A Memetic Variant of R-NSGA-II for Reference Point Problems

Jesús Alejandro Hernández Mejía[1], Oliver Schütze[1], and Kalyanmoy Deb[2]

[1] Computer Science Department, CINVESTAV-IPN, Av. IPN 2508, C. P. 07360,
Col. San Pedro Zacatenco, Mexico City, México
jhernandez@computacion.cs.cinvestav.mx, schuetze@cs.cinvestav.mx
[2] College of Engineering
Michigan State University
428 S. Shaw Lane, East Lansing, MI 48824, USA
kdeb@msu.edu

Abstract. The task in multi-objective optimization is to optimize several objectives concurrently. Since the solution P of such problems is typically given by an entire set, the entire approximation of P is not always possible or even desired. Instead, it makes sense to concentrate on particular points or regions of the solution set. In case the decision maker has a certain idea about the performance of his/her product, reference point methods can be used to find the solutions that are closest to the given reference point. Evolutionary algorithms are advantageous for the treatment of such problems in particular if there are multiple reference points and/or the objectives are highly multi-modal, however, they suffer the general drawback of a slow convergence rate.

In this paper, we argue that the recently proposed Directed Search Method is suitable for an integration into evolutionary algorithms for reference point problems. We investigate the Directed Search Method for the current context and discuss its integration into the state-of-the-art algorithm R-NSGA-II. Numerical results on several benchmark problems indicate that the novel memetic strategy significantly increases the performance of its base algorithm.

Keywords: multi-objective optimization, reference point problem, memetic strategy.

1 Introduction

In many applications one is faced with the problem that several objectives have to be optimized concurrently. One important characteristic of such *multi-objective optimization problems* (MOPs) is that their solutions are typically not given by a singleton as for 'classical' scalar optimization problems (SOPs) but instead form $(k-1)$-dimensional objects, where k is the number of objectives involved in the problem.

In many cases the decision maker is interested in obtaining an approximation of the entire solution set, the so-called Pareto set, since it gives him/her the overview of all possible realizations of the project. For this, many different algorithms have been proposed so far. Among them, set based algorithms such as multi-objective evolutionary algorithms (MOEAs, see [1–3]), subdivision ([4–6]) or cell mapping techniques ([7,8])

A.-A. Tantar et al. (eds.), *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V*, Advances in Intelligent Systems and Computing 288,
DOI: 10.1007/978-3-319-07494-8_17,

have caught the interest of many researchers. Reasons for this include that these set based methods allow for the approximation of the entire set of interest in one run of the algorithm, and that they are further characterized by a great robustness and minimal requirement on the model.

In certain cases, however, it is not wanted or desired to obtain the entire solution set. Instead, selected Pareto optimal solutions are of interest to the decision maker. To obtain a single solution, e.g. scalarization methods can be used that transform the MOP into a SOP (e.g., [9–12]). If the decision maker already has a rough idea about the properties of his/her product, he/she might use *reference point methods* ([10, 13, 14]). Methods of that kind try to compute solutions that are as close as possible to a given reference point (or to a set of reference points). As for the computation of the entire Pareto set, evolutionary algorithms may be beneficial for the numerical treatment of reference point problems. A state-of-the-art algorithm that addresses this problem is the R-NSGA-II (the Reference point Non-dominated Sorting Genetic Algorithm, see [15]). R-NSGA-II can be applied successfully even if more than one reference point is given and/or if the optimization model is highly multi-modal.

In this paper, we argue that the recently proposed *Directed Search Method* (DS, see [16]) is well-suited for a hybridization with an evolutionary algorithm such as R-NSGA-II. The DS is a point-wise iterative search procedure that allows to steer the search into any direction d given in objective space. It is important to note that DS can be used with and without gradient information. The latter can be done by exploiting neighborhood information which is typically given for set based optimization strategies. Thus, the potential of DS for its hybridization with specialized evolutionary algorithms. In this study, however, we make a first attempt for the efficient integration of DS and will restrict ourselves to its gradient based version. Nevertheless, we can already observe significant improvements in the performance of the memetic strategy compared to its base MOEA on several examples.

The remainder of this paper is organized as follows: In Section 2, we briefly state the background required for the understanding of this paper. In Section 3, we present the DS for the treatment of reference point methods. In Section 4, we discuss how to integrate DS into R-NSGA-II leading to a new memetic strategy. In Section 5, we present some numerical results on some widely used benchmark models. Finally, we conclude in Section 6 and give possible paths for future work.

2 Background

A MOP can be mathematically expressed as

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

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

$$F : \mathbb{R}^n \to \mathbb{R}^k, \quad F(x) = (f_1(x), \dots, f_k(x)), \tag{1}$$

and where each objective function is given by $f_i : \mathbb{R}^n \to \mathbb{R}$. Here we will handle mainly unconstrained problems (i.e., $Q = \mathbb{R}^n$), however, most of the problems considered in Section 4 are box constrained. That is, the domain is given by

$$Q = \{x \in \mathbb{R}^n \ : \ l_i \le x_i \le u_i, \quad i = 1, \dots, n\}, \tag{2}$$

where $l \in \mathbb{R}^n$ and $u \in \mathbb{R}^n$ are the lower and upper bounds, respectively.

