

A preference-based multi-objective evolutionary algorithm R-NSGA-II with stochastic local search

Ernestas Filatovas¹ · Algirdas Lančinskas¹ ·
Olga Kurasova¹ · Julius Žilinskas¹

Published online: 13 May 2016
© Springer-Verlag Berlin Heidelberg 2016

Abstract Incorporation of a decision maker's preferences into multi-objective evolutionary algorithms has become a relevant trend during the last decade, and several preference-based evolutionary algorithms have been proposed in the literature. Our research is focused on improvement of a well-known preference-based evolutionary algorithm R-NSGA-II by incorporating a local search strategy based on a single agent stochastic approach. The proposed memetic algorithm has been experimentally evaluated by solving a set of well-known multi-objective optimization benchmark problems. It has been experimentally shown that incorporation of the local search strategy has a positive impact to the quality of the algorithm in the sense of the precision and distribution evenness of approximation.

Keywords Multi-objective optimization · Preference-based evolutionary algorithms · Memetic algorithm · Stochastic local search

1 Introduction

Many real-world problems are multi-objective. Due to conflicting objectives usually there is no solution which would be the best by all objectives. However, a set of solutions optimal in a multi-objective sense exists. Such a set of solutions is called the Pareto set, and the corresponding set of objective vectors—the Pareto front. Determination of the Pareto front is the main goal of multi-objective optimization, however for some problems it is impossible to identify the exact Pareto front due to reasons

✉ Ernestas Filatovas
ernest.filatov@gmail.com

¹ Institute of Mathematics and Informatics, Vilnius University,
Akademijos str. 4, 08663 Vilnius, Lithuania

such as continuity of the front, nonexistence of analytical expression or complexity of the problem being solved. On the other hand in practical situations usually it is not necessary to find the whole Pareto front, but rather its approximation. Moreover, the decision maker (DM) commonly is interested in a certain part of the Pareto front and prefers to analyse that part deeper. Therefore, algorithms for approximation of the Pareto front are useful to tackle real-world multi-objective optimization problems.

A well-known class of such algorithms are Evolutionary Multi-objective Optimization (EMO) algorithms. The main target of a classic EMO is to find a set of well-distributed points on objective space that precisely approximate the entire Pareto front (Deb 2001; Lančinskas et al. 2013; Talbi 2009). The obtained set is presented to the DM for consideration and selection of the most preferable solutions. Some well-known EMO algorithms should be mentioned regarding to such a kind of approaches: PAES (Knowles and Corne 2000), SPEA2 (Zitzler et al. 2001), NSGA-II (Deb et al. 2002), IBEAZ (Zitzler and Künzli 2004), etc. The EMO algorithms have gained their popularity because they do not require deep knowledge about the problem to solve and are easy to implement. However, most of the EMO approaches are not suitable to problems with a big number of objectives (Knowles and Corne 2000). Recently several EMO algorithms have been developed that are suitable for solving many objective problems: MOEA/D (Ray et al. 2013; Zhang and Li 2007) and NSGA-III (Jain and Deb 2014). However, the problem of large computational cost arises here as big amount of solutions must be found to represent the entire Pareto front. Moreover, a reasonable number of solutions should be provided for the DM so that he/she could make an adequate decision avoiding complex analysis of large amount of information. Therefore, optimization methods based on DM's preferences must be used in which only the region of interest of the Pareto front is approximated.

During the last decade incorporation of DM's preference information into evolutionary approaches has become a new trend. So-called preference-based EMO algorithms are being actively developed that focus on only some parts of the Pareto front. Such algorithms aim to find an approximation of the Pareto front which elements are scattered regarding to the preference information provided by the DM. The well-known EMO algorithms of this kind have been proposed: the reference point-based (R-NSGA-II) (Deb et al. 2006; Siegmund et al. 2012), the light beam approach based EMO (Deb and Kumar 2007), the weighted hypervolume based EMO (Auger et al. 2009), achievement scalarizing function based EMOs—PIBEA (Thiele et al. 2009) and interactive EMO-based technique (López-Jaimes and Coello Coello 2014), interactive MOEA/D (Gong et al. 2011), Preference-based Interactive Evolutionary (PIE) algorithm (Sindhya et al. 2011), the Weighting Achievement Scalarizing Function based EMO (WASF-GA) (Ruiz et al. 2015). The comprehensive survey is given in Purshouse et al. (2014).

The preference-based EMO algorithms have been applied for solving different real-world optimization problems: R-NSGA-II was used for solving industrial-scale simulation-based optimization problem (Siegmund et al. 2012) and for deregulated power system in transmission expansion planning problem (Mohammadpour et al. 2013); PIBEA and PIE were applied for locating a pollution monitoring station in a two-dimensional decision space (Sindhya et al. 2011; Thiele et al. 2009); interactive MOEA/D was applied for airfoil design optimization problem (Gong et al. 2011), etc.

