Evolutionary Continuation Methods for Optimization Problems

Oliver Schuetze
CINVESTAV-IPN
Computer Science Department
Mexico City 07360, MEXICO
schuetze@cs.cinvestav.mx

Adriana Lara
CINVESTAV-IPN
Computer Science Department
Mexico City 07360
MEXICO
alara@cs.cinvestav.mx

Carlos A. Coello Coello
CINVESTAV-IPN
Computer Science Department
Mexico City 07360, MEXICO
ccoello@cs.cinvestav.mx

ABSTRACT

In this paper we develop evolutionary strategies for numerical continuation which we apply to scalar and multi-objective optimization problems. To be more precise, we will propose two different methods—an embedding algorithm and a multi-objectivization approach—which are designed to follow an implicitly defined curve where the aim can be to detect the endpoint of the curve (e.g., a root finding problem) or to approximate the entire curve (e.g., the Pareto set of a multi-objective optimization problem). We demonstrate that the novel approaches are very robust in finding the set of interest (point or curve) on several examples.

Categories and Subject Descriptors

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

General Terms

Algorithms, Performance

Keywords

scalar optimization, multi-objective optimization, continuation method, multi-objectivization, evolutionary computation

1. INTRODUCTION

One of the most common problems in optimization is the location of zeros, i.e., to find a point $x \in \mathbb{R}^n$ such that

$$F(x) = 0, \qquad (1)$$

where $F : \mathbb{R}^n \to \mathbb{R}^n$. Such problems arise e.g. when equilibria of systems are sought, in scalar optimization (e.g., $F := \nabla f$, i.e., the gradient of the objective f) or in multi-objective optimization (for the detection of Karush-Kuhn-Tucker (KKT) points). However, problems of the kind (1) may be hard to solve. It is for instance well reported in literature that the (classical) Newton method—certainly the most prominent root finding method—can fail for improper starting points, and that damping strategies can lead to sequences which 'creep' into the root or do even not converge. As a possible remedy, one can try to solve (1) using *homotopy* or *continuation* methods [1]: given a function $G : \mathbb{R}^n \to \mathbb{R}^n$ which is related to F and where a solution x_0 with $G(x_0) = 0$ is known or can be approximated with low effort, one can define the convex homotopy:

$$H(x, \lambda) = \lambda F(x) + (1 - \lambda)G(x). \qquad (2)$$

Since it is

$$H(x, 0) = G(x), \quad H(x, 1) = F(x), \qquad (3)$$

one can try to trace the implicitly defined curve[1]

$$H^{-1}(0) = \{x \in \mathbb{R}^n \mid \exists\lambda \in [0, 1] : \ H(x, \lambda) = 0\} \qquad (4)$$

from $\lambda_0 = 0$ to $\lambda_1 = 1$ starting with x_0 which is the solution of H for λ_0. One famous example for such a curve is the Pareto set of a multi-objective optimization problem (MOP). It is well known that in certain cases (but not in all) the Pareto set of a bi-objective MOP is equal to the set $H_P^{-1}(0)$, where

$$G(x) = \nabla f_1(x), \qquad F(x) = \nabla f_2(x), \qquad (5)$$

f_1 and f_2 the objectives of the MOP, and x_0 the minimizer of G (see Section 4.2 for such an example), which is a direct consequence of the theorem of Kuhn and Tucker [16]. Note that in this case the entire curve is of interest.

Numerical continuation methods have been studied intensively over the last decades and amazing theoretical and practical results have been generated in many fields. However, these methods heavily exploit gradient information which is not always given. In this paper, we propose two related evolutionary path following strategies which are derivative free. The latter of the two methods can be viewed as a certain *multi-objectivization* [15] approach. Both methods will be investigated with respect to their ability to solve scalar and multi-objective optimization problems.

The remainder of this paper is organized as follows: In Section 2, we state the background required for the understand-

GECCO'09, July 8–12, 2009, Montréal Québec, Canada.

[1]The implicit function theorem ensures that the set $H^{-1}(0)$ is under some assumptions on F and G at least locally a curve. Connecting curves do not exist in all cases.

ing of the sequel. In Section 3, we propose the evolutionary continuation strategies: a hill climber for root finding problems which is the core of the curve tracing method we suggest thereafter, and a multi-objectivization approach. In Section 4, we show some numerical results, and finally, some conclusions are drawn in Section 5.

2. BACKGROUND

2.1 Scalar and Multi-objective Optimization

Here we briefly state some notations required for the understanding of the sequel. For a more thorough discussion we refer e.g. to [12, 5].

In the following we consider unconstrained scalar objective optimization problems

$$\min f : \mathbb{R}^n \to \mathbb{R} \tag{6}$$

as well as unconstrained bi-objective optimization problems

$$\min F = (f_1, f_2) : \mathbb{R}^n \to \mathbb{R}^2. \tag{7}$$

While the optimality for one objective is clear, the bi-objective case needs some explanation.

Def 2.1 *(a) Let $v, w \in \mathbb{R}^2$. Then the vector v is* less than *w ($v <_p w$), if $v_i < w_i$ for all $i \in \{1, 2\}$. The relation $\leq_p$ is defined analogously.*

(b) A vector $y \in \mathbb{R}^n$ is dominated *by a vector $x \in \mathbb{R}^n$ (in short: $x \prec y$) with respect to* (7) *if $F(x) \leq_p F(y)$ and $F(x) \neq F(y)$ (i.e. there exists a $j \in \{1, 2\}$ 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 $y \in \mathbb{R}^n$ which dominates x.*

The set of all Pareto optimal solutions is called the *Pareto set*. This set typically—i.e., under mild regularity assumptions—forms a one-dimensional object (i.e., a curve or a set of curves). The image of the Pareto set is called the *Pareto front*.