The optimality of a MOP is defined by the concept of *dominance* ([17]): 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 $f_i(x) \le f_i(y)$, $i = 1, \dots, k$, and there exists an index j such that $f_j(x) < f_j(y)$, else y is non-dominated by x. A point $x \in \mathbb{R}^n$ 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*, and is denoted by $\mathcal{P}$. The image $F(\mathcal{P})$ of the Pareto set is called the *Pareto front*. Both sets typically form a $(k-1)$-dimensional object.

The classical reference point problem for point-wise iterative methods can be written as follows:

$$\min_{x \in Q} D(Z, F(x)), \tag{3}$$

where $Z \in \mathbb{R}^k$ is the given reference point and D is a chosen metric (we will use the Euclidean norm in this study). Since we are dealing here with entire sets of candidate solutions (populations or archives), it is advantageous to re-state the problem as:

$$\min_{\substack{A \subset Q \\ |A| = N}} \ dist(Z, F(A)), \tag{4}$$

where A is an archive of magnitude N, and $dist$ measures the distance of two sets as:

1. $\text{dist}(u, A) := \inf_{v \in A} D(u, v)$

2. $\text{dist}(B, A) := \sup_{u \in B} \ \text{dist}(u, A)$

Hereby, u and v denote points and A and B sets. The advantages of (4) over (3) in our context are that (i) archives can be considered to be ‘good‘ even if they contain elements $a_i \in A$ that are far away from Z (those elements will not be selected by the decision maker anyway), and (ii) this concept can be extended to the case $Z = Z_1 \cup \ldots \cup Z_l$ contains multiple reference points Z_i, $i = 1, \dots, l$.

3 Directed Search for Reference Point Problems

The DS ([16]) allows to steer the search into any direction $d \in \mathbb{R}^k$ in objective space: Given d, we are interested in a direction $\nu \in \mathbb{R}^n$ in parameter space such that

$$\lim_{t \searrow 0} \frac{f_i(x_0 + t\nu) - f_i(x_0)}{t} = d_i, \quad i = 1, \dots, k, \tag{5}$$

where $x_0 \in \mathbb{R}^n$ is the given starting point. Denote by $J(x_0)$ the Jacobian of F at x_0,

$$J(x_0) = \left(\nabla f_1(x_0)^T, \dots, \nabla f_k(x_0)^T\right)^T \in \mathbb{R}^{k \times n}, \tag{6}$$

then Equation (5) can in matrix vector notation be written as $J(x_0)\nu = d$. Since typically $k \ll n$, the above system can considered to be (highly) underdetermined. One suggesting choice is to use the greedy solution:

$$v_+ := J(x_0)^+ d \tag{7}$$

It is known that ν_+ solves $\min\{\|v\| \; : \; J(x_0)v = d\}$. Hence, for small values of t in the line search $x_1 := x_0 + t\nu$ the largest movement in $d-$direction is expected when choosing $v = v_+$. We note that the DS can be made derivative free via exploiting neighborhood information ([18]), however, we will not utilize this feature in this study.

In the following we discuss DS for reference point problems (3) for unconstrained MOPs. Given x_0 and Z the greedy direction d_Z in objective space is certainly given by:

$$d_Z = Z - F(x_0) \tag{8}$$

and can thus be used by the DS approach. That is, an application of DS is hence equivalent to the numerical realization of the following initial value problem (IVP):

$$\begin{aligned} x(0) &= x_0 \in \mathbb{R}^n \\ \dot{x}(t) &= J(x(t))^+(Z - F(x(t))), \quad t > 0 \end{aligned} \tag{9}$$

One appealing reason for choosing DS for the reference point problem is that we can expect (local) quadratic convergence in the special case that Z is feasible, that is, if there exists a $x^* \in Q$ such that $F(x^*) = Z$. To see this, consider the root finding problem:

$$\begin{aligned} g :&\mathbb{R}^n \to \mathbb{R}^k \\ g(x) &= F(x) - Z \end{aligned} \tag{10}$$

An iteration of the Gauss-Newton method (which converges locally quadratically, see [19]) is given by

$$\begin{aligned} x_{i+1} &= x_i - J_g(x_i)^+ g(x_i) = x_i - J(x_i)^+(F(x_i) - Z) \\ &= x_i + J(x_i)^+(Z - F(x_i)), \end{aligned} \tag{11}$$

where $J_g(x)$ denotes the Jacobian of g at x. On the other hand, an iteration performed via DS yields

$$x_{i+1} = x_i + t_i J(x_i)^+(Z - F(x_i)), \tag{12}$$

where $t_i \in \mathbb{R}_+$ is the chosen step size. Comparing (11) and (12) we see that both iterations coincide for $t_i = 1$.

