Engineering Optimization

ISSN: 0305-215X (Print) 1029-0273 (Online) Journal homepage: http://www.tandfonline.com/loi/geno20

RDS-NSGA-II: a memetic algorithm for reference point based multi-objective optimization

Jesus Alejandro Hernández Mejía, Oliver Schütze, Oliver Cuate, Adriana Lara & Kalyanmoy Deb

ENGINEERING OPTIMIZATION, 2016
http://dx.doi.org/10.1080/0305215X.2016.1211127

Taylor & Francis
Taylor & Francis Group

RDS-NSGA-II: a memetic algorithm for reference point based multi-objective optimization

Jesus Alejandro Hernández Mejía[a], Oliver Schütze[a], Oliver Cuate[a], Adriana Lara[b] and Kalyanmoy Deb[c]

[a]Computer Science Department, CINVESTAV-IPN, Mexico City, México; [b]ESFM del Instituto Politécnico Nacional, Mexico City, México; [c]College of Engineering, Michigan State University, East Lansing, USA

ABSTRACT
Reference point based optimization offers tools for the effective treatment of preference based multi-objective optimization problems, *e.g.* when the decision-maker has a rough idea about the target objective values. For the numerical solution of such problems, specialized evolutionary strategies have become popular, despite their possible slow convergence rates. Hybridizing such evolutionary algorithms with local search techniques have been shown to produce faster and more reliable algorithms. In this article, the directed search (DS) method is adapted to the context of reference point optimization problems, making this variant, called RDS, a well-suited option for integration into evolutionary algorithms. Numerical results on academic test problems with up to five objectives demonstrate the benefit of the novel hybrid (*i.e.* the same approximation quality can be obtained more efficiently by the new algorithm), using the state-of-the-art algorithm R-NSGA-II for this coupling. This represents an advantage when treating costly-to-evaluate real-world engineering design problems.

ARTICLE HISTORY
Received 10 June 2015
Accepted 1 July 2016

KEYWORDS
Multi-objective optimization; reference point problem; memetic strategy

1. Introduction

Multi-objective optimization is concerned with the simultaneous optimization of multiple objectives. Such multi-objective optimization problems (MOPs) frequently arise, for example, in engineering design. The knowledge of the solution set, the so-called Pareto set, of a given MOP is of great interest for the decision-maker since it gives them an overview of the possible optimal realizations of the project. For the computation of the Pareto set, respectively its image, the Pareto front, many algorithms can be found in the literature. Among them, there are set oriented methods such as multi-objective evolutionary algorithms (MOEAs) (*e.g.* Coello, Lamont and Van Veldhuizen [2007], Deb [2001]), subdivision techniques (*e.g.* Dellnitz, Schütze and Hestermeyer [2006], Jahn [2006]), or cell mapping techniques (*e.g.* Hernández *et al.* [2013], Hernández, Sun and Schütze [2013]). These methods have in common that they allow for the approximation of the entire Pareto set/front in one run of the algorithm, and that they are further characterized by a great robustness and minimal assumptions over the model. There is, however, one important characteristic of MOPs that hinders getting an approximation of the entire solution set for all problems, namely that the Pareto set/front typically forms a $(k-1)$-dimensional object, where k is the number of objectives involved in the MOP. That is, with increasing k it soon becomes impossible to obtain a suitable overview of the entire solution set. Instead, one can try to compute single optimal solutions—*e.g.*

CONTACT Oliver Schütze schuetze@cs.cinvestav.mx

via scalarization methods that transform the MOP into a scalar optimization problem (SOP) (as in Das and Dennis [1998], Eichfelder [2008], Eichfelder [2008] and Miettinen [1999])—or to focus the search on particular regions of the solution set (as in Branke and Deb [2005], Branke *et al.* [2004], Mattson, Mullur and Messac [2004], Trautmann *et al.* [2013], Wagner and Trautmann [2010] and Zhang, Tian and Jin [2015]).

If the decision-maker already has a rough idea about the target design (*e.g.* if a new car has to be placed in a certain market segment), they might utilize reference point methods (*e.g.* Ignizio [1976], Miettinen [1999] or Wierzbicki [1980]). Methods of that kind try to compute (optimal) solutions that are as close as possible to a given reference point Z defined in objective space. Similar to the computation of the entire solution set, set oriented methods—such as evolutionary algorithms—are beneficial for the treatment of reference point based optimization problems (RPPs) since their set based approach allows (i) handling more than one reference point in each run, (ii) exploring next to the (single) solutions x^* of those RPPs also the neighbourhood of x^*, and (iii) successfully addressing multi-modal problems. One state-of-the-art evolutionary algorithm that can cope with these issues and which will be used in this study is the reference point non-dominating sorting genetic algorithm (R-NSGA-II) proposed by Deb and Sundar (2006). Another efficient evolutionary algorithm for RPPs is RMEAD proposed by Mohammadi, Omidvar and Li (2012). One issue of R-NSGA-II, however, is that for some problems quite a few function evaluations have to be spent before an acceptable approximation of the given RPP is reached, a characteristic which is shared with most evolutionary algorithms. For the general case, several hybridizations of MOEAs with local search techniques have been proven to be effective in this regard (*e.g.* Gil *et al.* [2007] and Kim and Liou [2014]). Since function evaluations can be quite costly (*e.g.* Kipouros [2013] and Oliver, Kipouros and Savill [2013] in airfoil design), savings in the total number of them are thus desired.

The directed search (DS) method, introduced by Schütze *et al.* (2016), is proposed here as well suited to hybridization with an evolutionary RPP solver such as R-NSGA-II. The DS method is a point-wise iterative search procedure that allows the search to be steered from a point x in parameter space in any given direction d in objective space. In the context of RPPs, such a desired direction is always given: the most greedy direction is given by $d_G = F(x) - Z$, where $F(x)$ denotes the image of x. Thus, the potential of DS for its hybridization with specialized evolutionary algorithms. Another aspect (which is not exploited here) is that DS can be used with or without gradient information. The latter can be done by utilizing neighbourhood information, which is typically given for set based optimization strategies. An adaptation of DS to the context of RPPs is discussed in this study; also, how the resulting algorithm, RDS, can be integrated into R-NSGA-II. The advantage of the novel hybrid, called RDS-NSGA-II, over its base algorithm in terms of convergence speed is shown by numerical results. A preliminary study of this work was presented by Mejía, Schütze and Deb (2014). The remainder of this article is organized as follows: in Section 2, the background is set. In Section 3, the adaptations that have to be made to the DS method to treat RPPs efficiently is discussed. In Section 4, how to integrate the new local searcher, RDS, into R-NSGA-II is described. Next, some numerical results on several MOPs are presented in Section 5. Finally, conclusions and paths for future research are stated in Section 6.

2. Background

In the following, continuous MOPs problems are considered as

$$\min_{x \in Q} F(x), \tag{1}$$

where $Q \subset \mathbb{R}^n$ is the domain and $F : Q \to \mathbb{R}^k$ is defined as the vector of the objective functions $F(x) = (f_1(x), \ldots, f_k(x))^{\mathrm{T}}$. The optimality of an MOP is based on the concept of dominance.

Definition 2.1:

(a) *Let* $v, w \in \mathbb{R}^k$. *Then the vector* v *is less than* w ($v <_p w$), *if* $v_i < w_i$ *for all* $i \in \{1, \dots, k\}$. *The relation* $\leq_p$ *is defined analogously.*
(b) *A vector* $y \in D$ *is dominated by a vector* $x \in D$ ($x \prec y$) *with respect to Equation (1) if* $F(x) \leq_p F(y)$ *and* $F(x) \neq F(y)$; *otherwise* y *is called non-dominated by* x.
(c) *A point* $x \in D$ *is called (Pareto) optimal or a Pareto point if there is no* $y \in D$ *that dominates* x.
(d) *The set* P_D *of all Pareto optimal solutions is called the Pareto set and its image* $F(P_D)$ *the Pareto front.*

One important characteristic of MOPs is that both the Pareto set and front typically form $(k-1)$-dimensional objects (Hillermeier, 2001). That is, a suitable finite size approximation of the entire solution set may become a challenging task, if not impossible, as the number of objectives increases. One possible remedy is to use reference point based optimization methods; *e.g.* if one (or several) reference points $Z \in \mathbb{R}^k$ are given that represent the desired outcome for the decision-maker, one can try to find the Pareto optimal solution whose distance is closest to Z, which leads to the following SOP:

$$\min_{x \in P_Q} D(F(x), Z), \tag{2}$$

where D denotes a distance metric. In this study, of particularly interest are RPPs of this form using the 2-norm for D. In the literature, however, many other variants can be found (*e.g.* Ignizio [1976], Miettinen [1999] and Wierzbicki [1980]).

