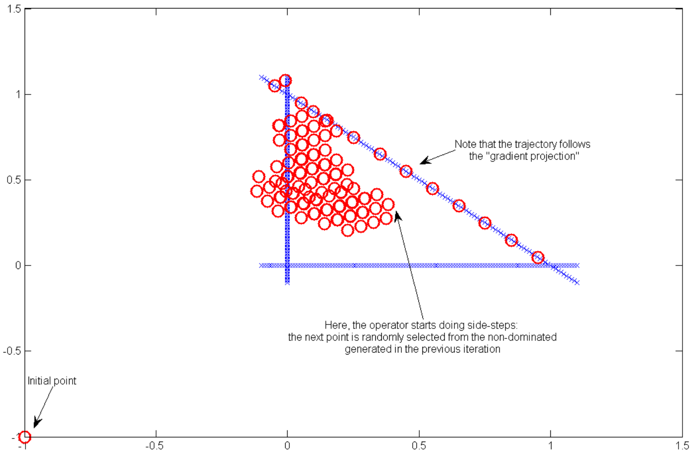


Figure 12: For problem P_2 and $n = 2$, these are points generated by HCS2 using the gradient projection method

Table 1: Number of function evaluations for each figure

Figure	N° Eval
Fig 1.1	2000
Fig 1.2	2000
Fig 1.3	5000
Fig 1.4	5000
Fig 1.5	200
Fig 1.6	200
Fig 1.7	2000
Fig 1.8	5000
Fig 1.9	2000
Fig 1.10	5000
Fig 1.11	200
Fig 1.12	200

4 HCS as Local Searcher within a MOEA

The sample trajectories shown above were intended to give an impression about the working principle of the resulting HCS variants. In all that cases, the HCS has been used as standalone algorithm. This is, however, only of limited value since all local search procedures (as the HCS) fail to find the global Pareto set in general (e.g., for multi-modal models). More important—at least in the context of global optimization—is the capability of a local method to improve the overall performance of a global search procedure. We investigate the latter by integrating the new HCS variants into the well-known and widely used SPEA2 and NSGA-II.

For the integration of HCS into a MOEA we follow the suggestions given in [6]. We denote the resulting hybrids by:

- SPEA2-HCS1c-1: SPEA2 together with the variant of HCS1 using augmented logarithmic barrier function.
- SPEA2-HCS1c-2: SPEA2 together with the variant of HCS1 using augmented constant-penalty function.
- SPEA2-HCS1c-3: SPEA2 together with the variant of HCS1 using back-tracking.
- SPEA2-HCS2c-1: SPEA2 together with the variant of HCS2 using augmented logarithmic barrier function.
- SPEA2-HCS2c-2: SPEA2 together with the variant of HCS2 using gradient projection.

Analog for the NSGA-II hybrids. Table 2 displays the parameters which have been used for tests. The local search is applied once each ten generations to individuals in the archive with a probability $P_{LS} = 0.1$.

To evaluate the performance of each algorithm we have used the following three indicators (see also [14]):

$$\begin{array}{lll} \text{Generational Distance} & : & GD = \frac{1}{n}\sqrt{\sum_{i=1}^{n} \delta_i^2} \\[2ex] \text{Efficient Set Space} & : & ESS = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(d_i - \overline{d})^2} \\[2ex] \text{Maximal Distance} & : & MD = \max\limits_{\substack{i,j=1,\dots,n \\ i\neq j}} d_{ij} \end{array}$$

Table 2: 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}	0.1

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

$$d_i := \min_{\substack{j=1,\ldots,n \\ i \neq j}} d_{ij} \quad \text{and} \quad \overline{d} := \frac{1}{n} \sum_{i=1}^{n} d_i, \tag{9}$$

where d_{ij} is the Euclidean distance between $F(x_i)$ and $F(x_j)$.

The numerical results in Table 3 show that in all cases the hybrids achieve better values than their base MOEA according to GD, which is a convergence indicator. The number of function calls was fixed to 30.000, and each algorithm was run 30 times. The resulting mean value and standard deviation (in brackets) are presented. It is also possible to observe that there is no statistical difference between the three versions of the operator which use no gradient information.