It can on the other hand certainly not be expected that Z is always feasible. In order to see how to proceed in the unfeasible case, we have to understand the geometry of the solution curves of IVP (9) (see Figure 1 for the image of such a solution curve). We can divide these solution curves into two parts: In part I, a movement in d-direction is performed. Once a boundary point[1] x of (MOP) is reached, the movement is steered

[1] We can characterize a boundary point x as follows: There exists a direction d such that no movement in that direction can be performed.

along the linearized Pareto front at $F(x)$. This movement along the Pareto front constitutes part II of the solution curve. Unfortunately, the ordinary differential equation in (9) is stiff in part II which means that its numerical treatment gets complicated. To see the stiffness, let x be a boundary point. Then there exists a direction d such that the equation $J(x)\nu = d$ has no solution. This is equivalent to $rank(J(x)) < k$ which means that the condition number of $J(x)$ is infinite.

In order to overcome this stiffness, we can perform a linearization to steer the search directly as follows: Let x be a boundary point with weight $\alpha \in \mathbb{R}^k$ such that

$$\sum_{i=1}^{k} \alpha_i \nabla f_i(x) = 0. \tag{13}$$

Further, let a QR factorization of α be given,

$$\alpha = QR = (q_1, \dots, q_k)R. \tag{14}$$

Then the column vectors of $Q_2 = (q_2, \dots, q_k) \in \mathbb{R}^{k\times(k-1)}$ form an orthonormal basis of the linearized Pareto front at $F(x)$. Given the direction d_Z, the projection onto the tangent space is thus given by

$$d_{new} = Q_2(\underbrace{Q_2^T Q_2}_{=I})^{-1} Q_2^T d_Z = Q_2 Q_2^T d_Z. \tag{15}$$

Denote by $P(x) := Q_2(x)Q_2(x)^T$ the projection described above. Then we suggest to solve the following IVP for points x_0 where $F(x_0)$ is on the boundary of the image

$$\begin{aligned} x(0) &= x_0 \in \mathbb{R}^n \\ \dot{x}(t) &= J(x(t))^+ P(x(t))(Z - F(x(t))), \quad t > 0 \end{aligned} \tag{16}$$

The switch between (9) and (16) can be handled via monitoring the condition number of $J(x)$: If $J(x) < tol$ for a certain tolerance value we suggest to choose to follow the flow in (9), else (16) should be taken.

It remains to select the step sizes in order to obtain efficient numerical realizations of (9) and (16) which is subject of ongoing research. For the implementations used for this study, we have followed the step size control of the DS Descent Method described in [20] together with the initial step size $t_0 = 1$ motivated by the above discussion. For the numerical realization of (16) we have adapted the step size control of the DS Continuation Method which is also described in [16].

4 Integrating DS into R-NSGA-II

In the following we describe a possible integration of DS into R-NSGA-II, however, we would like to stress that a similar integration can in principle be done into any other set based reference point method. We will briefly describe R-NSGA-II as well as where we decided to apply local search (LS) and the reason for it.

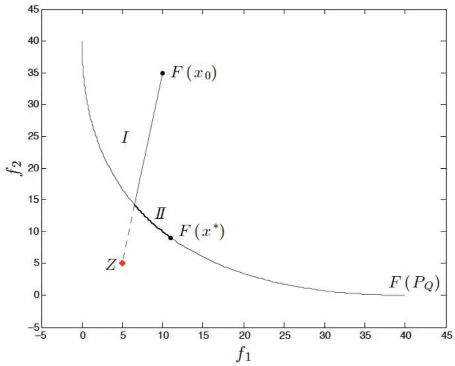


Fig. 1. Typical image of a solution curve of (9)

In one iteration of R-NSGA-II parent and offspring population are combined and non dominated sorting is performed to obtain different levels of non dominated fronts. Analogously to NSGA-II, the first non dominated fronts will be chosen to be included into the next parent population. However, since the last front could not be included but partially, a modified crowding distance is associated to each solution in order to decide which individuals to take. The new crowding distance is the lowest hierarchy the solution has in the sorted lists of distances to each reference point, being hiearchy one the closest solution, two the second closest and so on. A parameter ϵ is introduced to control the spread of solutions in the final archive. We decided to apply LS on the parent population before an iteration starts, hence, improved solutions are emphasized and are expected to create new solutions similar to them in the next generations.

As for the LS algorithm integration within a memetic strategy, many aspects have to be considered. Among the main (generic) parameters are: The number of individuals to which LS will be applied, the maximal iteration number (depth) of the LS applied to an individual, and frequency of the same.

Next to these generic parameters—see Table 1 for the choices we made for our implementations—there are some more specific ones. Among them, the decision of which solutions will be chosen for improvement is an important aspect. Considering R-NSGA-II, one could try to improve individuals on the first non dominated front and backwards from then on. This approach, however, is not useful in particular in higher dimensional spaces since most of the individuals will belong to this front. A second (and more promising) alternative is to improve solutions which are already close to reference points whether they are dominated or not. This arises the new problem of which reference points to be taken for improvement: Considering two or more reference points, it is likely that the closest solutions within the population are not the same (and even if they were), hence, some reference points would gain more attention than others. We propose a possible solution as follows: Assign an integer identifier to each reference point and store the index of the last reference point improved. Starting from improving the first reference point, switch to improve the next one (by modular addition) until the number of individuals designated for local search is reached.