One state-of-the-art evolutionary algorithm to tackle RPPs of the form (2) is R-NSGA-II (Deb and Sundar, 2006), which is a variant of the MOEA NSGA-II (Deb *et al.*, 2002). In order to solve RPPs of the form (2), RNSGA-II replaces NSGA-II's crowding distance computation as follows: for each reference point (RP), a sorting procedure is performed on the population using (2). The closest solution is then assigned to hierarchy one, the second closest one to hierarchy two, and so on. Finally, a solution will be assigned as the crowding distance to the lowest hierarchy it had in the sorting procedures. A parameter ϵ is introduced to control the spread of solutions in the final archive.

The generic constraint-domination relation proposed in Deb *et al.* (2002) can easily be included to tackle constrained problems. The constraint handling R-NSGA-II remains the same except for the replacement of the domination relation for the constraint domination. This new relation requires the constraint violation value defined as follows.

Definition 2.2 —Constraint violation: *The constraint violation of a solution* x *is defined as*

$$C_V(x) = \sum_{i=1}^{p} \max\,(0, g_i(x)) + \sum_{j=1}^{q} |h_j(x)|, \tag{3}$$

where g_i, $i = 1, \dots, p$, *and* h_j, $j = 1, \dots, q$, *denote the inequality and equality constraints, respectively.*

Thus, feasible solutions have a constraint violation value of zero, while the value is positive for infeasible solutions.

Definition 2.3 —Constraint domination (Deb, 2001): *A solution* $x \in \mathbb{R}$ *is said to constraint dominate a solution* $y \in \mathbb{R}$ *if any of the following conditions are true:*

(i) x *and* y *are infeasible and* $C_V(x) < C_V(y)$,
(ii) x *is feasible and* y *is infeasible, or*
(iii) x *and* y *are feasible and* $x \prec y$.

R-NSGA-II can be said to optimize two different functions during the evolution process. When only infeasible solutions are present, the algorithm becomes an SOP solver that directs the search towards less constraint violating regions and once feasible solutions are found the original RPP takes over as the main objective. The local searcher used and adapted here to the given context is DS, which allows the search to be steered in any desired direction in objective space. More precisely, given a point $x \in \mathbb{R}^n$ in decision space and a direction $d \in \mathbb{R}^k$ in objective space, a search direction $\nu \in \mathbb{R}^n$ is sought such that

$$\lim_{t \searrow 0} \frac{f_i(x_0 + t\nu) - f_i(x_0))}{t} = d_i, \quad i = 1, \ldots, k. \tag{4}$$

Note that ν solves the following linear system:

$$J(x_0)\nu = d, \tag{5}$$

where $J(x)$ denotes the Jacobian of F at x. Since typically $k \ll n$, the equation system (5) is typically underdetermined. Among the solutions of Equation (5), the one with the least 2-norm is given by

$$\nu_+ = J(x)^+ d, \tag{6}$$

where $J(x)^+$ denotes the pseudo inverse of $J(x)$. ν_+ can be seen as the most greedy solution since it yields the largest decay in the d-direction when used in numerical realizations. A movement in the d-direction via DS is thus related to the (numerical) solution of the following initial value problem (IVP):

$$\begin{aligned} x(0) &= x_0 \in \mathbb{R}^n \\ \dot{x}(t) &= J(x)^+ d, t > 0. \end{aligned} \tag{7}$$

3. RDS: the DS version for RPPs

The adaptations of the DS method for RPPs of the form

$$\min_{x \in Q} \frac{1}{2} \|Z - F(x)\|_2^2 \tag{8}$$

are discussed here. First, the basic idea for the unconstrained case is presented, and a discussion follows afterwards concerning possible adaptations for constrained MOPs.

3.1. Basic idea

Given a point x_0 and a reference point Z, the most greedy search direction in objective space is certainly given by

$$d_G = Z - F(x_0). \tag{9}$$

Using this direction together with the DS method, a movement toward Z thus leads to the solution of the following IVP:

$$\begin{aligned} x(0) &= x_0 \in \mathbb{R}^n \\ \dot{x}(t) &= J(x(t))^+ (Z - F(x(t))), t > 0. \end{aligned} \tag{10}$$

The following results show that $\dot{x}(t)$ is a descent direction of RPP (8) and that the solution of IVP (10) always leads to critical points of RPP (8).

Proposition 3.1: *Let $x \in \mathbb{R}^n$ with $F(x) \neq Z$ and $\nabla g(x) \neq 0$, where g denotes the objective in* (8)*. Then $v := J(x)^{+}(Z - F(x))$ is a descent direction of RPP* (8) *at x.*

Proof: It holds that $\nabla g(x_c) = -J(x)^{\mathrm{T}}(Z - F(x)) \neq 0$ and

$$\begin{aligned} \langle v, \nabla g(x) \rangle &= -\langle J(x)^{\mathrm{T}}(Z - F(x)), J(x)^{+}(Z - F(x)) \rangle \\ &= -(Z - F(x))^{\mathrm{T}} \underbrace{J(x)J(x)^{+}}_{=I}(Z - F(x)) = -\|Z - F(x)\|_2^2 < 0. \end{aligned} \tag{11}$$

∎

Proposition 3.2: *An end point of IVP (10) is a critical point of RPP (8).*

Proof: For a critical point x_c of (8) it holds that

$$\nabla g(x_c) = J(x_c)^{\mathrm{T}}(Z - F(x_c)) = 0. \tag{12}$$

An end point $x(t_e)$ of IVP (10) is characterized by $\dot{x}(t_e) = 0$, which is the case if one of the following three conditions is fulfilled:

(1) $J(x(t_c))^{+} = 0$,
(2) $Z - F(x(t_e)) = 0$, or
(3) $J(x(t_c))^{+}(Z - F(x(t_e))) = 0$.

In case (1), it is $J^{\mathrm{T}} = 0$, where $J := J(x(t_e))$, and condition (12) is satisfied. In case (2), it is $Z = F(x(t_e))$, which means that $x(t_e)$ is a global solution of (8), and thus a critical point. In case (3), let $U\Sigma^{+}V^{\mathrm{T}}$ be a singular value decomposition of J. Since $J^{+}(Z - F(x(t_e))) = V\Sigma^{+}U^{\mathrm{T}}(Z - F(x(t_e)))$ it follows that $J^{\mathrm{T}}(Z - F(x(t_e))) = V\Sigma U^{\mathrm{T}}(Z - F(x(t_e))) = 0$, which means that (12) is satisfied, and the proof is complete. ∎

It is worth noticing that the solution of IVP (10) might become computationally expensive in a case where the reference point is not feasible, since the ordinary differential equation (ODE) might be stiff. To be more precise, a solution curve of (10) can be divided into the two following phases, given that $\mathrm{rank}(J(x_0)) = k$.

Phase I. If $\mathrm{rank}(J(x(t))) = k$ for a $t > 0$, then the direction $v := \dot{x}(t)$ satisfies $J(x(t))v = d_G$. That is, the movement is performed in the direction d_G.

Phase II. If $\mathrm{rank}(J(x(t))) < k$ for a $t > 0$—which is, for example, the case for points x, where $F(x)$ is on the boundary of the image of F such as Pareto optimal solutions—it is not guaranteed to find a direction v that satisfies $J(x(t))v = d_G$. Instead, the direction v is a 'best fit' (which follows directly from the properties of the pseudoinverse; see Nocedal and Wright [2006]), *i.e.*

$$v = J(x(t))^{+}d = \arg\min_{v \in \mathbb{R}^n} \|J(x(t))v - d\|. \tag{13}$$

Note that the ODE is stiff in Phase II since $\mathrm{rank}(J(x(t)))^{+} = \mathrm{rank}(J(x(t))) < k$. To overcome this problem, it is possible to switch from IVP (10) to another, related, IVP that performs the same movement as in (13) as follows: let x be a critical point where it is assumed that $\mathrm{rank}(J(x)) = k - 1$. That is, there exists an $\alpha \in \mathbb{R}\backslash\{0\}$ with

$$\sum_{i=1}^{k} \alpha_i \nabla f_i(x) = 0, \tag{14}$$

and let $\alpha = QR(q_1, \ldots, q_k)R$ be a QR factorization of α. Then the column vectors of

$$Q_2 = (q_2, \ldots, q_k) \in \mathbb{R}^{k\times(k-1)} \tag{15}$$

form an orthonormal basis of the linearized boundary of the image at $F(x)$. Given the desired direction d_Z, the projection onto the tangent space is thus given by

$$d_{\text{new}} = Q_2\underbrace{(Q_2^{\mathrm{T}} Q_2)}_{=I}{}^{-1} Q_2^{\mathrm{T}} d_G = Q_2 Q_2^{\mathrm{T}} d_G. \tag{16}$$

Denote by

$$P(x) := Q_2(x)Q_2(x)^{\mathrm{T}} \tag{17}$$

the projection described above. Then, the suggestion is to solve the following IVP for points x with $\text{rank}(J(x)) < k$:

$$\begin{aligned} &x(0) = x_0 \in \mathbb{R}^n \\ &\dot{x}(t) = J(x(t))^{+} P(x(t))(Z - F(x(t))) = J(x(t))^{+} d_{\text{new}}(t), t > 0. \end{aligned} \tag{18}$$

By this, an IVP is obtained whose solution curves are identical to those of IVP (10). The difference between them is that, for IVP (18), there exists a vector ν such that $J(x(t))\nu = d_{\text{new}}$. The results of Propositions 3.1 and 3.2 hold analogously for IVP (18).