As preference-based algorithms are applied to different practical multi-objective problems, improvement of efficiency of such algorithms is undisputed. The one of the most common ways to enhance the EMO algorithm is to incorporate a local search procedure. Such hybrid algorithms, also called *memetic*, have gained a big attention as they converge faster and more precisely approximate the Pareto front (Zavala et al. 2014). Different strategies of local search have been applied for improvement of EMO algorithms: the fitness function based on weighted sum of multiple objectives was utilized when selecting solution for generating a new solution by crossover and mutation operations (Ishibuchi and Murata 1998); a similar idea was investigated in Goel and Deb (2002); local search procedure based on neighbourhood search was incorporated into the cellular multi-objective genetic algorithm (Murata et al. 2002); a memetic algorithm based on the cross dominant adaptation was proposed in Caponio and Neri (2009); a hybrid with achievement scalarizing function that can be solved with any appropriate local search method was presented in Sindhya et al. (2009); a gradient based sequential quadratic programming method was used in a local search for optimization of a scalarized objective function (Sindhya et al. 2013); Multi-Objective Single Agent Stochastic Search (MOSASS) (Lančinskas et al. 2011) was incorporated into NSGA-II (Lančinskas et al. 2013). More works related to memetic multi-objective optimization algorithms can be found in EMO survey (Zhou et al. 2011). Hence, hybridization of classic EMO algorithms with local search procedures was sufficiently deeply investigated, however to our knowledge, the preference-based EMO has not received great attention yet.

A memetic algorithm that integrates the Directed Search method into the most widely used preference-based EMO algorithm—R-NSGA-II has been proposed in Mejía et al. (2014). However, the algorithm involves Jacobian calculation that leads to specific requirements for a problem being solved or/and to additional computations for numerical estimation of gradients. In this paper, we propose to enhance R-NSGA-II by incorporating the heuristic local search technique MOSASS (Lančinskas et al. 2013, 2011) which does not require any gradient information. The proposed R-NSGA-II with MOSASS improves the quality of approximation of the Pareto front in a limited number of function evaluations, taking into account DM's preference information.

The rest of this paper is organized as follows. In Sect. 2 the description of the multi-objective problem is provided, as well as the preference-based EMO algorithm R-NSGA-II and the local search strategy MOSASS are described. Section 3 is devoted to proposition of a memetic algorithm derived by incorporating MOSASS into R-NSGA-II, and to adaptation of the performance metrics for evaluating the preference-based algorithms. The experimental investigation of the proposed algorithm and the obtained results are discussed in Sect. 4. Finally, conclusions are drawn in Sect. 5.

2 Background concepts

2.1 Multi-objective optimization problem

Here a formulation of a multi-objective optimization problem is introduced. Let us have $k \geq 2$ conflicting objectives, described by the functions $f_1(\mathbf{x})$, $f_2(\mathbf{x})$, \dots , $f_k(\mathbf{x})$,

where $\mathbf{x} = (x_1, x_2, \dots, x_n)$ is a vector of variables (*decision vector*), n is the number of variables. A multi-objective minimization problem is formulated as follows (Miettinen 1999):

$$\text{minimize } \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})], \quad (1)$$

$$\text{subject to } \mathbf{x} \in \mathbf{S}. \quad (2)$$

where $\mathbf{z} = \mathbf{f}(\mathbf{x}) \in \mathbb{R}^k$ is *objective vector*, and $\mathbf{S} \subset \mathbb{R}^n$ is an n -dimensional Euclidean space, called *feasible region*, which defines all feasible decision vectors.

In terms of multi-objective optimization, two decision vectors \mathbf{x} and \mathbf{x}' can be related with each other by dominance relation: the decision vector \mathbf{x} can dominate the decision vector \mathbf{x}' and vice versa, as well as none of the decision vectors dominate each other.

It is said that the decision vector \mathbf{x} *dominates* the decision vector \mathbf{x}' (denoted by $\mathbf{x} \succ \mathbf{x}'$) if

- (1) the decision vector \mathbf{x} is not worse than \mathbf{x}' by all objectives, and
- (2) the decision vector \mathbf{x} is strictly better than \mathbf{x}' by at least one objective.

If the conditions above are satisfied, then the decision vector \mathbf{x}' is called a dominator of \mathbf{x} . If none of the decision vectors can be identified as a dominator of the other, then both decision vectors are assumed to be *indifferent* (denoted by $\mathbf{x} \sim \mathbf{x}'$). Analogically, it is said that the objective vector $\mathbf{z} = (z_1, z_2, \dots, z_k)$ dominates $\mathbf{z}' = (z'_1, z'_2, \dots, z'_k)$, if each component of \mathbf{z} is not worse than the corresponding component of \mathbf{z}' and there exists at least one component of \mathbf{z} which is strictly better than the corresponding component of \mathbf{z}' .