Another point is that DS—as well as all other local search engines—can get stuck on regions which are not globally optimal. In order to prevent wasting function evaluations on currently non-improvable reference points, one last parameter will be used: The maximum number of improvement trails toward a given reference point.

As local search we use the numerical realization of DS explained in the previous section with one exception: Following the suggestion made in [21] we will perform a greedy search toward the Pareto set in case the individual x_0 is 'far away' from the solution set which can be monitored by the angle between the objectives gradients.

Finally, we note that the feasibility of reference points represent a potential problem for the DS approach. Though it makes sense to present to the decision maker better (i.e., dominating) solutions than for a given feasible reference point—as e.g. done in R-NSGA-II—it is ad hoc unclear in which direction to perform the movement. Further, it does not comply with the problem statement in (4) since optimal solutions for (4) are ones those image coincides with the reference point. Here, we will simply stop the movement toward a reference point once feasibility has been detected and current solutions are near enough.

Having stated this we are in the position to explain the hybrid RDS-R-NSGA-II (compare to Algorithm 1):

1. Compute the distance of each member of the population to each reference point in ascending order. (Line 1)
2. Mark the feasible reference points found so far as non improvable. Set a counter of solutions improved so far to zero. (Lines 2-3)
3. Cycle through the set of reference points (starting from the last one used, or one if this is the first call) until an improvable reference point is found. Set a test counter variable to zero and a direction vector as the empty set. (Lines 4-8)
4. If the test counter reaches the maximum tests, mark the current reference point as non improvable and continue to Step 3, else increment the test counter. (Line 10)
5. Obtain the next closest solution for the current reference point and set it as the working iterate, compute the Jacobian and the best descent direction. (Lines 11-21)
6. Set the initial step size as one. Modify the direction and initial step properly in order to avoid unfeasible regions. (Lines 23-24)
7. If the norm of the direction or step size are too small, discard the current iterate and continue in step 5, else, perform a search in the given direction. (Lines 25-26)
8. If the new solution is not better than the original and the Jacobian is available, compute an optimal step by approximating a second order polynomial. (Line 27)
9. Exchange the new solution for the one in the population if it is closer to the current reference point and continue the search in the given direction until the maximum iterations are reached or there is no further gain, else, continue in step 5. (Lines 28-30)
10. Stop the process if the number of sought improved solutions is reached or if the set of improvable reference points is empty. Go to Step 3 in any other case. (Line 33)

5 Numerical Results

In the following we will present some numerical results where we concentrate on models with small number of objectives (up to five). Table 1 shows the parameter values we

Algorithm 1. Algorithm RDS-R-NSGA-II

Require: Population P, Ref. pts $Z_1, ..., Z_l$, Feasible ref. pts. so far $i \in \{1, ..., l\}$, local search max iterations lsd, solutions to improve nsi, last reference point used lz, maximum tests on a ref. pt. $maxTst$
Ensure: Population with at most nsi improved solutions P', index of last reference point lz.
For each Z_i sort P using a chosen norm.
Set $solsImproved := 0$; $done :=$ **false** ; $improvableRefPts := \{1, ..., l\}$.
Remove feasible reference points indices from $improvableRefPts$.
while not $done$ **do**
  if $lz \notin improvableRefPts$ **and** $improvableRefPts \neq \emptyset$ **then**
    $lz := (lz\%l) + 1$, Continue;
  end if
  Set $\nu := \{\}$; $tstCntr := 0$
  for $j := 1$ **to** $|P|$ **do**
    if $tstCntr > maxTst$ **then** remove lz from $improvableRefPts$, break for loop **else** set $tstCntr := tstCntr + 1$ **end if**
    Set x_0 and F_0 as the decision and objective vector respectively of the j-th closest solution to Z_{lz} and $lzIters := 0$.
    if $\nu = \{\}$ **then**
      Compute the Jacobian $J \in \mathbb{R}^{k \times n}$
      if $k = 2$ **and** $\angle(\nabla f_1, \nabla f_2) < 20\circ$ **then**
        Compute $\nu = -\frac{1}{2}(\frac{\nabla f_1}{||\nabla f_1||} + \frac{\nabla f_2}{||\nabla f_2||})$. {Lara descent direction}
      else
        Set $d = Z_i - F_i$.
        Compute $\nu = J^+ d$ and set $\nu = \frac{\nu}{||\nu||}$ {DS descent direction}
      end if
      Set $gradAvailable :=$ **true**.
    end if
    while true do
      Set $t_0 := 1$.
      Call $BoxHandling$ to get ν_h and t_h.
      if t_h **or** $||\nu_h||$ are too small **then** break while loop **end if**
      Set $x_1 := x_0 + t_h nu_h$ and $F_1 := F(x_1)$.
      if $||F_1 - Z_{lz}|| > ||F_0 - Z_{lz}||$ **and** $gradAvailable$ **then** compute t_{opt} using F_0 , F_1 and J. Update x_1, F_1. **end if**
      if $||F_1 - Z_{lz}|| > ||F_0 - Z_{lz}||$ **or** $lzIters = lsd$ **then** break while loop. **end if**
      Exchange j-th individual's decision and objective vector with x_1 and F_1 respectively.
      Set $x_0 := x_1$, $F_0 := F_1$, $lzIters := lzIters + 1$ and $gradAvailable :=$ **false**.
    end while
  end for
  if $solsImproved = nsi$ **or** $improvableRefPts = \emptyset$ **then** set $done :=$ **true else** $lz := (lz\%l) + 1$.**end if**
end while