3.2. Numerical realization

Integration

It has been chosen to use the explicit Euler method for the numerical integration of the ODEs in (10) and (18). That is, an iteration step for the integration of the ODE in (10) reads as

$$x_{i+1} = x_i + t_i J(x_i)^{+}(Z - F(x_i)), \tag{19}$$

where $t_i > 0$ is the chosen step length. An iteration step for the ODE in (18)) is analog. Other, more sophisticated, integration schemes might be taken. One appealing advantage of the Euler method, however, (local) quadratic convergence is expected 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 following root finding problem:

$$\xi : \mathbb{R}^n \to \mathbb{R}^k, \xi(x) = F(x) - Z. \tag{20}$$

An iteration of the Gauss–Newton method (which converges locally quadratically, see Allgower and Georg 1990) is given by

$$x_{i+1} = x_i - J_\xi(x_i)^{+} g(x_i) = x_i - J(x_i)^{+}(F(x_i) - Z) = x_i + J(x_i)^{+}(Z - F(x_i)), \tag{21}$$

where $J_\xi(x)$ denotes the Jacobian of ξ at x. Comparing (21) and (19) it can be seen that both iterations coincide for the step length $t_i = 1$.

The 'switch' between the integration of IVP (10) and IVP (18) can be done by monitoring the condition number of $J(x_i)$ at the current iterate x_i in the following manner: If $\kappa(J(x_i)) < tol_\kappa$ for a threshold $tol_\kappa > 0$, then IVP (10) is chosen, else IVP (18). For the step size, $t_i^0 = 1$ has been chosen as the initial choice based on the above discussion together with an Armijo-like backtracking (see Nocedal and Wright, 2006) applied to g in case this step was too large.

Stopping criteria

By the proof of Proposition 3.2 it follows that a critical point x_c of RPP (8) is reached when $Z = F(x_c)$ or $J(x_c)^{\mathrm{T}}(Z - F(x_c)) = 0$. Thus, the iteration is stopped at an iterate x_i if $\|Z - F(x_i)\| \leq tol_1$ or $\|J(x_i)^{\mathrm{T}}(Z - F(x_i))\| \leq tol_2$, where $tol_1 > 0$ and $tol_2 > 0$ are given tolerance values, or a prescribed number of iterations is reached.

The pseudocode of the reference point directed search (RDS) for unconstrained MOPs is presented in Algorithm 1. It puts together the discussions made above.

Algorithm 1 RDS for unconstrained MOPs

Require: reference point Z, tolerance values tol_κ, tol_1, tol_2, initial point x_0

$i := 0$
repeat
  $d_G := Z = F(x_i)$
  if $\kappa(J(x_i)) < tol_\kappa$ **then** ▷ Phase I
    compute $t_i^I \in \mathbb{R}_+$
    $\nu_i^I := J(x_i)^+ d_G$
    $x_{i+1} = x_i^I + t_i^I \nu_i^I$
  else ▷ Phase II
    compute Q_2 as in Equation (15)
    $d_{\text{new}} = Q_2 Q_2^{\mathrm{T}} d_G$
    compute $t_i^{\mathrm{II}} \in \mathbb{R}_+$
    $\nu_i^{\mathrm{II}} := J(x_i)^+ d_{\text{new}}$
    $x_{i+1} = x_i + t_i^{\mathrm{II}} \nu_i^{\mathrm{II}}$
  end if
  $i := i + 1$
until $\|d_G\| < tol_1$ **or** $\|J(x_i)^{\mathrm{T}} d_G < tol_2\|$

3.3. Handling constraints

Box constraints

For the case when a box-constrained problem is considered in the form

$$\begin{aligned} &\min_{x \in Q} F(x), \\ &\text{s.t.} l \leq x \leq u, \end{aligned} \tag{22}$$

where l and u are the lower and upper bounds, respectively, it is convenient to borrow elements from gradient projection (see Rosen [1960]) to change the search direction as follows. For a point $x(t)$ compute the search direction $\nu = x(t)$ as in (10) or (18). If none of the constraints are violated for the search direction, then continue with ν as for the unconstrained case. If any of the constraints are violated, the respective inequality constraint can be handled as an equality constraint of the form

$$h_i(x) = \pm x_i + a_i, \tag{23}$$

where $a_i \in \mathbb{R}^n$. In this case a movement orthogonal to the gradient of h_i is sought, *i.e.* $\langle \nabla h_i(x), \nu \rangle = 0$. This property can easily be fulfilled since $\nabla h_i(x) = (0, \ldots, 0, \pm 1, 0, \ldots, 0)^{\mathrm{T}}$; then $\nu_i = 0$ can be chosen for all such active constraints h_i.

Equality constraints

If the MOP contains equality constraints $h_i : \mathbb{R}^n \to \mathbb{R}, i = 1, \ldots, m$, it is possible to use for the search direction $\nu = \dot{x}(t)$ the direction that performs the best fit direction in objective space. To be more precise, ν has to solve

$$\begin{aligned} &\min_{\nu} \|J(x)\nu - d\|_2^2 \\ &\text{s.t. } H(x)\nu = 0, \end{aligned} \tag{24}$$

where

$$H(x) = \begin{pmatrix} \nabla h_1(x)^{\mathrm{T}} \\ \vdots \\ \nabla h_m(x)^{\mathrm{T}} \end{pmatrix} \in \mathbb{R}^{m \times n}. \tag{25}$$

Since (24) is a quadratic programming problem, any solution (ν, λ) of (24), where λ denotes the Lagrange multiplier, solves the system

$$\begin{pmatrix} 2J(x)^{\mathrm{T}}J(x) & H(x)^{\mathrm{T}} \\ H(x) & 0 \end{pmatrix} \begin{pmatrix} \nu \\ \lambda^* \end{pmatrix} = \begin{pmatrix} J(x)^{\mathrm{T}}d \\ 0 \end{pmatrix}. \tag{26}$$

Note that the matrix on the left-hand side of (26) does not have full rank since $\mathrm{rank}(J(x)^{\mathrm{T}}J(x)) \leq k$. Proceeding as in (6), it is possible to compute the greedy solution via

$$\begin{pmatrix} \nu_+ \\ \lambda_+ \end{pmatrix} = \begin{pmatrix} 2J(x)^{\mathrm{T}}J(x) & H(x)^{\mathrm{T}} \\ H(x) & 0 \end{pmatrix}^+ \begin{pmatrix} J(x)^{\mathrm{T}}d \\ 0 \end{pmatrix} \tag{27}$$

and to replace the ODE in IVPs (10) and (18) by $\dot{x} = v_+(x(t))$, where $v_+(x(t))$ is computed via (27). The projection $P(x)$ for IVP (18) can be computed analogously to (17) since the weight vector α is also in this case orthogonal to the linearized boundary of the image at $F(x)$ (see Hillermeier [2001]). It is worth mentioning that one of the by-products of this approach is the computation of λ.

3.4. Neighbourhood exploration

Given an RPP, in many cases not only the optimal solution x^*—which is assumed here to be Pareto optimal for the sake of simplicity—of this scalar optimization problem is wanted. In addition, it might be interesting as well to explore the neighbourhood of x^* (respectively $F(x^*)$) to provide the decision-maker other promising solutions. While this neighbourhood exploration is a relatively easy task for bi-objective problems since there exist only neighbouring solutions 'left up' or 'right down' of the Pareto front, this does not hold for MOPs with more than two objectives since the solution set typically forms a set of dimension $k-1$. The proposal here is to utilize the steering feature of DS leading to evenly spaced solutions around $F(x^*)$. Given $J(x^*)$, the neighbourhood exploration can be performed without additional Jacobian calculations as follows.