Decision vectors which have no dominators in the whole search space \mathbf{S} are called *non-dominated* or *Pareto-optimal* (optimal in Pareto sense). The set of all non-dominated decision vectors is called the *Pareto set*, and the corresponding set of non-dominated objective vectors is called the *Pareto front*.

Furthermore, two more objective vectors that describe the ranges of the Pareto front are usually used—the *ideal objective vector* and the *nadir objective vector*. Lower bounds of the Pareto front constitute the ideal objective vector, its components are obtained by minimizing each of the objective functions individually in the feasible region. The upper bounds of the Pareto front are the components of the nadir objective vector.

2.2 Preference-based EMO algorithms

As it was mentioned in Sect. 1 the classic EMO algorithms aim to approximate the whole Pareto front and to distribute the obtained non-dominated objective vectors evenly. On contrary, the objective vectors obtained by preference-based algorithms are concentrated on a particular region(s) of the Pareto front approximated according to the preference information expressed by the DM. This information can be expressed in different ways: using reference direction, reference points, aspiration levels, weights or other (Miettinen 1999). Wide range of the preference-based EMO algorithms use

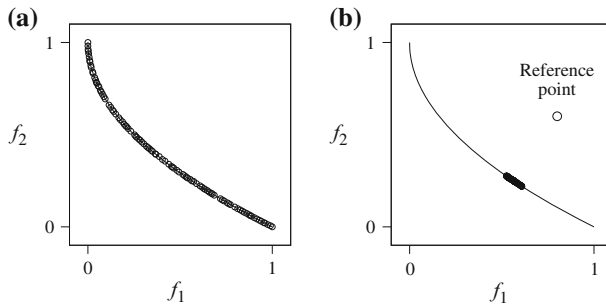


Fig. 1 Non-dominated objective vectors of ZDT1 problem, obtained by: **a** classic EMO algorithm, **b** preference-based EMO algorithm

reference points that consist of desirable aspiration levels for each objective function. Such a way of expressing preference is easy to implement and incorporate into EMO approach. Moreover, other types of information (as aspiration levels or reference directions) can be easily transformed into reference points. The differences between results obtained by the classic EMO and preference-based EMO algorithms are demonstrated using the popular test problem ZDT1 in Fig. 1. We see that the points of the Pareto front approximation are distributed evenly enough on the whole Pareto front in Fig. 1a, while the points are concentrated on one region in Fig. 1b.

Among the preference-based EMO algorithms R-NSGA-II (Deb et al. 2006; Siegmund et al. 2012) is the most commonly used one. It is an extension of the well-known classic EMO approach NSGA-II (Deb et al. 2002), where the crowding distance is changed to the Euclidean distance from the reference point which expresses the DM's preference. The general idea of R-NSGA-II is that the parents with smaller Euclidean distances to the reference point are preferred. R-NSGA-II can be described as follows:

1. A random initial population consisting of decision vectors is randomly generated in the feasible region.
2. A new offspring population is created by applying genetic operators (crossover and mutation) to the individuals of the parent population. The crossover is related to the recombination of two selected parent individuals by taking some values of elements from the first parent, while the remaining values are taken from the second one. The selected individuals are recombined with a predefined probability; if recombination is not performed, then better of the parents is used to represent an offspring. The derived offspring is mutated by adding a small random change to the values of some variables. Each variable is mutated with a predefined (usually small) probability; all variables usually have equal probabilities to be mutated.
3. The parent and offspring populations are combined into one joint population.
4. The new population is sorted into different non-domination levels (so-called fronts) by a non-dominated sorting procedure.
5. Each front of the joint population is clustered, and representatives of clusters are identified.
6. The obtained joint population is reduced to the size of the parent population by leaving the representatives of the clusters starting from the best non-domination

level. If not all representatives can be selected to the next generation, then ones with the smaller Euclidean distance to the reference point are preserved.

7. If the termination condition is not satisfied, then the reduced population is used as a parent population in the next generation (go to Step 2).

It should be noted that the clustering performed in Step 5 is required in order to control the diversity of the obtained solutions. The clustering parameter value δ ($0 < \delta \leq 1$) (Deb et al. 2006) defines the size of clusters and in such a way controls the extent of the population.

2.3 MOSASS

The local search strategy Multi-Objective Single Agent Stochastic Search (MOSASS) (Lančinskas et al. 2011, 2013) has been developed by modifying a random search strategy for single-objective optimization, called Single Agent Stochastic Search (SASS) (Solis and Wets 1981), in order to make it suitable for multi-objective optimization.