Table 4 shows promising results for problem P_2: the values of GD obtained by pure MOEAs are between 100 and 1000 times greater with respect to those obtained by the hybrid variant using gradient projection.

Table 3: Results for problem P_1

	Indicators		
Algorithms	*GD*	*ESS*	*MD*
SPEA2	1.6052(0.1494)	0.2757(0.1293)	76.4997(4.4145)
SPEA2-HCS1c-1	1.0645(0.0845)	0.4248(0.1110)	74.8373(3.1295)
SPEA2-HCS1c-2	1.0351(0.0811)	0.3936(0.0637)	75.1354(3.1651)
SPEA2-HCS1c-3	1.0484(0.0662)	0.4077(0.0817)	73.4754(3.5484)
SPEA2-HCS2c-1	0.7875(0.0907)	0.2815(0.0684)	79.5214(4.6603)
SPEA2-HCS2c-2	0.5203(0.0280)	0.4439(0.0966)	86.3232(2.8784)
NSGA-II	0.9743(0.1932)	0.1532(0.0304)	34.3753(5.2383)
NSGA-II-HCS1c-1	0.6940(0.0810)	0.2160(0.0683)	45.8302(5.6351)
NSGA-II-HCS1c-2	0.6611(0.0510)	0.2550(0.1088)	47.4220(6.5028)
NSGA-II-HCS1c-3	0.6691(0.0669)	0.2184(0.0512)	45.3911(5.6244)
NSGA-II-HCS2c-1	0.5679(0.0376)	0.2644(0.0784)	52.4182(7.2925)
NSGA-II-HCS2c-2	0.5254(0.0349)	0.3380(0.1290)	66.5242(16.2378)

Table 4: Results for problem P_2

	Indicators		
Algorithms	*GD*	*ESS*	*MD*
SPEA2	189.8874(51.4709)	0.01(0.02)	0.1800(0.6068)
SPEA2-HCS2c-2	0.1514(0.0280)	0.0436(0.0694)	2.3306(1.0993)
NSGA-II	4.7646(2.3375)	0.0007(0.0018)	0.0777(0.0847)
NSGA-II-HCS2c-2	0.0068(0.0005)	0.0396(0.0046)	2.7634(0.1505)

5 Conclusions and Future Work

In this paper, we have investigated the HCS with respect to its applicability to handle constrained MOPs, based on different constraint handling techniques. Numerical results have shown the operator is capable of handling constrained problems to a certain extent, keeping its original features (i.e. Pareto set exploration and plug and play philosophy). The new hybrids have outperformed their base MOEAS significantly in all cases. For future work, we intend to make a more rigorous investigation of HCS when applied to constrained models, including a more thorough comparison to other methods.

References

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

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

[3] 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*.

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

[5] J. Knowles and D. Corne and. *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.

[6] A. Lara, G. Sanchez, C. Coello-Coello, and O. Schuetze. HCS: A New Local Search Strategy for Memetic Multiobjective Evolutionary Algorithms. *IEEE Trans. on Evol. Comput.*, 14:112–132, 2010.

[7] C.-Y. Lin and W.-H.Wu. Self-organizing adaptive penalty strategy in constrained genetic search. *Struct Multidisc Optim*, 26:417–428, 2004.

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

[9] J. Nocedal and S. Wright. *Numerical optimization*. Springer, 1999.

[10] 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.

[11] O. Schütze, A. Dell'Aere, and M. Dellnitz. On continuation methods for the numerical treatment of multi-objective optimization problems. In Jürgen Branke, Kalyanmoy Deb, Kaisa Miettinen, and Ralph E. Steuer, editors, *Practical Approaches to Multi-Objective Optimization*, number 04461 in Dagstuhl Seminar Proceedings. Internationales Begegnungs- und Forschungszentrum (IBFI), Schloss Dagstuhl, Germany, 2005. <http://drops.dagstuhl.de/opus/volltexte/2005/349>.

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

[13] Yong Wang, Zixing Cai, Yuren Zhou, and Zhun Fan. Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique. *Struct Multidisc Optim*, 37:395–413, 2009.

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

[15] 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.