2.2 Continuation Methods

The first use of continuation methods dates back for more than one century (see, e.g., [1] for an overview), however, numerical continuation has started in the 1950's when computers became available (e.g., [9]) and has been permanently developed further on since then. Common to all algorithms is that the curve tracing problem is transferred to a finite sequence of minimization problems which are executed consecutively. The first techniques directly used λ in (2) to parametrize the curve (so-called *embedding algorithms*, as the one proposed in Section 3.2). Such methods are in general quite effective, but they may lead to problems in certain cases since the curve (4) can not always be parametrized by λ. Alternatively, particular differential equations can be used instead of (2) [4, 14]. This has the advantage that in that case the arclength of the curve, i.e., its 'natural' parameter, can be used leading to a larger numerical stability for the treatment of such problems. On the other hand, such techniques can only be applied if the system is differentiable.

2.3 Evolutionary Methods

Transforming a single objective problem into a suitable multi-objective one leads, in some cases [15, 13, 10], to a reduction over the computational cost of their treatment.

In [15], Knowles et al. introduced the concept of *multi-objectivization* as a procedure in which decomposing a single objective into several ones, or adding new ones, can help to reduce the number of local minimal from the original problem. By doing so, local methods like hill climbers can again be used successfully. In [13], Jensen set the use of these additional objectives (*helper objectives*) changing dynamically in running time.

Successful examples for concrete problems have been presented [15, 13, 10, 11] showing ways to use this approach in other problems. Even though, there does so far not exist a general method to transform the single objective problem into a multi-objective one. There are some details to take care about; for example, the optima of the original problem must remain reachable from the new problem, and it is typically important that the new constructed objectives are in conflict with the original one.

According with some experimental examples [13], using a reduced number of helper objectives is convenient; but, a promising research path is still studying how this affects the diversity of spaces during the search, and also the reduction in terms of solution time.

Finally, a problem which is related to the scope of this paper is dynamic multi-objective optimization where the task is to trace the entire Pareto set with changing external parameters. For this, we refer to [8, 23, 17].

2.4 The Hill Climbers

In the following we shortly present two local search strategies, one for scalar optimization and one for multi-objective optimization, which we will use later on.

Hill Climber with Line Search. Algorithm 1 shows the basic variant of the Hill Climber with Line Search (HCLS) as proposed in [20] for scalar optimization problems. The basic oberservation behind the algorithm is the following one which is well known in mathematical programming: given a point $x_0 \in \mathbb{R}^n$ and a (arbitrary) search direction $\nu \in \mathbb{R}^n \backslash \{0\}$ there is—under mild assumptions on the objective f—a 100 percent chance that either ν or $-\nu$ is a descent direction for f at x_0. The HCLS makes use of this as follows: for x_0 in parameter space a point x_1 from a neigborhood $B(x_0, r)\backslash\{0\}$ is chosen at random, where

$$B(x, r) := \{x \in \mathbb{R}^n : x_i - r_i \leq x_i \leq x_i + r_i \ \forall i = 1, .., n\}, \tag{8}$$

and $r \in \mathbb{R}^n_+$ the 'radius' of the neigborhood. If $f(x_1) < f(x_0)$, then the line search is performed from x_0 in direction $\nu := (x_1 - x_0)$ (lines 4-6 of Algorithm 1, see Section 3.1 for a particular choice of t). In case $f(x_0) < f(x_1)$, an analog line search is performed starting at x_1 but using $-\nu$ as search direction (lines 7-9 of Algorithm 1). The strength of the algorithm has been reported ([20]), however, it has been observed that r is a crucial design parameter for the efficiency of the approach: if the values of r are too large, then the search directions can be misleading (for instance, if x_0 is already near the minimizer, points x_1 which are too far from x_0 yield objective values which are worse, and thus, the search is started from x_1, see line 9 of Alg. 1). If on the other hand r is too small, the 'right' step size t is hard to find as for all line search methods, and thus, the solutions 'creep' toward the minimizer. We will see below that this problem can be solved in the root finding context.

Algorithm 1 Hill Climber with Line Search

Require: starting point $x_0 \in Q$, radius $r \in \mathbb{R}^n_+$
Ensure: sequence $\{x_i\}_{i\in\mathbb{N}}$ of candidate solutions
for $l = 1, 2, \ldots$ **do**
  set $\tilde{x}_1 := x_{l-1}$
  choose $\tilde{x}_2 \in B(\tilde{x}_1, r)\backslash\{\tilde{x}_1\}$ at random
  if $f(\tilde{x}_2) < f(\tilde{x}_1)$ **then**
    $\nu := (\tilde{x}_2 - \tilde{x}_1)/\|\tilde{x}_2 - \tilde{x}_1\|$
    compute $t \in \mathbb{R}_+$ and set $x_l := \tilde{x}_1 + t\nu$.
  else if $f(\tilde{x}_1) < f(\tilde{x}_2)$ **then**
    $\nu := (\tilde{x}_1 - \tilde{x}_2)/\|\tilde{x}_1 - \tilde{x}_2\|$
    compute $t \in \mathbb{R}_+$ and set $x_l := \tilde{x}_2 + t\nu$.
  end if
end for