The MOSASS iterative algorithm begins with an initial decision vector $\mathbf{x} = (x_1, x_2, \dots, x_n)$, which is assumed to be locally optimized, and an empty archive A for storing non-dominated decision vectors which are expected to be found during runtime of the algorithm. A new decision vector \mathbf{x}' is generated by changing values of some variables of \mathbf{x} . Each variable x_i ($i = 1, 2, \dots, n$) is modified with probability $1/n$ by adding a random value ξ_i generated following Gaussian distribution. In general, the new decision vector can be expressed mathematically as

$$\mathbf{x}' \leftarrow \mathbf{x} + \xi, \quad (3)$$

where $\xi = (\xi_1, \xi_2, \dots, \xi_n)$ and

$$\xi_i \leftarrow \begin{cases} \mathcal{N}(b_i, \sigma), & \text{if } r_i \leq 1/n, \\ 0, & \text{if } r_i > 1/n. \end{cases} \quad (4)$$

Here $\mathcal{N}(b_i, \sigma)$ is a random number generated following Gaussian distribution with the bias b_i and the standard deviation σ , r_i is a random number uniformly generated within interval $[0, 1]$. Such a probabilistic approach for generation of a new decision vector leads to the change of a single variable in average; see Lančinskas et al. (2013) for details.

The generated decision vector \mathbf{x}' is compared with its precursor \mathbf{x} in the sense of dominance relation. If the new decision vector dominates its precursor, then \mathbf{x} is changed to the newly generated one. If \mathbf{x}' is indifferent with \mathbf{x} , then \mathbf{x}' is compared with all non-dominated decision vectors found so far and stored in the archive A . If \mathbf{x}' is non-dominated in A , then the archive is supplemented by \mathbf{x}' . If \mathbf{x}' is used either to change \mathbf{x} or to supplement the archive, all decision vectors in A dominated by \mathbf{x}' are removed in order to keep the archive of non-dominated decision vectors only.

If the newly generated decision vector is either dominated by its precursor or a decision vector from A , then \mathbf{x}' is rejected and an opposite decision vector

$$\mathbf{x}'' \leftarrow \mathbf{x} - \xi \quad (5)$$

is investigated following the same strategy.

If the decision vector \mathbf{x} is updated or the archive A is supplemented by \mathbf{x}' , then the iteration is assumed to be successful and the biases of Gaussian perturbation are updated:

$$b_i \leftarrow 0.2b_i + 0.4\xi_i, \quad i = 1, 2, \dots, n. \quad (6)$$

If the decision vector \mathbf{x} is updated or the archive A is supplemented by \mathbf{x}'' , then iteration is assumed to be successful and the biases of Gaussian perturbation are updated by

$$b_i \leftarrow b_i - 0.4\xi_i, \quad i = 1, 2, \dots, n. \quad (7)$$

If both \mathbf{x}' and \mathbf{x}'' are rejected, then the iteration is assumed to be failed and the biases of Gaussian perturbation are updated by

$$b_i \leftarrow 0.5b_i, \quad i = 1, 2, \dots, n. \quad (8)$$

The standard deviation σ of the Gaussian perturbation is dynamically adjusted with respect to the repetitive successful and failed iterations. If the number *scnt* of repetitive successful iterations reaches the predefined number *Scnt*, then the standard deviation is increased twice. Analogically, if the number *fcnt* of repetitive failed iterations reaches the predefined number *Fcnt*, then the standard deviation is reduced by a half. If the standard deviation becomes smaller than the predefined value σ_{min} , then σ is set to σ_{min} .

The iterative process is continued while a stopping criterion is not satisfied, which is usually based on the maximum number of functions evaluations. The algorithm returns a set $A \cup \{\mathbf{x}\}$ of non-dominated decision vectors found during the runtime of the algorithm. The detailed MOSASS algorithm is given in Algorithm 1.

3 Memetic preference-based EMO algorithm

The original R-NSGA-II algorithm aims to provide the points concentrated on the Pareto front region considering the DM's preferences. When solving real-world multi-objective optimization problems, the results should be provided to DM for his/her analysis in appropriate amount of time, therefore the number of function evaluations must be limited. In such a case, the widely-used R-NSGA-II algorithm is faced with difficulties when obtaining in the well-distributed points that approximate a part of the Pareto front.