have chosen for each algorithm, R-NSGA-II and its hybrid RDS-R-NSGA-II. For a fair comparison, both algorithms have the same function budget. In addition, LS function calls are also subtracted from the function budget. In all cases we have only taken unfeasible reference points. In order to assess the performances, we have chosen to take the IGD indicator ([22]) applied to reference problems (denote by IGD_Z) which can be viewed as an averaged version of the distance measurement used in (4):

$$IGD_Z(F(A), Z) = \frac{1}{|Z|} \sum_{i=1}^{|Z|} \min_{j=1}^{|A|} dist(Z_i, F(a_j)) \qquad (IGD_Z)$$

Hereby, $A \subset Q$ is the given population or archive and $dist$ the chosen distance metric (for this, we have taken the Euclidean norm).

Table 1. Chosen parameter values for the different algorithms. Hereby, we use the following notation: N_{pop} denotes the population size, Gen the number of generations, P_X and P_M the probability for crossover and mutation, LS freq and LS dep the frequency and maximum iterations for local search, StI the number of solutions to improve and Max LSC the maximum number of LS Calls.

	R-NSGA-II							RDS			
	N_{pop}	Gen	P_X	P_M	ϵ	η_c	η_m	LS freq	LS dep	StI	Max LSC
F_1	100	100	0.9	0.033	5E-6	15	20	5	30	1	9
$ZDT1$	100	100	0.9	0.033	5E-6	15	20	10	30	5	4
$ZDT2$	100	100	0.9	0.033	5E-6	15	20	10	30	6	4
$ZDT3$	100	100	0.9	0.033	5E-4	15	20	5	30	2	10
$DTLZ2, k=3$	100	100	0.9	0.083	1E-3	15	20	5	30	1	5
$DTLZ2, k=5$	100	100	0.9	0.083	1E-3	15	20	5	30	2	5
$DTLZ3, k=3$	100	150	0.9	0.1	1E-3	15	20	15	30	3	8

First we consider the bi-objective problem CONV ([23])

$$f_1(x) = ||x - a_1||_2^2, \quad f_2(x) = ||x - a_2||_2^2, \qquad (17)$$

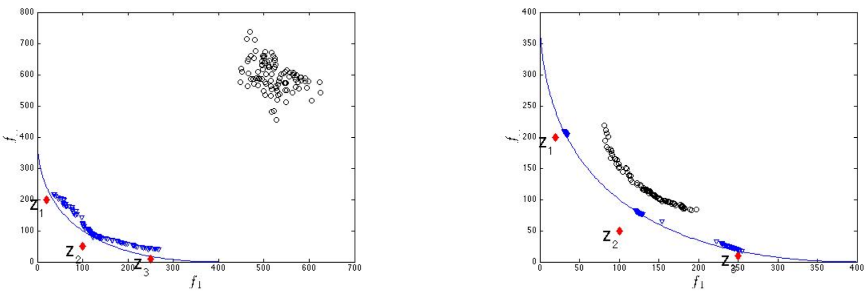
where $a_1 = (1, ..., 1)^T \in \mathbb{R}^{100}$ and $a_2 = -a_1$, and where $Q = \mathbb{R}^{100}$ is the domain. We chose the three reference points $Z = \{(20, 200)^T, (100, 50)^T(250, 10)^T\}$. Since CONV is a convex problem large improvements are expected via the help of local search. This is indeed the case as can be seen in Figure 2 as well as in Table 2, where the IGD_Z values for all the examples considered in this study are presented obtained after 20, 40 and 100 percent of the given function budget.

Next, we consider MOP ZDT1 from the ZDT benchmark suite ([24]) using $n = 30$ (as well as for all remaining MOPs) and $Z = \{(0.1, 0.6)^T, (0.5, 0.2)^T, (0.9, 0)^T\}$. Compared to CONV, RDS-R-NSGA-II needs more function evaluations on ZDT1 to obtain a covering of the optimal solutions near the reference points, but comes quite close after already 40 percent of the given function budget (compare to Figure 3 and Table 2).