Hill Climber with Sidestep. The Hill Climber with Sidestep (HCS) has been designed for the numerical treatment of MOPs [19]. It is a local iterative search procedure and intended to be able of both moving toward and along the Pareto set depending on the location of the current iterate. The method is based on the observations on the geometry of multi-objective optimization made in [3] which we present here briefly for the bi-objective case. Whereas one can (roughly) divide the search directions ν in the scalar objective case for a point x_0 into descent direction (i.e., the function value decreases when moving from x_0 in direction ν) or ascent direction, there are (basically) four cases when two objectives are considered. These are the cones

$$\{+,+\}, \quad \{+,-\}, \quad \{-,+\}, \quad \{-,-\}. \tag{9}$$

If, for instance, ν is in $\{-,+\}$ it means that ν is a descent direction for f_1 and an ascent direction for f_2, where f_1 and f_2 are the two objectives of the MOP (analog for the other cones). The observation made in [3] is as follows: if a point x_0 is 'far away' from the (local) Pareto set, then there is a high chance that a randomly chosen search direction is either in $\{-,-\}$ or in $\{+,+\}$ (as in the scalar objective case, if ν is in $\{+,+\}$, then the opposite direction $-\nu$ is in $\{-,-\}$ which points toward the Pareto set. In both cases, the 'hill climbing' can be performed). If on the other hand x_0 is 'near' to the Pareto set, then the chance is high that a randomly chosen search direction is either in $\{-,+\}$ or $\{+,-\}$ (which is pointing, roughly speaking, 'along' the Pareto set– or a 'sidestep' direction in the upward movement. However, it has been observed that for a proper movement along the Pareto set some directions within one cone have to be averaged (design parameter N_{nd} in Algorithm 2).
Algorithm 2 shows a variant of the HCS which utilizes these observations. The algorithm is basically identical to the version which has been proposed in [19] except for one modification (which we will need later on): the sidestep is only performed along directions which are inside $\{+,-\}$, i.e., seeking for improvement of f_2. The algorithm thus generates a sequence of points which are first guided toward the Pareto set, and when the set is reached, the iterates are guided along the Pareto front in direction of the minimizer of f_2.

3. THE ALGORITHMS

Algorithm 2 Hill Climber with Sidestep (HCS)

Require: starting point $x_0 \in Q$, radius $r \in \mathbb{R}^n_+$, number N_{nd} of trials
Ensure: sequence $\{x_l\}_{l\in\mathbb{N}}$ of candidate solutions
$a := (0, \ldots, 0) \in \mathbb{R}^n$
nondom $:= 0$
for $l = 1, 2, \ldots$ **do**
  set $\tilde{x}_1 := x_{l-1}$
  choose $\tilde{x}_2 \in B(\tilde{x}_1, r)\backslash\{\tilde{x}_1\}$ at random
  if $\tilde{x}_1 \prec \tilde{x}_2$ **then**
    $\nu := (\tilde{x}_2 - \tilde{x}_1)/\|\tilde{x}_2 - \tilde{x}_1\|$
    compute $t \in \mathbb{R}_+$ and set $x_l := \tilde{x}_2 + t\nu$.
    nondom $:= 0, \quad a := (0, \ldots, 0)$
  else if $\tilde{x}_2 \prec \tilde{x}_1$ **then**
    $\nu := (\tilde{x}_1 - \tilde{x}_2)/\|\tilde{x}_1 - \tilde{x}_2\|$
    compute $t \in \mathbb{R}_+$ and set $x_l := \tilde{x}_1 + t\nu$.
    nondom $:= 0, \quad a := (0, \ldots, 0)$
  else
    if $f_2(\tilde{x}_2) < f_2(\tilde{x}_1)$ **then**
      $s := 1$
    else
      $s := -1$
    end if
    $a := a + \frac{s}{N_{nd}} \frac{\tilde{x}_2 - \tilde{x}_1}{\|\tilde{x}_2 - \tilde{x}_1\|}$
    nondom $:=$ *nondom* $+ 1$
    if *nondom* $= N_{nd}$ **then**
      compute $\tilde{t} \in \mathbb{R}_+$ and set $x_l := \tilde{x}_l + \tilde{t}a$.
      nondom $:= 0, \quad a := (0, \ldots, 0)$
    end if
  end if
end for

Here we propose two evolutionary strategies for continuation: an embedding algorithm which uses a variant of the HCLS (which is adapted for the root finding context) and a multi-objectivization approach.

3.1 A Hill Climber for Root Finding Problems

Here we adapt the HCS which is presented in Section 2 to the context of root finding. Assume we are given a problem

$$g(x) = 0, \tag{10}$$

where $g : \mathbb{R}^n \to \mathbb{R}^n$ is a nonlinear function. It is well known that problem (10) can be transformed into the minimization problem

$$\min f(x) = \|g(x)\|_2^2, \tag{11}$$

and thus, the original root finding can be attacked by any method for scalar optimization (such as the HCLS) by using the auxiliary problem (11). However, the additional information that the minimum of (11) is given by 0 has two important implications on the hill climber—one obvious implication and one which is less obvious—which make the procedure more effective.
Implication 1: The hill climber can be equipped with a stopping criterion. If the function value of the current iterate x_l is below a given threshold, say $f(x_l) \leq tol$ with $tol \in \mathbb{R}_+$, the iteration can be terminated.
Implication 2: The step size control can be adjusted to the given context. For this, we propose the following strategy: Assume we are given points $x_0, x_1 \in \mathbb{R}^n$ with $\|x_1 - x_0\| \leq r$ (r 'small') and $f(x_1) < f(x_0)$ (as given by the neighborhood