The local search strategy MOSASS uses a single decision vector \mathbf{x} to generate a new decision vector, therefore it is likely that newly generated non-dominated decision vector will be in the neighborhood of \mathbf{x} . Examples of the Pareto front approximations obtained by 400 function evaluations of MOSASS are presented in Fig. 2. One can

Algorithm 1 MOSASS

```

1: procedure MOSASS( $\mathbf{x}, E_L$ )
2:   Set the boundary values for the standard deviation:  $\sigma_{min} = 1 \times 10^{-5}, \sigma_{max} = 0.5$ ;
3:   Set the initial value of the standard deviation:  $\sigma \leftarrow 0.5 \times (\sigma_{min} + \sigma_{max})$ ;
4:    $b_i \leftarrow 0, i = 1, 2, \dots, n$ ;
5:    $scnt \leftarrow 0; fcnt \leftarrow 0$ ;
6:    $Scnt = 5; Fcnt = 3$ ;
7:    $e_L \leftarrow 0$ ;
8:   while  $e_L < E_L$  do
9:     Generate a new decision vector  $\mathbf{x}'$  following equation (3);
10:    Evaluate objective vector of  $\mathbf{x}'$  and increase by one the value  $e_L$ ;
11:    if  $\mathbf{x}' > \mathbf{x}$  then
12:       $\mathbf{x} \leftarrow \mathbf{x}'$ ;
13:       $scnt \leftarrow scnt + 1; fcnt \leftarrow 0$ ;
14:       $b_i \leftarrow 0.2b_i + 0.4\xi_i, i = 1, 2, \dots, n$ ;
15:    else
16:      if  $\mathbf{x}' \sim \mathbf{x}$  and  $\mathbf{x}'$  is non-dominated in  $A$  then
17:        Remove from  $A$  all  $\mathbf{x}$  such that  $\mathbf{x}' > \mathbf{x}$ ;
18:        Supplement  $A$  by  $\mathbf{x}'$ ;
19:         $scnt \leftarrow scnt + 1; fcnt \leftarrow 0$ ;
20:         $b_i \leftarrow 0.2b_i + 0.4\xi_i, i = 1, 2, \dots, n$ ;
21:      else
22:        Generate a new decision vector  $\mathbf{x}''$  following equation (5);
23:        Evaluate objective vector of  $\mathbf{x}''$  and increase by one the value  $e_L$ ;
24:        if  $\mathbf{x}'' > \mathbf{x}$  then
25:           $\mathbf{x} \leftarrow \mathbf{x}''$ ;
26:           $scnt \leftarrow scnt + 1; fcnt \leftarrow 0$ ;
27:           $b_i \leftarrow b_i - 0.4\xi_i, i = 1, 2, \dots, n$ ;
28:        else
29:          if  $\mathbf{x}'' \sim \mathbf{x}$  and  $\mathbf{x}''$  is non-dominated in  $A$  then
30:            Remove from  $A$  all  $\mathbf{x}$  such that  $\mathbf{x}'' > \mathbf{x}$ ;
31:            Supplement  $A$  by  $\mathbf{x}''$ ;
32:             $scnt \leftarrow scnt + 1; fcnt \leftarrow 0$ ;
33:             $b_i \leftarrow b_i - 0.4\xi_i, i = 1, 2, \dots, n$ ;
34:          else
35:             $fcnt \leftarrow fcnt + 1; scnt \leftarrow 0$ ;
36:             $b_i \leftarrow 0.5b_i, i = 1, 2, \dots, n$ ;
37:          end if
38:        end if
39:      end if
40:    end if
41:     $\sigma \leftarrow \begin{cases} 2\sigma, & \text{if } scnt \geq Scnt, \\ 0.5\sigma, & \text{if } fcnt \geq Fcnt, \\ \sigma, & \text{otherwise.} \end{cases}$ 
42:  end while
43:  return set of non-dominated decision vectors  $A \cup \{\mathbf{x}\}$ .
44: end procedure

```

see, non-dominated decision vectors determined by the algorithm are concentrated around the initial decision vectors \mathbf{x} , denoted by the squares.

Such a generation of non-dominated vectors in the neighbourhood of an initial decision vector meets the concept of preference-based approaches. Moreover, the latter process is much quicker comparing to the genetic operators used in EMO. Therefore, it is reasonable to incorporate the local search strategy MOSASS into the preference-

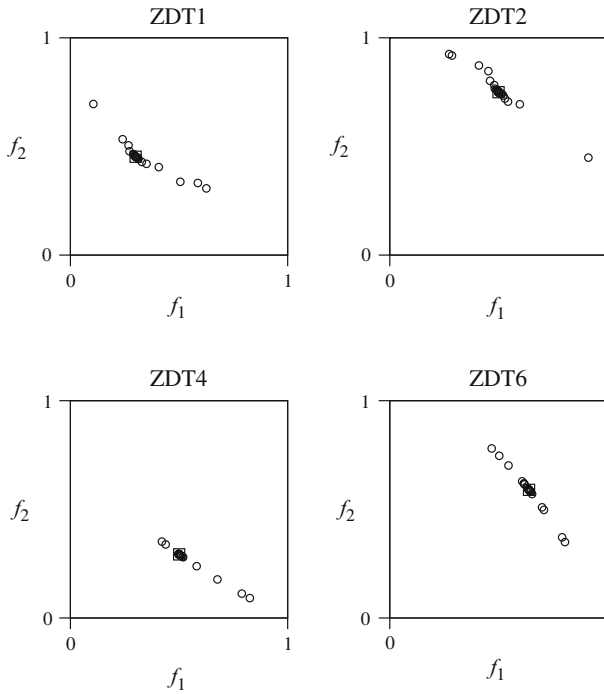


Fig. 2 Approximations of the Pareto fronts of different test problems obtained by MOSASS

based R-NSGA-II algorithm to improve the quality of approximation of the Pareto front in the limited number of function evaluations. In next subsection, we will describe the proposed memetic algorithm which we call R-NSGA-II with MOSASS.