ZDT2 represents a challenge for evolutionary algorithms since typically in first stages of the search all solutions concentrate on the left top area of the objective space. Here, we can see in Figure 4 that RDS-R-NSGA-II achieves a focus on the desired areas here given by $Z = \{(0.1, 0.9)^T, (0.8, 0.2)^T, (0.6, 0.5)^T\}$ after 40 percent of the function budget. At the end of the given function budget, R-NSGA-II still performs a movement along the Pareto front while its memetic variant is covering the optimal areas.

ZDT3 has a more complex geometry than the previous ones, where not all boundary solutions are optimal. Here we have chosen $Z = \{(0.8, -0.6)^T, (0.1, 0.6)^T, (0.35, 0.1)^T\}$. At twenty percent of function budget, the memetic strategy is outperforming R-NSGA-II (compare to Figure 5). However, it has to be noted that some optimal solutions near Z_1 could hardly be reached since a directed search may lead outside the feasible region. Nevertheless, at hundred percent solutions of both algorithms are very close together.

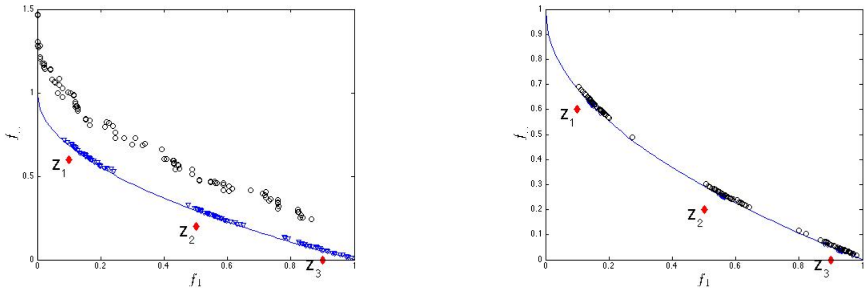
Analogously, we have tested the two methods on two three-objective problems—DTLZ2 and DTLZ3 taken from [25], see Figures 6 and 7—as well as on DTLZ2 with $k = 5$ objectives yielding a similar evolution of the performance.



(a) 20% of budget (b) 100% of budget

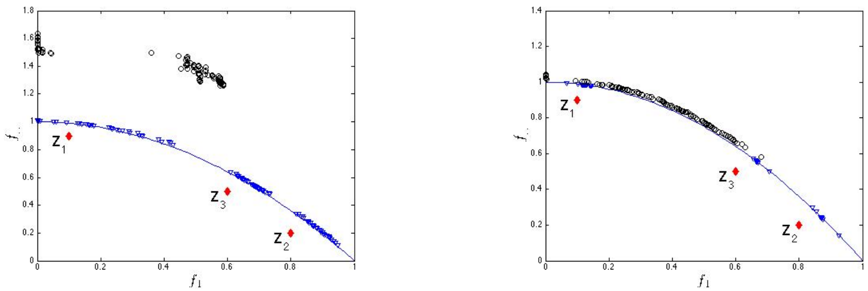
Fig. 2. Numerical results of R-NGSA-II (black circles) and RDS-R-NSGA-II (blue triangles) on CONV

Considering these seven test examples we see from Table 2 that RDS-R-NSGA-II wins over its base algorithm in seven out of seven cases when we stop the search after 20 percent of the function budget and still in all 7 cases after 40 percent. The more the search continues, the more both methods converge toward the optimal regions. Hence, both methods get solutions near to the optimum after the given function budget. However, it is apparent that the memetic strategy yields faster convergence. In fact, using RDS-R-NSGA-II, solutions that are nearly optimal can already be reached after 40 percent of the given budget in most cases. Thus, one can say that RDS-R-NSGA-II accomplishes its task.



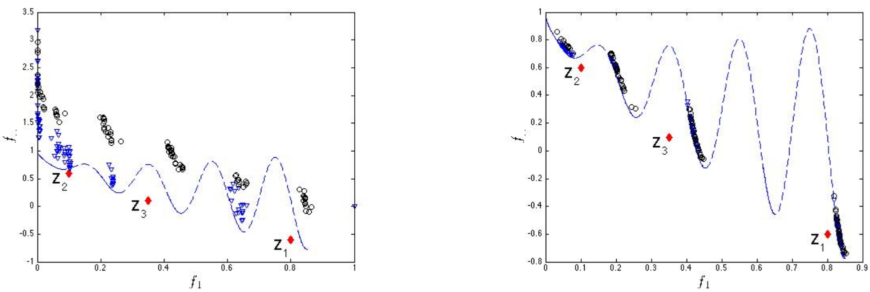
(a) 40% of budget (b) 100% of budget

Fig. 3. Numerical results of R-NGSA-II (black circles) and RDS-R-NSGA-II (blue triangles) on ZDT1