search of the HCLS, see line 3 of Algorithm 1). The question is how to perform the search along $\nu := (x_1 - x_0)/\|x_1 - x_0\|$, i.e., how to find a 'suitable' point $x_2 = x_0 + t\nu$, which reduces to the choice of an appropriate step length $t \in \mathbb{R}_+$. It would be desirable to ask for a point x_2 such that

$$f(x_2) \approx \kappa f(x_0), \tag{12}$$

where $\kappa \in (0, 1)$. Such a procedure would ideally yield linear convergence, which is certainly the most one can expect for an evolutionary strategy ([22]). Assuming that x_2 is not too far from x_0 we can use the Mean Value Theorem to obtain the following estimation

$$f(x_2) - f(x_0) \approx \nabla f(x_0) \cdot (x_2 - x_0), \tag{13}$$

where $a \cdot b$ denotes the inner product of a and b. Using (12) and applying the norm on both sides of (13) leads to

$$(1 - \kappa) f(x_0) \approx t|\nabla f(x_0) \cdot \nu| \tag{14}$$

Finally, since x_1 is near to x_0 the directional derivative $\nabla f(x_0) \cdot \nu$ can be approximated by $(f(x_1) - f(x_0))/\|x_1 - x_0\|$ which leads to the step size

$$t^* := \frac{(1 - \kappa) f(x_0)\|x_1 - x_0\|}{|f(x_1) - f(x_0)|}, \tag{15}$$

which does not use gradient information. The choice of a small value of κ leads to a small value of t^*, and thus, slows down convergence. In turn, large values of κ could lead to a faster convergence, but the probability that (12) is true is lower (note that (13) is a first order approximation).
If t^* is too large, i.e., if $f(x_2) > f(x_0)$ where $x_2 = x_0 + t^*\nu$ (which can certainly be the case in early stages of the iteration process) we propose to use backtracking methods [7]. For the computations done in this paper we have used an Armijo-like quadratic approximation analog to the one presented in [20].
Algorithm 3 shows the principle of the *Hill Climber for Root Finding* (HCR) which is a variant of the HCLS involving the two points discussed above (i.e., the line search in lines 6 and 13 of Algorithm 3 has to be performed as in (15) and the discussion below). Note that x_1 is used to estimate the directional derivative $\nabla f(x_0) \cdot \nu$ which implies that r is not an important design parameter any more: since r determines the maximal distance from x_0 to x_1 it can be chosen 'small', for instance as the square root of the machine precision which is sometimes suggested for finite difference techniques.

3.2 Evolutionary Continuation by Embedding

Assume we are given the problem

$$H_\lambda(x) = 0, \tag{16}$$

where $H : \mathbb{R}^n \to \mathbb{R}^n$ and λ an additional parameter which varies without loss of generality from 0 to 1 (e.g., the convex homotopy (2)). The first evolutionary continuation method we propose is a straightforward application of the embedding algorithm which has been used (successfully) since several decades. The basic idea is to use λ to parametrize the desired curve $H^{-1}(0)$. Starting from a solution x_0 for $\lambda_0 = 0$ the value of λ is increased by a small value, i.e., $\lambda_1 := \lambda_0 + \Delta$, and the problem $H_{\lambda_1}(x) = 0$ is tried to solve. Since continuity of H is assumed and Δ is small, it makes sense to use the former solution x_0 as starting point for this optimization

Algorithm 3 $x_{end} = HCR(f, x_0, tol)$

Require: radius $r \in \mathbb{R}^n_+$, maximal number of iterations $MaxIter$, tolerance value $tol \in \mathbb{R}_+$.
Ensure: best found solution x_{end}, stopping criterion: $f(x_l) \le tol$.
1: **for** $l = 1, 2, \ldots, MaxIter$ **do**
2: set $\tilde{x}_1 := x_{l-1}$
3: choose $\tilde{x}_2 \in B(\tilde{x}_1, r)\backslash\{\tilde{x}_1\}$ at random
4: **if** $f(\tilde{x}_2) < f(\tilde{x}_1)$ **then**
5: $\nu := (\tilde{x}_2 - \tilde{x}_1)/\|\tilde{x}_2 - \tilde{x}_1\|$
6: compute $t \in \mathbb{R}_+$ and set $x_l := \tilde{x}_1 + t\nu$.
7: **if** $f(x_l) \le tol$ **then**
8: $x_{end} := x_l$
9: **return**
10: **end if**
11: **else if** $f(\tilde{x}_1) < f(\tilde{x}_2)$ **then**
12: $\nu := (\tilde{x}_1 - \tilde{x}_2)/\|\tilde{x}_1 - \tilde{x}_2\|$
13: compute $t \in \mathbb{R}_+$ and set $x_l := \tilde{x}_2 + t\nu$.
14: **if** $f(x_l) \le tol$ **then**
15: $x_{end} := x_l$
16: **return**
17: **end if**
18: **end if**
19: **end for**
20: $x_{end} := x_l$

problem since the minimizer of H_{λ_1} is assumed to be 'close' to x_0. This procedure is repeated until $\lambda_{final} = 1$.
Algorithm 4 shows the basic version of this approach. The root finding is done by the HCR. For simplicity, we have chosen the increment Δ as constant. For complicated problems, however, this incremental value can easily be chosen adaptively: the number of function calls of HCR required to compute the 'update' x_{i+1} from a given solution x_i of H_{λ_i} is a measure for the choice of $\Delta_i = \lambda_{i+1} - \lambda_i$ (if too much function calls have been spent, the next increment has to be chosen smaller, and vice versa).