3.1 The proposed R-NSGA-II with MOSASS algorithm

The memetic R-NSGA-II with MOSASS algorithm begins with initial parent population P_0 consisting of N decision vectors (see Algorithm 2). A value of clustering parameter δ ($0 < \delta \leq 1$) and the number of function evaluations E_L dedicated for the local search are also predefined. The algorithm runs while the termination criterion is not satisfied (for example the number of generations or the number of function evaluations is not reached). After sorting the population into different non-domination levels (fronts) and calculating rank for each individual an offspring population is created using local search procedure MOSASS or using standard genetic operators (recombination and mutation). MOSASS procedure is activated if condition for the local search is satisfied, i.e. after every predefined number of function evaluations E_G (see Row 6 of Algorithm 2). Individual \mathbf{x} with the best rank of the population P_t is passed to MOSASS procedure (see Algorithm 1) and local search runs till the number of function evaluations reaches the value E_L . The result of MOSASS procedure is the offspring population Q_t .

Algorithm 2 R-NSGA-II with MOSASS

```

1: procedure R-NSGA-II-MOSASS( $P_0, N, \delta, E_L$ )
2:   Generation counter  $t \leftarrow 0$ ;
3:   while the termination criterion is not satisfied do
4:     Sort the parent population  $P_t$  into different non-domination levels (fronts);
5:     Calculate rank for each individual;
6:     if condition for the local search activation is satisfied then
7:       Set  $\mathbf{x}$  to the individual from  $P_t$  with the best rank;
8:       Perform the local search:  $Q_t \leftarrow \text{MOSASS}(\mathbf{x}, E_L)$ ;
9:     else
10:      Create an offspring population  $Q_t$  using recombination and mutation;
11:    end if
12:    Combine the parents and offspring into the joint population:  $R_t = P_t \cup Q_t$ ;
13:    Sort  $R_t$  into fronts  $\bar{F}_i, i = 1, 2, \dots, p$  by non-dominated sorting;
14:     $P_{t+1} \leftarrow \emptyset$ ;
15:    while  $|P_{t+1}| < N$  do
16:      Cluster each front  $\bar{F}_i, i = 1, \dots, p$ , and identify representatives  $\bar{R}_{ti}$  of clusters;
17:       $i \leftarrow 1$ ;
18:      while  $i < p$  and  $|P_{t+1}| + |\bar{R}_{ti}| \leq N$  do
19:        Supplement  $P_{t+1}$  by representatives  $\bar{R}_{ti}$ ;
20:        Remove representatives  $\bar{R}_{ti}$  from the front  $\bar{F}_i$ ;
21:         $i \leftarrow i + 1$ ;
22:      end while
23:      if  $i < p$  then
24:        Supplement  $P_{t+1}$  by  $(N - |P_{t+1}|)$  representatives of the front  $\bar{F}_i$ 
25:        with the smallest rank;
26:      end if
27:    end while
28:     $t \leftarrow t + 1$ ;
29:  end while
30:  return  $P_t$ .
31: end procedure

```

After the joint population is created and sorted, a new population is started to be formed (see Row 14 of Algorithm 2). Each front \bar{F}_i of the joint population is clustered in such a way: the individuals which difference of the Euclidean distance to the reference point is less than δ are assigned to one cluster, where \bar{R}_t ($\bar{R}_t \subset R_t$) is a set of representatives of each cluster, \bar{R}_{ti} ($\bar{R}_{ti} \in \bar{R}_t$) are representatives in front \bar{F}_i . The representatives of each cluster are added to the new population (see Row 19). If all representatives in the front cannot be added to the population, only representatives with the smallest rank are selected (see Rows 24–25). In the case when there are not enough representatives from all the fronts to form the population of size N , clustering is performed again after removing the individuals already added to a new population. The result of the algorithm is the population P_t , which consists of decision vectors, and their corresponding objective vectors approximate the Pareto front.

3.2 Performance metrics for preference-based EMO

The EMO algorithms are usually measured on two aspects: the accuracy (how close the obtained approximation is to the Pareto front), and the evenness of distribution of the points in approximation.