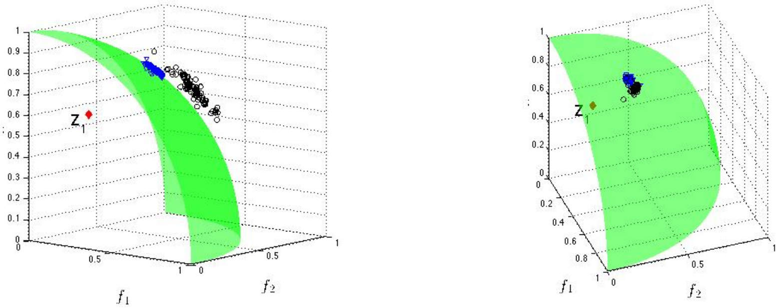
(a) 40% of budget (b) 100% of budget

Fig. 4. Numerical results of R-NGSA-II (black circles) and RDS-R-NSGA-II (blue triangles) on ZDT2



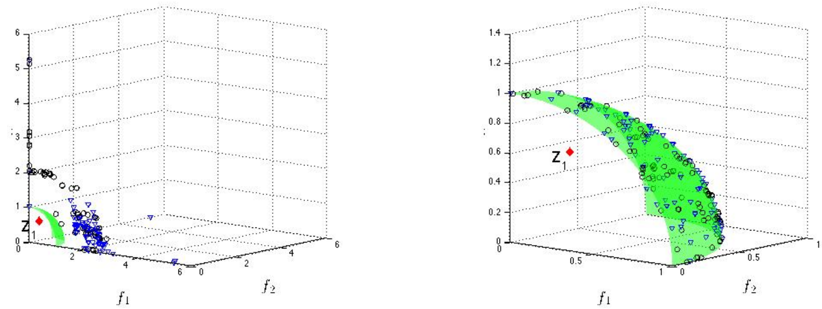
(a) 20% of budget (b) 100% of budget

Fig. 5. Numerical results of R-NGSA-II (black circles) and RDS-R-NSGA-II (blue triangles) on ZDT3



(a) 40% of budget (b) 100% of budget

Fig. 6. Numerical results of R-NGSA-II (black circles) and RDS-R-NSGA-II (blue triangles) on DTLZ2



(a) 20% of budget (b) 100% of budget

Fig. 7. Numerical results of R-NGSA-II (black circles) and RDS-R-NSGA-II (blue triangles) on DTLZ3

Table 2. IGD_Z values for the two different algorithms on the considered reference point problems. The results are averaged over 30 independent runs.

	RDS-R-NSGA-II			R-NSGA-II		
	mean (std. dev)					
	20	40	100	20	40	100
F1	31.107 (2.019)	21.633 (1.386)	20.190 (0.627)	559.465 (43.855)	221.012 (19.335)	74.720 (4.789)
ZDT1	0.177 (0.118)	0.062 (0.010)	0.056 (0.000)	0.565 (0.063)	0.224 (0.041)	0.063 (0.002)
ZDT2	0.164 (0.063)	0.091 (0.013)	0.085 (1E-4)	1.346 (0.212)	0.734 (0.264)	0.1764 (0.112)
ZDT3	0.247 (0.124)	0.099 (0.037)	0.068 (0.019)	0.416 (0.051)	0.137 (0.049)	0.069 (0.011)
DTLZ2 $k = 3$	0.359 (0.021)	0.358 (0.014)	0.344 (0.005)	0.583 (0.043)	0.427 (0.030)	0.348 (0.004)
DTLZ2 $k = 5$	0.475 (0.025)	0.463 (0.012)	0.473 (0.011)	0.675 (0.048)	0.503 (0.021)	0.474 (0.011)
DTLZ3	0.530 (0.310)	0.374 (0.029)	0.346 (0.009)	0.910 (0.598)	0.417 (0.186)	0.346 (0.007)

6 Conclusions and Future Work

In this paper, we have addressed the numerical treatment of reference point problems by means of a memetic strategy. For this, we have discussed and adapted the recently proposed Directed Search (DS) method that allows to steer the search into any direction given in objective space. Further, we have made a first attempt to integrate the DS into R-NSGA-II, a state-of-the-art evolutionary algorithm designed for reference point problems. First results where we have focussed on multi-objective problems with a low number of objectives ($2 \leq k \leq 5$) indicate that such a hybrid leads to a significant speed up of the computations in almost all cases.

Though the first results are very promising, there are, however, many points to be addressed in the future. This includes the more thorough discussion of the DS for the problem at hand as well as an improvement of its numerical realization. Next, the interplay of local and global search can be further enhanced for a better performance in particular for multi-modal problems. Finally, we will consider problems with more objectives since in that case the advantage over classical methods—i.e., the approximation of the entire solution set—will become more significant.

Acknowledgement. The first author acknowledges support from CONACyT through a scholarship to pursue undergraduate studies at the Computer Science Department of CINVESTAV-IPN. The second author acknowledges support from CONACyT project no. 128554.