Denote by e_i the ith canonical vector in $\mathbb{R}^k$, and let $\bar{d}_{i,1} = e_i$ and $\bar{d}_{i,2} = -e_i$ (*i.e.* both directions are being considered). Then the search via DS is performed in the directions that result from the projection of these vectors onto the linearized Pareto front at $F(x^*)$:

$$d_{i,j} = P(x^*)\bar{d}_{i,j}, \quad i = 1, \ldots, k, j = 1, 2. \tag{28}$$

In this search, different (small) step sizes $t \in T = \{t_1, \ldots, t_l\}$ can be used in order to get a broader view of the neighbourhood.

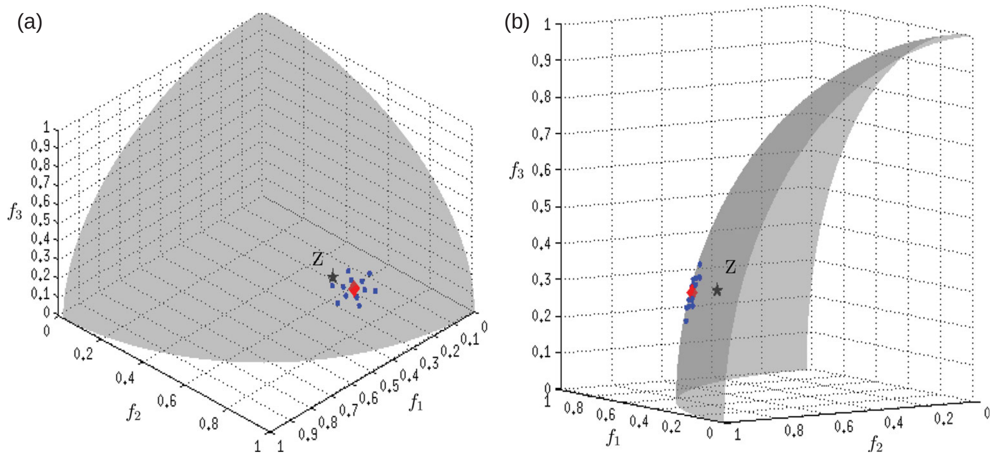

Figure 1. DS neighbourhood exploration on DTLZ2, regarding a particular reference point Z (two different angles are shown).

The pseudocode of the DS neighbourhood exploration is given in Algorithm 2.

Algorithm 2 DS neighbourhood exploration

Require: initial solution x^*, set of steps of size $T = \{t_1, \ldots, t_l\}$
Ensure: set N of neighbouring solutions of s^*
$N := \{x^*\}$
for $i = 1, \ldots, k$ **do**
  for $j = 1, 2$ **do**
    compute $d_{i,j}$ as in Equation (28)
    for $s = 1, \ldots, l$ **do**
      $N := N \cup \{x^* + t_s J(x^*)^+ d_{i,j}\}$
    end for
  end for
end for
return N

Figure 1 show the numerical result of this neighbourhood exploration on the MOPs known as DTLZ2 (Deb *et al.*, 2005). As can be seen, a set of well-spread solutions that are sufficiently close to the Pareto front is obtained.

It is important to notice that Algorithm 2 works particularly well for small step sizes. If the steps are chosen larger, the points $x^* + t_s J(x^*)^+ d_{i,j}$ do not have to be close enough to the Pareto set any more. In this case, it makes sense to *correct* these solutions back to the solution set, leading however to a higher computational cost.

4. Integrating RDS into R-NSGA-II

When designing a memetic algorithm—such as RDS-NSGA-II, which is proposed here—many aspects have to be taken into account in order to get a proper balance between the local search (LS) and the base evolutionary algorithm. The generic parameters suggested by Ishibuchi, Yoshida and Murata (2003) are:

(i) the number of individuals to which LS will be applied,

(ii) the maximal iteration number (depth) of the LS applied to an individual, and
(iii) the frequency of the same.

Next to these generic parameters there are some more specific ones related to the treatment of RPPs. Firstly, in order to guarantee balanced convergence to each RP, the improvements of each of them are sought cyclically.

The second, and probably most important, issue to tackle is which solution to chose for improvement and how to perform the LS. Starting with the latter, the LS is done here by RDS. Note that the RPPs (2) for $D = \| \cdot \|_2$ and (8) coincide if Z is utopian, but differences occur if Z is feasible. Since RDS is intended to support R-NSGA-II, the following procedure is performed for feasible reference points:

(i) approach a solution x with $F(x) = Z$ as fast as possible using RDS. The above discussion shows that the convergence is ideally quadratic in that case though only Jacobian information is used;
(ii) once feasibility is detected, perform the search in direction $d = (1, \ldots, 1)^T$ in order to find better (*i.e.* dominating) solutions.

Regarding the first question and considering the non-dominated sorting of R-NSGA-II, a local search algorithm could try to improve individuals on the first non-dominated front and backwards from then on. This approach, however, is not useful in particular when the number of objectives increases since most of the individuals will belong to this front. Hence, a second and more promising alternative is to improve solutions which are already close to RPs whether they are dominated or not. However, this selection scheme introduces a problem when an RP is found to be feasible during the evolution since, eventually, better (*i.e.* dominating) solutions will be found. Recall that the first goal of R-NSGA-II is to obtain non-dominated individuals and afterwards solutions close to the supplied RPs (which is in accord with RPP (2)).

In RDS-NSGA-II, individuals are sorted according to a fitness function that takes into account the feasibility of the RP. Then, the best individual is improved. The fitness functions for each case is as follows.

- If the RP is not feasible, the distance towards the RP becomes the fitness function. This way, closest solutions are preferred.
- If the RP is feasible, the distance towards the RP plus the non-dominated rank as used in R-NSGA-II times a constant becomes the fitness function. In this sense, dominating solutions are preferred and, within the same front, closest solutions are taken.

When constraints have to be considered in a memetic algorithm, the selection of individuals is, once again, crucial. Here, in order to comply with R-NSGA-II, the LS has to minimize the constraint violation value of the best individual. Nevertheless, solutions x that are already close to RPs with small constraint violation may yield less constraint violation than solutions y that dominate x depending on the landscape of the MOP. Motivated by this, the fitness function for the selection of individuals is reformulated as follows so as further to include the constraint violation of a solution.

- If the RP is not feasible, the distance towards the RP plus the constraint violation value becomes the fitness function.
- If the RP is feasible, the distance towards the RP plus the constraint violation value plus the non-dominated rank times a constant becomes the fitness function.

Finally, in order to avoid premature convergence by super elite individuals, the trail of solutions improved by the LS is stored by replacing the worst solutions in the population.

The whole process of selection and replacement of RDS-NSGA-II described so far is presented in Algorithm 3, wherein Sr denotes the success rate of the LS used for its application within R-NSGA-II. It is worth noticing that these considerations are completely independent of the choice of the LS algorithm. Finally, the pseudocode of the RDS as local searcher within R-NSGA-II is presented in Algorithm 4.

Algorithm 3 Local search engine of R-NSGA-II

Require: population P, RPs $Z_1, \ldots, Z_l$, feasible RPs so far FRP, iterations $XDSI$, solutions to improve per RP $XDSPZ$

```
Sr := 0                                                          ▷ Success rate of the RDS
for i = 1,...,XDSPZ do
    for each Z_i do
        if Z_i is feasible then
            Set the fitness function F_f(I) as the distance to Z_i plus a dominated penalty plus the constraint violation.
        else
            Set the fitness function F_f(I) as the distance to Z_i plus the constraint violation.
        end if
        Sort P using F_f(I).
        Take the best solution and apply the LS for XDSI iterations.
        Store the trail of solutions obtained in a set TS.
        if TS ≠ 0 then
            Replace the worst solutions of P by each solution in TS one by one.
            Sr := Sr + 1
        end if
        Check feasibility of Z_i.
    end for
end for
Set Sr := Sr/|Z| × XDSPZ.
```

5. Numerical results

Some numerical results for a comparison of RDS-NSGA-II against its base algorithm R-NSGA-II are presented in this section. Comparisons against the weighting achievement scalarizing function genetic algorithm (WASF-GA) (see Ruiz, Saborido and Luque [2015]), which considers a reference point and an achievement scalarizing function (ASF), are also included. For the following computations, the compared algorithms share some design parameters (see Table 1).