Algorithm 4 Embedding Algorithm using HCR

Require: starting point $x_0 \in \mathbb{R}^n$, incremental step $\Delta \in \mathbb{R}_+$, tolerance value $tol \in \mathbb{R}_+$.
Ensure: finite sequence of approximate solutions x_i
set $\lambda := 0$
set $i := 1$
while $\lambda < 1$ **do**
$x_i := HCR(H_\lambda, x_{i-1}, tol)$
$i := i + 1$
$\lambda := \lambda + \Delta$
end while

Though this algorithm is already very efficient in general (see next section for examples), it has two potential drawbacks: (i) The parameter λ is used to trace the curve $H^{-1}(0)$, which is not always the best choice as mentioned above, and (ii) if the underlying problem is an optimization problem (and not a 'classical' root finding problem) the gradients of the objectives are required. The next algorithm is designed to overcome these two problems.

3.3 A Multi-objectivization Approach

Assume we are given either the problem

$$f(x) = 0, \qquad g(x) = 0, \tag{17}$$

where $f, g : \mathbb{R}^n \to \mathbb{R}^n$ (these functions could be given by F and G as in (2) or by H_0 and H_1 as in (16)), or the problem

$$\min f(x), \qquad \min g(x), \tag{18}$$

where $f, g : \mathbb{R}^n \to \mathbb{R}$. Both problems, the root finding problem and the minimization problem, can be multi-objectivized into the following bi-objective MOP:

$$\min F := (\tilde{f}, \tilde{g}) : \mathbb{R}^n \to \mathbb{R}^2, \tag{19}$$

where $\tilde{f} = \|f\|$ and $\tilde{g} = \|g\|$ in the root finding context and $\tilde{f} = f$ and $\tilde{g} = g$ in case of (18).
Problem (19) can in principle tried to be solved by any multi-objective search strategy, however, since the main interest in this case is either the extreme point of the curve (in case a root or a minimizer of f is sought) or the curve (e.g., when the KKT points of a MOP have to be traced), a global approach is not needed in most cases since a local approach can accomplish this task faster. Here we propose to use the HCS since that algorithm is specialized on the movement along the curve. Experiences in numerical continuation have shown that a movement along a curve which leads to the desired point are more stable and successful than 'just' using hill climbing methods applied on f [1]. So far, however, we do not have an example which confirms this observation with the HCS except for problems related to multi-objective optimization. This will be one topic for future research.
Note that in this curve tracing problem the effect of the 'hill climber' and the 'sidestep' procedures of the HCS (see Algorithm 2) are interchanged: the 'sidestep' (line 23 of Algorithm 2) is in fact a movement along the curve, and thus, a movement toward the minimizer of f, whereas the 'hill climbing step' (lines 8 and 12 of Algorithm 2) can be viewed as a correction step toward the Pareto set (i.e., the solution path).
In the following we describe the modifications we have done to the HCS as described in [19] in order to adapt it to the path tracing context:
Adaptation to root finding problem (17): (i) the search direction in the sidestep has to be changed (as already done in Algorithm 2). In the current context we are particularly interested in directions ν in $\{+, -\}$. (ii) The step size control for the sidestep has been changed to the one in (15) since the root of f is of interest. We have observed that the performance of the strategy was positively affected by this change. (iii) as for the HCR the knowledge of the minimum of f allows for a stopping criterion (i.e., to stop the process if $\|f(x_l)\| \leq tol_1$). Further, we have used another threshold $tol_2 > tol_1$ such that the path following process was switched to the HCR when $\|f(x_l)\| \leq tol_2$ since correction steps are not required any more when the current iterate is near to a solution.
Adaptation to minimization problem (18): in that case the HCS can be applied as suggested in [19]. The only modification which is required is the change of the sidestep search direction as described above.
Finally, a note on the choice of g: if there is no 'suitable' choice of g (e.g., given by the application or coming from engineering judgement) one can e.g. define for a given starting point x_0 the function

$$g(x) = x - x_0, \tag{20}$$

which we have done in Section 4.1.

Note that this strategy overcomes the two potential drawbacks discussed in the previous subsection: in the context of optimization, no gradient information is required, and no parameter is used to trace the curve such as it is λ in the embedding algorithm. However, one expects by construction of the two strategies that if the latter algorithm succeeds it will outperform the present multi-objectivization approach (which is e.g. the case in Section 4.2).

4. NUMERICAL RESULTS

Here we present some numerical results for 'classical' root finding problems as well as for some path approximation problems.