The following performance metrics are commonly used to evaluate the performance of the EMO algorithms:

- *Generational Distance* measures the accuracy of the approximation of the Pareto front. The generational distance for an obtained set of points \tilde{P} in objective space is calculated by the formula (Deb 2001):

$$GD = \frac{1}{|\tilde{P}|} \sum_{j=1}^{|\tilde{P}|} \min_{l=1, \dots, |P^*|} \left(\sum_{i=1}^k \|\tilde{P}_{ji} - P_{li}^*\| \right), \tag{9}$$

Here we sum the Euclidean distances between the j th objective vector of \tilde{P} and the nearest objective vector of the Pareto front P^* . It is aimed to find approximating points as close as possible to the Pareto front, therefore lower value of Generational Distance means better quality of the approximation.

- *Spread* metric measures the distribution of the points in approximation. For a set \tilde{P} obtained by a classic EMO algorithm Spread is calculated by the formula (Deb 2001):

$$SP = \frac{\sum_{i=1}^k d_i^e + \sum_{j=1}^{|\tilde{P}|} |d_j - \hat{d}|}{\sum_{i=1}^k d_i^e + |\tilde{P}| \hat{d}}, \tag{10}$$

where

$$d_j = \min_{l=1, \dots, |\tilde{P}|, l \neq j} \|\tilde{P}_l - \tilde{P}_j\|, \tag{11}$$

$$\hat{d} = \frac{1}{|\tilde{P}|} \sum_{j=1}^{|\tilde{P}|} d_j, \tag{12}$$

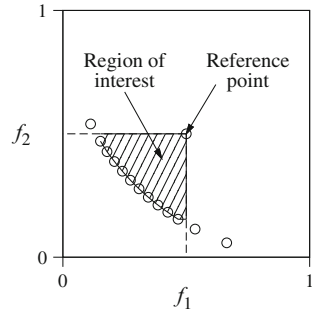
$$d_i^e = \|\tilde{P}_i^e - P_i^{*e}\|. \tag{13}$$

Thus, d_j is Euclidean distance between neighbouring objective vectors, \hat{d} is an average of these distances, d_i^e is a distance between the extreme objective vectors \tilde{P}_i^e and P_i^{*e} . In the case when the points of the approximated Pareto front are well-distributed, the Spread value is close to 0.

- *Hypervolume (HV)* metric measures the volume of the dominated region made by the obtained objective vectors approximating Pareto front and the given point R (Zitzler and Thiele 1998). The larger hypervolume value means the better quality of the Pareto front approximation and distribution.

These metrics have been developed for evaluation performance of classical EMO algorithms. In this paper, we investigate preference-based EMO algorithms where the DM’s preferences are expressed by reference points, therefore we must consider them when evaluating performance of the algorithms. Hence, we introduce one new metric and adapt the aforementioned metrics for evaluation of the preference-based EMO algorithms.

Fig. 3 Region of interest of the DM



In the case of a preference-based EMO algorithm, the DM is interested only in the objective vectors that dominate the achievable reference points, i.e. they get into the region of the DM’s interest (see Fig. 3). Therefore, we propose a *PR metric* (14) for evaluation the preference-based EMO algorithms according to percentage of solutions that get into the region of the DM’s interest.

$$PR = \frac{|\tilde{P}^r|}{|\tilde{P}|} \times 100. \tag{14}$$

Here \tilde{P}^r is a set of solutions that get into the region of the DM’s interest, \tilde{P} is a set of all the solutions, obtained by the EMO algorithm. The metric evaluates performance of the algorithm in sense of an ability to obtain the concentrated solutions satisfying the DM’s preferences.

As it is mentioned before, evaluation of hypervolume metric is already based on a given point. In this investigation for hypervolume metric evaluation, we consider the reference point provided by the DM instead of the nadir point that is commonly used for classical EMO algorithms.

Moreover, we have proposed to modify Spread metric (Filatovas et al. 2015), because incorporation of d_i^e distances in the metric [Formula (10)] can distort representation of diversity of the objective vectors obtained. We propose to eliminate d_i^e distance, and to calculate the Spread metric by the following formula:

$$SP_p = \frac{\sum_{j=1}^{|\tilde{P}|} |d_j - \hat{d}|}{|\tilde{P}| \hat{d}}. \tag{15}$$

In classical EMO algorithms all the obtained solutions are considered when evaluating GD and Spread metrics. Here we propose to consider only solutions from the region of the DM’s interest when evaluating GD (9) and Spread (15) metrics.

4 Investigation of the proposed memetic algorithm

In this section the performed computational experiments with the proposed memetic algorithm R-NSGA-II with MOSASS as well as the obtained results are described.

The comparative analysis with the original R-NSGA-II algorithm is carried out. The performance of the algorithms is evaluated by metrics, described in Sect. 3.2.