In order to compare the results of the algorithms, the inverted generational distance (IGD) (see Coello Coello and Cortés [2005]) had to be adapted to the problem at hand to make it work for utopian reference points. To be more precise, given the set Z of reference points and an archive A, the distance of Z toward $F(A)$ is measured as follows:

$$\mathrm{IGD}_Z(F(A), Z) := \frac{1}{|Z|} \sum_{i=1}^{|Z|} \min_{j=1,\ldots,|A|} dist(Z_i^*, F(a_j)), \tag{29}$$

where Z_i^* denotes the point on the Pareto front closest to Z_i, *i.e.* $\|Z_i - Z_i^*\| = dist(Z, F(P_Q))$, wherein $dist$ measures the distance between point and set and between two sets as $dist(u, A) = \inf_{v \in A} \|u - v\|$ and $dist(B, A) = \sup_{u \in B} dist(u, A)$, where u and v denote points and A and B sets. Accordingly, the

Algorithm 4 RDS within R-NSGA-II

Require: Initial point x_0, RP Z, flag of Z's feasibility, no. of iterations $XDSI$
Ensure: best found solution x_{XDSI}

$i := 0$
$x_H := \{x_0\}$ ▷ to store the trials
$F_H := \{F(x_0)\}$
while $i < XDSI$ **do**
  $d_G := Z - F(x_0)$
  if $\kappa(J(x_0)) > tol_\kappa$ **then** ▷ Phase II
    $d := P(x_i)d_G$
  else ▷ Phase I
    if Z is feasible **then** ▷ Looking for dominating solutions
      $d := (-1, \ldots, -1)^{\mathrm{T}}$
    else ▷ Looking for closer solutions
      Set $d := d_G$
    end if
  end if
  $d := d/\|d\|_2$
  Compute search direction ν using d
  Compute $p = x_i + t\nu$ s.t. $C_V(p) \leq C_V(x_i)$
  if (Z is not feasible **and** $D(Z, F(p)) < D(Z, F(x_i))$) **or** ($Z$ is feasible **and** $p \prec x$) **or** ($D(Z, F(p)) < D(Z, F(x_i)$ **and** $x_i \nprec p$) **then** ▷ accept p
    $x_{i+1} := p$
    $x_H := x_H \cup \{x_i\}$
    $F_H := F_H \cup \{F(x_i)\}$
  else
    $x_{i+1} := x_i$
  end if
  $i = i + 1$
end while
return x_{i+1}

optimal IGD_Z value is always zero. Note, however, that the evaluation of the IGD_Z value requires knowledge of the exact Pareto front. A performance indicator without this property might be interesting, and it is left for future research. The stopping criterion for the compared algorithms is determined by reaching a certain computational cost, given in terms of objective function evaluations; and, the same amount of this computational effort was used for each of them. The considered amount of function evaluations is specified, for each problem, in the corresponding comparison tables. Additionally, all the algorithms were independently executed until a maximal number of generations, specified in Table 1, in order to verify that all of them reached an acceptable solution, which indeed happened. The reference points used in all the experiments (including the real-world application examples) were chosen arbitrarily, just in order to analyse numerically the general ability of the algorithm to compute the complete sets of interest—defined by these reference points. The assumption is made that, in a particular application, the reference points are provided by the user. The presented results have a statistical significance based on the t-test using a 95% confidence interval. The values are retrieved from 30 independent runs.

The first experiment corresponds to the bi-objective problem CONV (Köppen and Yoshida, 2007):

$$f_1(x) = \|x - a_1\|_2^2, \quad f_2(x) = \|x - a_2\|_2^2, \tag{30}$$

Table 1. Values of the design parameters population size (P), maximum number of generations (G), crossover (C_P) and mutation probability (M_P), and ϵ spread (ϵ). Four additional design parameters are included for RDS: the initial call (I_C), frequency F (Initial,[min,max]), depth D of LS (Initial,[min,max]), and the number $XDSPZ$ of solutions to improve for each RP. Parameters are updated based on the success rate of the LS and stated in relation to a specific generation (G) or a percentage of the budget of generations (B). Meanwhile the parameter ϵ is replaced by N_μ (number of weights for each reference point) for the WASF-GA.

WASF-GA R-NSGA-II RDS-NSGA-II	P P P	G G G	C_P C_P C_P	M_P M_P M_P	 ϵ ϵ	 I_C	 F	 D	 $XDSPZ$	 N_μ
CONV	100	200	0.9	0.01	1E-3	1 (G)	0.15 (B),[0.1 (B), 11 (B)]	2 (G), [2 (G), 4 (G)]	1	10
ZDT1	100	200	0.9	0.03	1E-3	1 (G)	0.15 (B),[0.1 (B), 11 (B)]	2 (G), [2 (G), 4 (G)]	1	10
ZDT2	100	200	0.9	0.03	1E-3	1 (G)	0.15 (B),[0.1 (B), 11 (B)]	2 (G), [2 (G), 4 (G)]	1	10
ZDT3	100	200	0.9	0.03	1E-3	1 (G)	0.15 (B),[0.1 (B), 11 (B)]	2 (G), [2 (G), 4 (G)]	1	10
ZDT4	100	200	0.9	0.10	1E-3	1 (G)	0.15 (B),[0.1 (B), 11 (B)]	2 (G), [2 (G), 4 (G)]	1	10
DTLZ2 $k = 3$	100	200	0.9	0.03	1E-3	1 (G)	0.15 (B),[0.1 (B), 11 (B)]	2 (G), [2 (G), 4 (G)]	1	15
DTLZ2 $k = 5$	100	200	0.9	0.03	1E-3	1 (G)	0.15 (B),[0.1 (B), 11 (B)]	2 (G), [2 (G), 4 (G)]	1	25
DTLZ3	100	500	0.9	0.14	1E-3	1 (G)	0.15 (B),[0.1 (B), 11 (B)]	2 (G), [2 (G), 4 (G)]	1	15
C1-DTLZ1	100	250	0.9	0.14	1E-5	1 (G)	0.15 (B),[0.1 (B), 11 (B)]	2 (G), [2 (G), 4 (G)]	1	15
C2-DTLZ2	100	150	0.9	0.003	1E-5	1 (G)	0.15 (B),[0.1 (B), 11 (B)]	2 (G), [2 (G), 4 (G)]	1	15
CONVEX-C2-DTLZ2	100	150	0.9	0.083	1E-5	1 (G)	0.15 (B),[0.1 (B), 11 (B)]	2 (G), [2 (G), 4 (G)]	1	15

where $Q = \mathbb{R}^{100}$, $a_1 = (1, \ldots, 1)^T \in \mathbb{R}^{100}$ and $a_2 = -a_1$. The four RPs chosen were $Z = \{(20, 200)^T, (100, 50)^T, (250, 10)^T, (150, 150)^T\}$. Since CONV is a convex problem, large improvements are expected via the help of RDS. This is indeed the case as can be seen in Table 2, where the IGD$_Z$ values for all the examples are significantly better in all three stages. Next, the ZDT series of box-constrained, two-objective problems (Zitzler, Deb and Thiele, 2000) were considered. In the three first cases, $n = 30$ was used. Compared to CONV, RDS-NSGA-II needs more function evaluations on ZDT1 to obtain a covering of the optimal solutions near the RPs, with $Z = \{(0.1, 0.6)^T, (0.5, 0.2)^T, (0.9, 0)^T\}$, but comes quite close after only 6000 function calls (see Table 2). The t-test showed that there is indeed a difference in performance between these algorithms. Problem ZDT2 represents a challenge for evolutionary algorithms since typically, in the early stages of the search, all solutions concentrate on the top left area of the objective space where weak Pareto points exist. In this case $Z = \{(0.1, 0.9)^T, (0.8, 0.2)^T, (0.6, 0.5)^T\}$. Table 2 confirms that the performance of the memetic algorithm is significantly better in the best, median and worst cases for the checkpoints considered. ZDT3 has a more complex geometry than the previous ones, where not all boundary solutions are optimal. Here, $Z = \{(0.8, -0.6)^T, (0.1, 0.6)^T, (0.35, 0.1)^T\}$ has been used. In the early stages of the evolution, the memetic strategy outperforms R-NSGA-II (see Table 2). Nevertheless, with a budget of 5400 function calls, the solutions of both algorithms are very close together. ZDT4 is a multi-modal problem with a convex Pareto front where the challenge is to overcome 21 local Pareto fronts for $n = 10$. The search space is $Q = [0, 1] \times [5, 5]^9$. In this case, the RPs described by $Z = \{(0, 0.6)^T (0.35, 0.25)^T (0.6, 0.1)^T (0.85, 0)^T\}$ were used. The multi-modal nature of ZDT4 makes it more difficult to reach the Pareto front in comparison to the previous problems. However, it can be seen from Table 2 that there is indeed a better distribution and convergence of solutions of the memetic algorithm.