4.1 Root Finding

To test our algorithms we consider the following two well known test functions which we treat as root finding problems:

$$g_1, g_2 : \mathbb{R}^n \to \mathbb{R}$$

$$\begin{aligned} g_1(x) &:= \sum_{i=1}^{n-1} 100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2 \quad \text{(Rosenbrock [18])} \\ g_2(x) &:= \sum_{i=1}^{n} |x_i| + \prod_{i=1}^{n} |x_i| \qquad \text{(Schwefel [21])} \end{aligned} \tag{21}$$

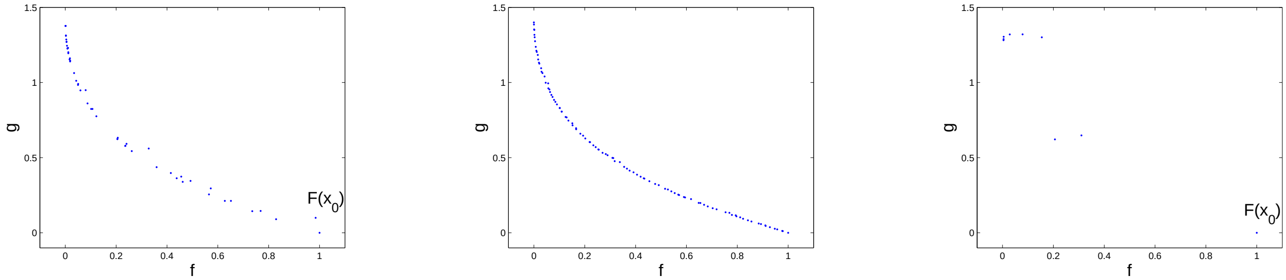
For comparison we use the hill climbers $(1+1)$ES [21], the HCLS (i.e., the version which is described in [20]), and the HCR. Further, we have applied NSGA-II [6] and the HCS on the multi-objectivized problems

$$\min(f, \|x - x_0\|) : \mathbb{R}^n \to \mathbb{R}^2, \quad i = 1, 2, \tag{22}$$

taking the best value of g at the end of the run. Tables 1-3 show the results on g_1 and for dimensions $n = 2, 5$ and 10, Tables 4-6 show the results on g_2.
Several observations can be made: The 'best' strategy (according to the two test problems) seems to be the HCR. The direct comparison to the HCLS indicates that the reason has to be the newly developed step size control (15). Also the HCS whose main focus is the approximation of the entire curve (see Figure 1) is highly competitive. The comparison to the other multi-objectivization approach (using NSGA-II on (22)) is of course a bit unfair since the approach of NSGA-II is global. Thus, if a larger budget of function calls would have been available, this approach would have won (see also Figure 1). However, the scope was to find the roots quickly, and in this case local methods are advantageous (if they converge). The situation changes completely for f_2 and $n = 10$ (Table 6). In that case, all local methods fail and only NSGA-II delivers useful results.

4.2 Curve Approximation

Now we present two examples of continuation problems where the entire curve is of interest.



(a) HCS, $N_{nd} = 6$ (b) Pareto front (c) HCS, $N_{nd} = 1$

Figure 1: Numerical result of MOP (22) for g_1 of (21) and using the starting point $x_0 = (0, 0)$. The Pareto front has been computed using NSGA-II. Subfigures (a) and (c) show solution paths of the HCS with different values of N_{nd}. Note that the higher the value of N_{nd}, the better the approximation of the Pareto curve gets.

Table 1: Result of 100 runs on Rosenbrock's example (21) and for dimension $n = 2$. The table shows the percentage of runs that reached a specified tolerance for error, using each algorithm over the same number of function evaluations. $g(x_0)$ is the function value for the initial point. x_f is the final point (root reached).

n=2, Nmax=1000, $g(x_0) \approx 8.6\,\text{x}10^7$ $err = \|\|g(x_f)\|\|^2$			
Method	err< 1	err< 0.1	err < 0.01
$(1+1)$ES	91%	82%	70%
HCLS	81%	81%	37%
HCR	82%	81%	80%
HCS	100%	100%	98%
NSGA-II	49%	19%	4%

Table 2: Result of 100 runs on Rosenbrock's example and for $n = 5$ (see Table 1 for details).

n=5, Nmax=10,000, $g(x_0) \approx 3.4\,\text{x}10^8$ $err = \|\|g(x_f)\|\|^2$			
Method	err< 10	err< 1	err < 0.1
(1+1) ES	100%	83%	82%
HCLS	99%	49%	1%
HCR	100%	84%	84%
HCS	99%	88%	78%
NSGA-II	36%	1%	1%

Table 3: Result of 100 runs on Rosenbrock's example and for $n = 10$ (see Table 1 for details).

n=10, Nmax=100,000, $g(x_0) \approx 7.7\,\text{x}10^8$ $err = \|\|g(x_f)\|\|^2$			
Method	err< 10	err< 1	err < 0.1
(1+1) ES	100%	84%	84%
HCLS	98%	0%	0%
HCR	100%	80%	80%
HCS	99%	91%	91%
NSGA-II	23%	0%	0%

Table 4: Result of 100 runs on Schwefel's example and for $n = 2$ (see Table 1 for details).

n=2, Nmax=1000, $g(x_0) \approx 440$ $err = \|\|g(x_f)\|\|^2$			
Method	err< 1	err< 0.1	err < 0.01
(1+1) ES	53%	53%	52%
HCLS	86%	86%	76%
HCR	97%	96%	96%
HCS	79%	78%	77%
NSGA-II	53%	46%	9%

Table 5: Result of 100 runs on Schwefel's example and for $n = 5$ (see Table 1 for details).

n=5, Nmax=30000, $g(x_0) \approx 3.2\,\text{x}10^6$ $err = \|\|g(x_f)\|\|^2$			
Method	err< 10	err< 1	err < 0.1
(1+1) ES	9%	8%	8%
HCLS	3 %	3%	0%
HCR	48%	48%	48%
HCS	20%	19%	19%
NSGA-II	53%	36%	1%