References

[1] Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons, Chichester (2001) ISBN 0-471-87339-X

[2] Beume, N., Naujoks, B., Emmerich, M.: SMS-EMOA: Multiobjective selection based on dominated hypervolume. European Journal of Operational Research (2006)

[3] Coello Coello, C.A., Lamont, G.B., Van Veldhuizen, D.A.: Evolutionary Algorithms for Solving Multi-Objective Problems, 2nd edn. Springer, New York (2007) ISBN 978-0-387-33254-3

[4] Dellnitz, M., Schütze, O., Hestermeyer, T.: Covering Pareto sets by multilevel subdivision techniques. Journal of Optimization Theory and Applications 124, 113–155 (2005)

[5] Jahn, J.: Multiobjective search algorithm with subdivision technique. Computational Optimization and Applications 35(2), 161–175 (2006)

[6] Schütze, O., Vasile, M., Junge, O., Dellnitz, M., Izzo, D.: Designing optimal low thrust gravity assist trajectories using space pruning and a multi-objective approach. Engineering Optimization 41(2), 155–181 (2009)

[7] Hernández, C., Naranjani, Y., Sardahi, Y., Liang, W., Schütze, O., Sun, J.Q.: Simple cell mapping method for multi-objective optimal feedback control design. International Journal of Dynamics and Control 1(3), 231–238 (2013)

[8] Hernández, C., Sun, J.-Q., Schütze, O.: Computing the set of approximate solutions of a multi-objective optimization problem by means of cell mapping techniques. In: Emmerich, M., et al. (eds.) EVOLVE - A Bridge between Probability, Set Oriented Numerics,and Evolutionary Computation IV. AISC, vol. 227, pp. 171–188. Springer, Heidelberg (2013)

[9] Das, I., Dennis, J.: Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. SIAM Journal of Optimization 8, 631–657 (1998)
[10] Miettinen, K.: Nonlinear Multiobjective Optimization. Kluwer Academic Publishers, Boston (1999)
[11] Fliege, J.: Gap-free computation of Pareto-points by quadratic scalarizations. Mathematical Methods of Operations Research 59, 69–89 (2004)
[12] Eichfelder, G.: Adaptive Scalarization Methods in Multiobjective Optimization. Springer, Heidelberg (2008) ISBN 978-3-540-79157-7
[13] Ignizio, J.: Goal programming and extensions. Lexington books. Lexington Books (1976)
[14] Wierzbicki, A.P.: The use of reference objectives in multiobjective optimization. In: Fandel, G., Gal, T. (eds.) Multiple Criteria Decision Making Theory and Application. Lecture Notes in Economics and Mathematical Systems, vol. 177, pp. 468–486. Springer, Heidelberg (1980)
[15] Deb, K., Sundar, J., Udaya Bhaskara Rao, N., Chaudhuri, S.: Reference point based multi-objective optimization using evolutionary algorithms. International Journal of Computational Intelligence Research, 635–642 (2006)
[16] Schütze, O., Lara, A., Coello, C.A.C.: The directed search method for unconstrained multi-objective optimization problems. In: Proceedings of the EVOLVE – A Bridge Between Probability, Set Oriented Numerics, and Evolutionary Computation (2011)
[17] Pareto, V.: Manual of Political Economy. The MacMillan Press (1971 (original edition in French in 1927))
[18] Lara, A., Alvarado, S., Salomon, S., Avigad, G., Coello, C.A.C., Schütze, O.: The gradient free directed search method as local search within multi-objective evolutionary algorithms. In: Schütze, O., Coello Coello, C.A., Tantar, A.-A., Tantar, E., Bouvry, P., Del Moral, P., Legrand, P. (eds.) EVOLVE - A Bridge Between Probability, Set Oriented Numerics, and Evolutionary Computation II. AISC, vol. 175, pp. 153–168. Springer, Heidelberg (2012)
[19] Allgower, E.L., Georg, K.: Numerical Continuation Methods. Springer (1990)
[20] Mejia, E., Schütze, O.: A predictor corrector method for the computation of boundary points of a multi-objective optimization problem. In: International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE 2010), pp. 395–399 (2010)
[21] Hernández, V.A.S., Schütze, O., Rudolph, G., Trautmann, H.: The directed search method for pareto front approximations with maximum dominated hypervolume. In: Emmerich, M., et al. (eds.) EVOLVE - A Bridge between Probability, Set Oriented Numerics,and Evolutionary Computation IV. AISC, vol. 227, pp. 189–205. Springer, Heidelberg (2013)
[22] Coello Coello, C.A., Cruz Cortés, N.: Solving Multiobjective Optimization Problems using an Artificial Immune System. Genetic Programming and Evolvable Machines 6(2), 163–190 (2005)
[23] Köppen, M., Yoshida, K.: Substitute distance assignments in NSGA-II for handling many-objective optimization problems. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 727–741. Springer, Heidelberg (2007)
[24] Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary Computation 8, 173–195 (2000)
[25] Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable test problems for evolutionary multiobjective optimization. In: Abraham, A., Jain, L., Goldberg, R. (eds.) Evolutionary Multiobjective Optimization. Theoretical Advances and Applications, pp. 105–145. Springer, USA (2005)