The second group of problems considered comes from the DTLZ series (Deb *et al.*, 2005). For the three- and five-objective DTLZ2, the setting was $n = 20$ and as RPs $Z = \{(0, 0, 0.9)^T, (0, 0.9, 0)^T, (0.9, 0, 0)^T\}$ and $Z = \{(0.25, 0.25, 0.25, 0.25, 0.25)^T\}$, respectively. It can be seen from Table 3 that a great improvement is reached already in the early stage, and the performance of RDS-NSGA-II is significantly better in all stages of the search. This problem indicates that RDS-NSGA-II keeps significantly outperforming the traditional MOEA, regardless of the number of objectives. Next, DTLZ3 is a highly multi-modal problem with $3^n - 1$ local Pareto fronts. The setting for this case was $n = 7$ and $Z = \{(0, 0, 0.9)^T, (0, 0.9, 0)^T, (0.9, 0, 0)^T\}$. It is possible to see from Table 3 that the performance of RDS-NSGA-II in this problem is better only in the early and the middle stage, although the t-test hints that the performance of both algorithms is similar. This behaviour was somehow expected as

Table 2. Best, median and worst IGD_Z values for RDS-NSGA-II, RNSGA-II and WASF-GA at three different stages of the search process. The values for Z^* in each case are (a) $\{(31.36, 207.36)^T, (125.44, 77.44)^T, (249.64, 176.40)^T, (100, 100)^T\}$, (b) $\{(0.1356, 0.6316)^T, (0.5427, 0.2633)^T, (0.9195, 0.0410)^T\}$, (c) $\{(0.2814, 0.9208)^T, (0.7839, 0.3854)^T, (0.9798, 0.0397)^T\}$, (d) $\{(0.8291, -0.5680)^T, (0.0854, 0.6699)^T, (0.4221, 0.0801)^T\}$, (e) $\{(0.1105, 0.6675)^T, (0.4271, 0.3464)^T, (0.6532, 0.1917)^T, (0.8844, 0.0595)^T\}$.

Problem	Stage	RDS-NSGA-II	RNSGA-II	WASF-GA
(a) CONV	Early (600)	**0.906,710** **1.101,572** (0.094,642) **1.254,400**	2.582,100 2.899,363 (0.121,790) 3.154,800	1.380,672 1.476,267 (0.246,6623) 1.577,117
	Middle (2000)	**0.003,985** **0.011,901** (0.004,613) **0.020,219**	1.002,100 1.146,750 (0.073,128) 1.300,100	1.283,984 1.360,019 (0.156,892) 1.455,836
	Later (5000)	**0.000,496** **0.002,284** (0.001,145) **0.004,600**	0.259,280 0.338,415 (0.039,805) 0.452,310	1.087,200 1.160,084 (0.121,992) 1.249,745
(b) ZDT1	Early (400)	**0.314,820** **0.505,073** (0.081,484) **0.633,980**	1.335,100 1.540,096 (0.116,520) 1.742,100	1.832,003 2.163,5441 (0.168,249) 2.424,708
	Middle (1800)	**0.067,492** **0.141,993** (0.056,467) **0.271,670**	0.429,280 0.490,541 (0.055,729) 0.636,430	1.473,402 1.824,819 (0.176,080) 2.117,937
	Later (6000)	**0.000,253** **0.001,451** (0.000,837) **0.003,725**	0.030,279 0.044,529 (0.009,092) 0.067,333	0.893,479 1.156,984 (0.136,877) 1.402,120
(c) ZDT2	Early (200)	**0.138,220** **0.282,305** (0.114,797) **0.581,880**	0.760,730 1.058,190 (0.195,260) 1.487,200	2.971,300 3.305,086 (0.206,542) 3.719,120
	Middle (3600)	**0.008,232** **0.028,267** (0.015,262) **0.065,586**	0.294,620 0.553,861 (0.226,002) 0.995,980	2.570,304 2.848,948 (0.205,532) 3.245,749
	Later (6200)	**0.000,145** **0.000,724** (0.000,371) **0.001,714**	0.056,378 0.199,139 (0.212,441) 0.683,100	1.895,402 2.276,140 (0.201,358) 2.682,438
(d) ZDT3	Early (800)	**0.322,600** **0.494,672** (0.112,499) **0.712,720**	0.561,860 0.655,991 (0.045,888) 0.772,640	1.262,182 1.758,837 (0.230,404) 2.152,277
	Middle (2600)	**0.054,966** **0.115,268** (0.035,003) **0.176,390**	0.118,280 0.163,294 (0.031,888) 0.217,180	1.065,078 1.514,358 (0.224,013) 1.895,566
	Later (5400)	**0.003,117** **0.027,388** (0.018,857) **0.078,693**	0.027,756 0.046,208 (0.015,386) 0.111,150	0.882,999 1.214,069 (0.189,828) 1.618,889
(e) ZDT4	Early (400)	**8.220,500** **27.265,883** (10.165,459) 56.936,000	33.492,000 48.271,767 (6.739,564) 61.591,000	20.491,350 35.702,965 (7.547,347) **50.136,089**
	Middle (2000)	**3.780,700** **8.507,060** (3.037,916) **15.567,000**	4.933,700 11.554,433 (3.592,855) 20.110,000	6.179,283 14.167,501 (5.115,166) 26.827,975
	Later (6000)	**0.237,630** **0.692,766** (0.371,938) **1.673,900**	0.571,870 1.509,929 (0.586,590) 3.092,100	2.280,254 7.458,826 (4.194,146) 20.810,458

the LS operator can get stuck in any of the many local Pareto fronts. Nevertheless, it is important to note that the memetic strategy does not perform worse.

To evaluate the performance on constrained MOPs, problem C1-DTLZ1 (Jain and Deb, 2014) is first analysed. This instance is based on DTLZ1, where the majority of the search space is made infeasible. DTLZ1 has $11^n - 1$ local Pareto fronts to overcome. The setting in this case is $n = 7$

Table 3. Best, median and worst IGD$_Z$ for RDS-NSGA-II, RNSGA-II and WASF-GA at three different stages of the search process. The values for Z^* in each case are (a) $\{(0.0013, 0.0003, 0.9999)^T, (0.0250, 0.9993, 0.0243)^T, (0.9915, 0.1117, 0.0666)^T\}$, (b) $\{(0.4390, 0.5116, 0.4961, 0.4634, 0.2906)^T\}$, (c) $\{(0.0000, 0.0008, 0.9999)^T, (0.0316, 0.9953, 0.0904)^T, (0.9950, 0.0843, 0.0517)^T\}$.

Problem	Stage	RDS-NSGA-II	RNSGA-II	WASF-GA
		0.009,449	0.480,290	0.880,316
	Early (400)	**0.021,091** (0.007,473)	0.583,997 (0.057,965)	1.057,196 (0.082,406)
		0.042,114	0.736,570	1.174,723
		0.001,590	0.051,691	0.663,798
(a) DTLZ2 ($k = 3$)	Middle (2000)	**0.004,786** (0.001,778)	0.075,625 (0.012,442)	0.831,029 (0.085,443)
		0.007,548	0.097,156	0.989,220
		0.000,282	0.006,714	0.408,818
	Later (5000)	**0.001,445** (0.000,733)	0.011,200 (0.002,145)	0.553,241 (0.062,776)
		0.003,115	0.014,696	0.661,738
		0.451,640	0.765,460	0.692,515
	Early (400)	**0.521,481** (0.058,174)	0.928,997 (0.083,361)	1.062,761 (0.126,726)
		0.634,230	1.089,100	1.257,843
		0.441,670	0.473,330	0.663,666
(b) DTLZ2 ($k = 5$)	Middle (2000)	**0.449,111** (0.007,101)	0.492,402 (0.013,223)	0.907,353 (0.116,759)
		0.469,450	0.524,710	1.130,416
		0.440,990	0.442,150	0.475,169
	Later (5000)	**0.441,105** (0.000,109)	0.443,092 (0.000,635)	0.691,885 (0.100,908)
		0.441,490	0.444,720	0.906,363
		4.062,700	**2.435,800**	12.657,996
	Early (2000)	**11.356,508** (5.231,146)	13.495,268 (5.021,763)	30.967,616 (11.010,124)
		20.234,000	27.018,000	53.594,380
		0.126,080	**0.117,000**	8.421,146
(c) DTLZ3	Middle (6000)	**2.123,615** (1.603,333)	2.130,263 (1.603,003)	18.512,025 (9.134,932)
		6.393,600	6.417,900	41.364,659
		0.001,074	0.007,968	7.722,101
	Later (10,000)	0.742,888 (0.963,273)	**0.492,401** (0.627,942)	17.170,733 (9.198,662)
		3.635,100	**2.768,300**	40.392,409