Table 6: Result of 100 runs on Schwefel's example and for $n = 10$ (see Table 1 for details).

n=10, Nmax=100,000, $g(x_0) \approx 1.0\,\text{x}10^{13}$ $err = \|\|g(x_f)\|\|^2$			
Method	err< 10	err< 1	err < 0.1
(1+1) ES	1%	1%	1%
HCLS	0%	0%	0%
HCR	0%	0%	0%
HCS	0%	0%	0%
NSGA-II	23%	15%	1%

4.2.1 *Computation of a Pareto Front*

The first example comes from multi-objective optimization. Consider the bi-objective MOP

$$
\begin{aligned}
F &= (f_1, f_2) : \mathbb{R}^{50} \to \mathbb{R}^2 \\
f_1(x) &= (x_1 - 1)^4 + \sum_{i=2}^{50} (x_i - 1)^2 \\
f_2(x) &= \sum_{\substack{i=1 \\ i \neq 2}}^{50} (x_i + 1)^2 + (x_2 + 1)^2
\end{aligned}
\tag{23}
$$

Since the two objectives are convex, the Pareto set is given due to the theorem of Kuhn and Tucker [16] by $H^{-1}(0)$, where

$$H(x, \lambda) = \lambda \nabla f_1(x) + (1 - \lambda) \nabla f_2(x) \tag{24}$$

and $\lambda \in [0, 1]$. Figure 4.2.1 shows numerical results coming from the embedding method applied on homotopy (24) and from the HCS applied on MOP (23). For the embedding method we have chosen $x_0 = (-1, \ldots, -1) \in \mathbb{R}^{50}$, i.e., the minimizer of f_2, $tol = 0.01$, and $\Delta = 0.01$. The HCS was started with x_0.

The embedding algorithm spent 12,646 function calls (H) for the 100 root finding problems (i.e., one for each value of λ), and the HCS spent 21,706 function calls (F). That is, in this case, the embedding algorithm was faster (roughly by a factor of two), however, used in turn gradient information. Since both algorithms came up with a good representation of the Pareto set/front with a reasonable amount of function calls, it can be said that both algorithms accomplished the task sufficiently.

4.2.2 *A Nondifferentiable Continuation Problem*

Finally, we consider the problem of approximating the non-differentiable curve $H_\lambda^{-1}(0)$, where

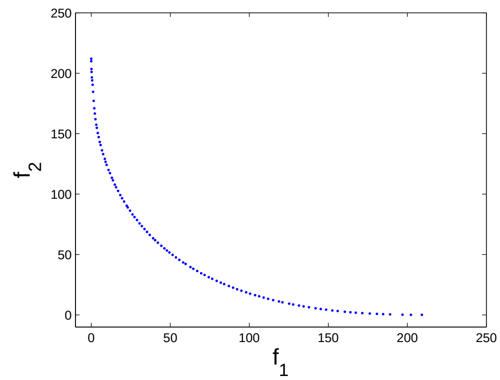
$$
\begin{aligned}
&H_\lambda : \mathbb{R}^3 \to \mathbb{R}^3 \\
&H_\lambda(x) = \begin{pmatrix} \|x\|_\infty - 1 \\ x_3 - 0.5 \max(\cos(2\pi \min_{i=1,2} |x_i|), \sin(2\pi \min_{i=1,2} |x_i|)) \\ x_1 - 2\lambda + 1 \end{pmatrix},
\end{aligned}
\tag{25}
$$

and $\lambda \in [0, 1]$. This problem is at first sight not closely related to optimization, however, problems of that kind (but harder ones) can occur in optimal control theory when solving Hamilton Jaboci Bellman (HJB) equations [2]. A HJB equation is a particular partial differential equation which has to be solved backward in time. Since such solutions are typically not smooth, problem (25) can be viewed as a possible starting point of the new methods to that challenging field.

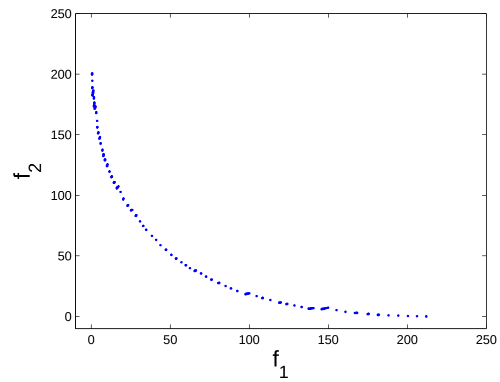
Figure 3 shows a numerical result of the embedding algorithm, where we have used $x_0 = [-1, -1, 0]$, $tol = 0.001$, and $\Delta = 0.001$. That is, to trace the curve 1000 runs of HCR have been performed successfully. The total number of function calls was 4758, i.e., in average less than five function calls had to be spent for each optimization problem.

5. CONCLUSIONS AND FUTURE WORK

In this paper we have proposed two different evolutionary strategies for numerical continuation and have investigated



(a) Result Embedding



(b) Result HCS

Figure 2: Numerical result for homotopy (24) using the embedding algorithm (above) respectively for MOP (23) using the HCS (below).