and $Z = \{(0.4, 0, 0.05)^T, (0.05, 0.35, 0.05)^T, (0.3, 0.1, 0.05)^T, (0.2, 0.2, 0.2)^T\}$. Table 4 shows that, for the memetic algorithm, a big improvement can be expected even when the population lies in the infeasible region (early stage); however, the LS operator gains the most when solutions lie close to the Pareto front (later stage). Table 4 confirms that the IGD$_Z$ values are better in these two stages for the memetic algorithm. C2-DTLZ2 (Jain and Deb, 2014) extends the original DTLZ2 problem by adding one inequality constraint. The feasible objective space lies inside $k{+}1$ hyper spheres of radius $r = 0.4$. For $n = 12$ and $Z = \{(0.4, 0.4, 0.4)^T, (0.6, 0.6, 0)^T, (0.8, 0, 0)^T, (0.15, 0.8, 0.15)^T, (0, 0, 0.9)^T\}$, Table 4 show that RDS-NSGA-II clearly outperforms R-NSGA-II in all three stages. Next, a convex reformulation of DTLZ2 (Jain and Deb, 2014) is taken with one nonlinear constraint and $Z = \{(0.9, 0.1, 0.1)^T, (0, 0, 0.8)^T, (0.2, 0.4, 0.1)^T, (0.35, 0.1, 0.1)^T, (0.4, 0.5, 0.15)^T\}$. Table 4 shows that such constraints can be handled easily by the memetic algorithm with a significant improvement in the early, middle or later stages. It is obvious that there is a clear advantage of the memetic algorithm over its base MOEA.

6. Conclusions and future work

A novel hybrid algorithm has been presented in this article for reference point problems (RPPs) in the treatment of multi-objective optimization problems (MOPs) as follows.

- The recently proposed directed search (DS) method was adapted to the context of a particular RPP. It has been shown that the flow induced by the search direction always leads to a critical point of the given RPP. For the numerical realization it has been shown that local quadratic

Table 4. Best, median and worst IGD_Z for RDS-NSGA-II, RNSGA-II and WASF-GA at three different stages of the search process. The values for Z^* in each case are (a) $\{(0.4093, 0.0270, 0.0635)^T, (0.0614, 0.3638, 0.0746)^T, (0.3147, 0.1206, 0.0645)^T, (0.1599, 0.1667, 0.1732)^T\}$, (b) $\{(0.5828, 0.5908, 0.5577)^T, (0.6582, 0.7527, 0.0068)^T, (0.9965, 0.0623, 0.0549)^T, (0.2084, 0.9540, 0.2154)^T, (0.0003, 0.0000, 0.9999)^T\}$, (c) $\{(0.8558, 0.0006, 0.0499)^T, (0.0114, 0.0099, 0.7935)^T, (0.1696, 0.3304, 0.0132)^T, (0.3545, 0.0843, 0.1142)^T, (0.1657, 0.3514, 0.0000)^T\}$.

Problem	Stage	RDS-NSGA-II	RNSGA-II	WASF-GA
(a) C1-DTLZ1	Early (750)	**4.471,100** **12.549,900** (6.227,609) **23.485,000**	13.465,000 27.557,950 (8.708,799) 43.109,000	14.220,454 37.075,567 (12.585,674) 69.596,294
	Middle (6000)	**1.018,600** **3.417,825** (1.597,864) **6.621,500**	0.746,190 3.980,884 (2.048,932) 8.420,100	6.104,418 15.280,234 (4.494,986) 22.117,270
	Later (9000)	**0.001,753** **0.051,518** (0.139,535) **0.637,570**	0.007,376 0.494,893 (0.468,109) 1.806,900	2.129,570 10.092,250 (4.539,223) 18.760,968
(b) C2-DTLZ2	Early (600)	**0.043,942** **0.114,567** (0.082,092) 0.433,460	0.139,890 0.256,823 (0.068,890) **0.385,260**	0.551,369 0.730,026 (0.079,847) 0.888,241
	Middle (2000)	**0.002,885** **0.013,328** (0.011,808) **0.041,206**	0.030,398 0.086,089 (0.048,306) 0.182,830	0.536,580 0.715,307 (0.076,737) 0.848,311
	Later (5000)	**0.000,944** **0.001,684** (0.000,561) **0.002,941**	0.003,215 0.013,915 (0.022,673) 0.104,700	0.500,838 0.701,967 (0.093,301) 0.924,922
(c) C2-CONVEX-DTLZ2	Early (600)	**0.023,379** **0.049,699** (0.024,897) **0.118,890**	0.115,840 0.205,667 (0.049,033) 0.308,500	0.617,886 1.485,955 (0.388,340) 2.167,592
	Middle (2000)	**0.002,878** **0.005,857** (0.002,477) 0.011,516	0.025,375 0.039,196 (0.010,318) **0.065,317**	0.485,541 1.366,203 (0.368,362) 1.915,107
	Later (5000)	**0.000,749** **0.001,445** (0.000,532) **0.003,154**	0.002,721 0.004,061 (0.000,942) 0.005,898	0.447,860 1.100,603 (0.316,929) 1.895,596

convergence can be expected of the numerical method in cases where the reference point is feasible. Further, the treatment of constrained problems was addressed, and given a novel way to explore the Pareto front near to a given solution. For the latter, the steering property of the DS has been exploited.

- A possible integration of the resulting local searcher, RDS, into R-NSGA-II, which is a state-of-the-art evolutionary algorithm for the efficient treatment of RPPs, has been presented.
- This work includes numerical results on 11 benchmark problems that exhibited different characteristics, such as different shapes of the domain and the Pareto set/front, modality and dimensions. Comparison of the new hybrid, RDS-NSGA-II, to its base algorithm R-NSGA-II showed superior performance in almost all cases. One exception, as anticipated, were highly multi-modal functions since for those functions there is a high chance for any local searcher to get stuck in local optimal solutions. However, it is important to note that in none of the cases was the performance of the hybrid inferior. The new algorithm can thus be regarded as advantageous, and the difference seems to become most important for the treatment of real-world applications where function evaluations tend to be very costly. It must also be highlighted that RDS-NSGA-II reaches the Pareto-optimal front with a fraction of the number of function evaluations compared to R-NSGA-II.

Though the results obtained in this work are very promising, there are some issues that can be addressed in future work such as the improvement/adaptation of the RDS. The constraint handling techniques may be advanced. This work considered box and equality constraints. The treatment of general inequality constraints has (so far) been left to the base algorithm, but it might be worth integrating directly into the RDS. Finally, it would be interesting to demonstrate the usefulness of RDS-NSGA-II on a computationally expensive real-world application.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

Alejandro Hernández and Oliver Cuate acknowledge support by the Consejo Nacional de Ciencia y Tecnología (CONACYT); Adriana Lara acknowledges project SIP20152082.

References

Allgower, Eugene L., and Kurt Georg. 1990. *Numerical Continuation Methods*. Vol. 13 of the *Springer Series in Computational Mathematics*. Berlin: Springer-Verlag. doi:10.1007/978-3-642-61257-2.

Branke, Jürgen, and Kalyanmoy Deb. 2005. "Integrating User Preferences into Evolutionary Multi-Objective Optimization." In *Knowledge Incorporation in Evolutionary Computation*, Vol. 167 of the series *Studies in Fuzziness and Soft Computing*, 461–477. Berlin: Springer. doi:10.1007/978-3-540-44511-1_21.

Branke, Jürgen, Kalyanmoy Deb, Henning Dierolf, and Matthias Osswald. 2004. "Finding Knees in Multi-Objective Optimization." In *Proceedings of the 8th International Conference on Parallel Problem Solving from Nature (PPSN VIII)*, Vol. 3242 of the series *Lecture Notes in Computer Science*, 722–731. Berlin: Springer. doi:10.1007/978-3-540-30217-9_73.

Coello, Coello, Carlos A., and Nareli Cortés. 2005. "Solving Multiobjective Optimization Problems Using an Artificial Immune System." *Genetic Programming and Evolvable Machines* 6 (2): 163–190.

Coello, Coello, Carlos A., Gary Lamont, and David Van Veldhuizen. 2007. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Genetic and Evolutionary Computation Series. New York: Springer Science & Business Media LLC. doi:10.1007/978-0-387-36797-2.

Das, Indraneel, and John Dennis. 1998. "Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems." *SIAM Journal on Optimization* 8 (3): 631–657.

Deb, Kalyanmoy. 2001. *Multi-Objective Optimization Using Evolutionary Algorithms*. Vol. 16 of *Wiley–Interscience Series in Systems and Optimization*. Chichester, UK: Wiley.

Deb, Kalyanmoy, Amrit Pratap, Sameer Agarwal, and T. A. M. T. Meyarivan. 2002. "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II." *IEEE Transactions on Evolutionary Computation* 6 (2): 182–197.

Deb, Kalyanmoy, and J. Sundar. 2006. "Reference Point Based Multi-Objective Optimization Using Evolutionary Algorithms." In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation (GECCO '06)*, 635–642. New York: ACM. doi:10.1145/1143997.1144112.

Deb, Kalyanmoy, Lothar Thiele, Marco Laumanns, Eckart Zitzler. 2005. "Scalable Test Problems for Evolutionary Multiobjective Optimization." In *Evolutionary Multiobjective Optimization—Theoretical Advances and Applications*, Part of the series *Advanced Information and Knowledge Processing*, 105–145. Springer. doi:10.1007/1-84628-137-7_6.

Dellnitz, Michael, Oliver Sühtze, and Thorsten Hestermeyer. 2005. "Covering Pareto Sets by Multilevel Subdivision Techniques." *Journal of Optimization Theory and Applications* 124 (1): 113–136.

Eichfelder, Gabriele. 2008. *Adaptive Scalarization Methods in Multiobjective Optimization*. Part of the series Vector Optimization. Berlin: Springer-Verlag. doi:10.1007/978-3-540-79159-1.

Fliege, Jörg. 2004. "Gap-Free Computation of Pareto-Points by Quadratic Scalarizations." *Mathematical Methods of Operations Research* 59 (1): 69–89.

Gil, Consolación, Antonio Márquez, Raul Baños, María Montoya, and Julio Gómez. 2007. "A Hybrid Method for Solving Multi-Objective Global Optimization Problems." *Journal of Global Optimization* 38 (2): 265–281.

Hernández, Carlos, Yousef Naranjani, Yousef Sardahi, Wei Liang, Oliver Schütze, and Jian-Qiao Sun. 2013. "Simple Cell Mapping Method for Multi-Objective Optimal Feedback Control Design." *International Journal of Dynamics and Control* 1 (3): 231–238.

Hernández, Carlos, Jian-Qiao Sun, and Oliver Schütze. 2013. "Computing the Set of Approximate Solutions of a Multi-Objective Optimization Problem by Means of Cell Mapping Techniques." In *EVOLVE—A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation IV* , Vol. 227 of the series *Advances in Intelligent Systems and Computing*, 171–188. Cham, Switzerland: Springer International Publishing. doi:10.1007/978-3-319-01128-8_12.

Hillermeier, Claus. 2001. *Nonlinear Multiobjective Optimization: A Generalized Homotopy Approach*. Vol. 135 of the *International Series of Numerical Mathematics*. Basel, Switzerland: Birkhšuser. doi:10.1007/978-3-0348-8280-4.

Ignizio, James P. 1976. *Goal Programming and Extensions*. Lanham, MD: Lexington Books.

Ishibuchi, Hisao, Tadashi Yoshida, and Tadahiko Murata. 2003. "Balance between Genetic Search and Local Search in Memetic Algorithms for Multiobjective Permutation Flowshop Scheduling." *IEEE Transactions on Evolutionary Computation* 7 (2): 204–223.

Jahn, Johannes. 2006. "Multiobjective Search Algorithm with Subdivision Technique." *Computational Optimization and Applications* 35 (2): 161–175.

Jain, Himanshu, and Kaushik Deb. 2014. "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point Based Nondominated Sorting Approach, Part II: Handling Constraints and Extending to an Adaptive Approach." *IEEE Transactions on Evolutionary Computation* 18 (4): 602–622.

Kim, Hyoungjin, and Meng-Sing Liou. 2014. "Adaptive Directional Local Search Strategy for Hybrid Evolutionary Multiobjective Optimization." *Applied Soft Computing* 19 (Jun.): 290–311.

Kipouros, Timoleon. 2013. "Stochastic Optimisation in Computational Engineering Design." In *EVOLVE—A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation II*, Vol. 175 of the series *Advances in Intelligent Systems and Computing*, 475–490. Berlin: Springer. doi:10.1007/978-3-642-31519-0_31.

Koppen, Mario, and Kaori Yoshida. 2007. "Substitute Distance Assignments in NSGA-II for Handling Many-Objective Optimization Problems." In *Evolutionary Multi-Criterion Optimization*, Vol. 4403 of the series *Lecture Notes in Computer Science*, 727–741. Berlin: Springer. doi:10.1007/978-3-540-70928-2_55.

Mattson, Christopher, Anoop Mullur, and Achille Messac. 2004. "Smart Pareto Filter: Obtaining a Minimal Representation of Multiobjective Design Space." *Engineering Optimization* 36 (6): 721–740.

Mejí, aJesús, Oliver Schütze, and Kalyanmoy Deb. 2014. "A Memetic Variant of R-NSGA-II for Reference Point Problems." In *EVOLVE—A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V* , Vol. 288 of the series *Advances in Intelligent Systems and Computing*, 247–260. Cham, Switzerland: Springer International Publishing. doi:10.1007/978-3-319-07494-8_17.

Miettinen, Kaisa. 1999. *Nonlinear Multiobjective Optimization*. Vol. 12 of the *International Series in Operations Research & Management Science*. New York: Springer Science & Business Media. doi:10.1007/978-1-4615-5563-6.

Mohammadi, Asad, Mohammad Nabi Omidvar, and Xiaodong Li. 2012. "Reference Point Based Multi-Objective Optimization through Decomposition." In *IEEE Congress on Evolutionary Computation (CEC 2012)*, 1–8. Piscataway, NJ: IEEE. doi:10.1109/CEC.2012.6256486.

Nocedal, Jorge, and Stephen Wright. 2006. *Numerical Optimization*. *Springer Series in Operations Research and Financial Engineering*. New York: Springer Science & Business Media LLC. doi:10.1007/978-0-387-40065-5.

Oliver, John, Timoleon Kipouros, and Mark Savill. 2013. "A Self-Adaptive Genetic Algorithm Applied to Multi-Objective Optimization of an Airfoil." In *EVOLVE—A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation IV* , Vol. 227 of the series *Advances in Intelligent Systems and Computing*, 261–276. Cham, Switzerland: Springer International Publishing. doi:10.1007/978-3-319-01128-8_17.

Rosen, Jo. 1960. "The Gradient Projection Method for Nonlinear Programming. Part I: Linear Constraints." *Journal of the Society for Industrial and Applied Mathematics* 8 (1): 181–217.

Ruiz, Ana, Rubén Saborido, and Mariano Luque. 2015. "A Preference-Based Evolutionary Algorithm for Multiobjective Optimization: The Weighting Achievement Scalarizing Function Genetic Algorithm." *Journal of Global Optimization* 62 (1): 101–129.

Schütze, Oliver, Adanay Martín, Adriana Lara, Sergio Alvarado, Eduardo Salinas, and Carlos A Coello Coello. 2016. "The Directed Search Method for Multi-Objective Memetic Algorithms." *Computational Optimization and Applications* 63 (2): 305–332.

Trautmann, Heike, Tobias Wagner, Dirk Biermann, and Claus Weihs. 2013. "Indicator-Based Selection in Evolutionary Multiobjective Optimization Algorithms Based on the Desirability Index." *Journal of Multi-Criteria Decision Analysis* 20 (56): 319–337.

Wagner, Tobias, and Heike Trautmann. 2010. "Integration of Preferences in Hypervolume-Based Multiobjective Evolutionary Algorithms by Means of Desirability Functions." *IEEE Transactions on Evolutionary Computation* 14 (5): 688–701.

Wierzbicki, Andrzej. 1980. "The Use of Reference Objectives in Multiobjective Optimization." In *Multiple Criteria Decision Making Theory and Application*, Vol. 177 of the series *Lecture Notes in Economics and Mathematical Systems*, 468–486. Berlin: Springer-Verlag. doi:10.1007/978-3-642-48782-8_32.

Zhang, Xingyi, Ye Tian, and Yaochu Jin. 2015. "A Knee Point-Driven Evolutionary Algorithm for Many-Objective Optimization." *IEEE Transactions on Evolutionary Computation* 19 (6): 761–776.

Zitzler, Eckart, Kalyanmoy Deb, and Lothar Thiele. 2000. "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results." *Evolutionary Computation* 8 (2): 173–195.