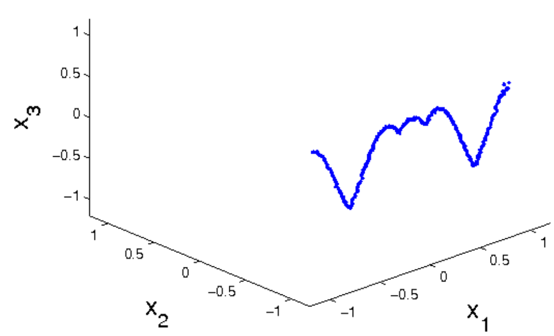


Figure 3: Numerical result of the embedding algorithm on problem (25).

their ability to solve both scalar and multi-objective optimization problems. The first method is an embedding method which utilizes a hill climber which was adapted in this paper for the root finding context. The resulting method is capable of tracing the curve efficiently, but requires an external parameter to trace the curve. Further, in the context of optimization, the first gradients of the objectives—e.g., to detect the KKT points of a given multi-objective optimization problem—are required. The second method is a multi-objectivization approach which overcomes both potential drawbacks and is capable of approximating the efficient set of the auxiliary multi-objectivized problem without using gradient information and without the use of an external parameter. It remains, however, to show the particular benefit of this approach beyond its use for multi-objective optimization problems.

Next to this question there are some interesting issues which we would like to address in the future. There is for instance the general improvement of the methods with respect to their performance which involves the adaptive choice of the design parameters. Further, we will investigate the potential of the new methods with respect to hybridizations with other methods in order to obtain fast and efficient optimization strategies. Finally, we seek to apply our new methods. One interesting application could be to solve numerically Hamilton Jacobi Bellman equations which arise in optimal control and in stochastic programming.

6. REFERENCES

[1] E. L. Allgower and K. Georg. *Numerical Continuation Methods*. Springer, 1990.

[2] D. P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 2001.

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

[4] D. F. Davidenko. On a new method of numerical solution of systems of nonlinear equations. *Doklady Akad. Nauk SSSR (N.S.)*, 88:601–602, 1953.

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

[6] K. Deb, S. Agrawal, A. Pratab, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In *PPSN IV: Proceedings of the 6th International Conference on Parallel Problem Solving from Nature*, pages 849–858, Paris, France, 2000.

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

[8] M. Farina, K. Deb, and P. Amato. Dynamic Multi-objective Optimization Problems: Test Cases, Approximation, and Applications. *Lecture Notes in Computer Science*, 8(5):425–442, 2004.

[9] F. Ficken. The continuation method for functional equations. *Communications on Pure and Applied Mathematics*, 4:435–456, 1951.

[10] D. Greiner, J. M. Emperador, G. Winter, and B. Galván. Improving Computational Mechanics Optimum Design Using Helper Objectives: An Application in Frame Bar Structures. In *Proceedings of the Fourth International Conference on Evolutionary Multi-Criterion Optimization (EMO 2007)*, volume 4403, pages 575–589. Springer Berlin / Heidelberg, 2007.

[11] J. Handl, S. C. Lovell, and J. Knowles. Investigations into the effect of multiobjectivization in protein structure prediction. In *Proceedings of the 10th international conference on Parallel Problem Solving from Nature*, pages 702–711. Springer-Verlag, 2008.

[12] D. M. Himmelblau. *Applied Nonlinear Programming*. McGraw Hill Book Company, 1972.

[13] M. T. Jensen. Helper-objectives: Using Mlti-objective Evolutionary Algorithms for Single-objective Optimisation. *Journal of Mathematical Modelling and Algorithms*, (4):323–347, Dec. 2004.

[14] H. B. Keller. Global homotopies and newton methods. In *Recent Advances in Numerical Analysis*, pages 73–94. Academic Press, New York, 1978.

[15] J. D. Knowles, R. A. Watson, and D. W. Corne. Reducing local optima in single-objective problems by multi-objectivization. In *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO 2001)*, pages 269–283, London, UK, 2001. Springer-Verlag.

[16] 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.

[17] X. Li, J. Branke, and M. Kirley. On Performance Metrics and Particle Swarm Methods for Dynamic Multiobjective Optimization Problems. In *2007 IEEE Congress on Evolutionary Computation (CEC'2007)*, pages 576–583, 2007.

[18] H. H. Rosenbrock. An automatic method for finding the greatest of least value of a function. *Computer Journal*, 3(3):175–184, 1960.

[19] O. Schütze, G. Sanchez, and C. A. Coello Coello. A new memetic strategy for the numerical treatment of multi-objective optimization problems. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2008)*, pages 705–712. ACM Press, Atlanta, USA, 2008.

[20] O. Schütze, E.-G. Talbi, C. A. Coello Coello, L. V. Santana-Quintero, and G. Toscano Pulido. A Memetic PSO Algorithm for Scalar Optimization Problems. In *Proceedings of the 2007 IEEE Swarm Intelligence Symposium (SIS 2007)*, pages 128–134, Honolulu, Hawaii, USA, April 2007. IEEE Press.

[21] H.-P. Schwefel. *Evolution and Optimum Seeking*. Wiley-Interscience, 1995.

[22] H.-M. Voight, H. Mühlenbein, and D. Cvetkovic. Fuzzy recombination for the continuous breeder genetic algorithms. In L. J. Eshelman, editor, *Sixth International Conference on Genetic Algorithms*, pages 104–112, 1995.

[23] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, and E. Tsang. Prediction-Based Population Re-initialization for Evolutionary Dynamic Multi-objective Optimization. In *Proceedings of the Fourth International Conference on Evolutionary Multi-Criterion Optimization (EMO 2007)*, pages 832–846. Springer Berlin / Heidelberg, 2007.