A set of well-known test problems has been considered: two-objective problems ZDT1–ZDT4, ZDT6, and three-objective problems DTLZ1–DTLZ7 (Deb et al. 2002). It should be noted, that these benchmark problems has different complexity and characteristics as convexity, concavity, discontinuity, non-uniformity. Moreover, some of them have many local Pareto fronts. A definition of the local Pareto front is introduced in Deb (1999).

Each experiment has been performed for 100 independent runs using different initial populations and average results have been evaluated. We selected a population size of 100 individuals and 100 generations for the problems with two objectives, and a population size of 150 individuals and 150 generations for the three-objective problems, as such population sizes are enough to sufficiently represent an approximation of a region of the Pareto front. We select relative small number of generation in order to find good enough approximation in reasonable computation time.

The value of the clustering parameter $\delta = 0.0001$ was fixed as it ensures both diversity and concentration of the obtained solutions (Siegmund et al. 2012). Several trials has been performed for different combinations of parameters (E_G , E_L): (1000, 100), (1000, 200), (1000, 400), (2000, 100), (2000, 200), (2000, 400). The value E_G defines the number of function evaluations after which local search MOSASS is performed, and E_L sets the number of function evaluations to be performed during each local search.

Preference information provided by the DM that is expressed as a reference point is required for the evaluated algorithms, therefore various reference points were selected. The used reference points, the numbers of objectives and variables for each test problem considered are presented in Table 1. All the reference points are achievable—it means that all objectives can be improved at the same time without having to impair any of them.

Table 1 Test problems and reference points used in the evaluated algorithms

Problem	Number of objectives	Number of variables	Reference point
ZDT1	2	30	(0.50, 0.90)
ZDT2	2	30	(0.90, 0.60)
ZDT3	2	30	(0.35, 0.85)
ZDT4	2	10	(0.90, 0.60)
ZDT6	2	10	(0.90, 0.60)
DTLZ1	3	7	(0.10, 0.20, 0.40)
DTLZ2	3	12	(0.60, 0.90, 0.50)
DTLZ3	3	12	(0.60, 0.90, 0.50)
DTLZ4	3	12	(0.60, 0.90, 0.50)
DTLZ5	3	12	(0.60, 0.50, 0.95)
DTLZ6	3	12	(0.50, 0.80, 0.70)
DTLZ7	3	22	(0.10, 0.80, 2.70)

Table 2 Values of *PR* metric

Problem	R-NSGA-II	R-NSGA-II with MOSASS					
		$E_G = 1000$			$E_G = 2000$		
		$E_L = 100$	$E_L = 200$	$E_L = 400$	$E_L = 100$	$E_L = 200$	$E_L = 400$
ZDT1	<i>99.96</i>	99.29	99.20	91.85	99.85	99.70	99.58
ZDT2	99.68	98.68	98.17	74.03	99.88	99.79	98.04
ZDT3	<i>100.00</i>	99.99	99.65	90.76	<i>100.00</i>	<i>100.00</i>	99.91
ZDT4	99.62	81.23	85.65	74.80	99.63	99.47	98.45
ZDT6	95.14	92.92	88.74	71.54	95.05	95.20	94.88
DTLZ1	0.00	25.00	16.67	22.88	29.17	24.54	18.33
DTLZ2	<i>100.00</i>	99.98	99.98	99.98	99.99	99.99	99.98
DTLZ3	0.00	0.00	0.00	0.00	0.00	0.00	0.00
DTLZ4	99.67	90.92	93.01	94.96	99.93	99.99	84.84
DTLZ5	<i>100.00</i>	99.83	99.82	99.74	99.86	99.99	99.75
DTLZ6	95.89	83.46	87.04	80.86	88.65	90.76	93.24
DTLZ7	0.00	0.00	0.00	0.00	0.00	0.00	0.00

After the final population has been obtained, the number of the objective vectors that get into the region of the DM's interest is calculated and its percentage of the whole final population, i.e. value of *PR* metric (14) is evaluated. The mean values of *PR* for all performed runs with each test problem are presented in Table 2 for the original R-NSGA-II algorithm and for each trial of the memetic R-NSGA-II with MOSASS algorithm. The best average values of *PR* metric are presented in Italic and Bold italic. The values obtained by the proposed R-NSGA-II with MOSSAS algorithm which are better than obtained by original R-NSGA-II are marked in bold. One can see that the majority of the final population solutions get into the region of the DM's interest for all the test problems except DTLZ1, DTLZ3, and DTLZ7. It means that the fixed number of function evaluations is enough to find crowded set of solutions when running the investigated algorithms. DTLZ1, DTLZ3, and DTLZ7 problems are hard to approximate in limited number of function evaluations, therefore *PR* values are low in these cases. However, in the case of DTLZ1 problem, the proposed R-NSGA-II with MOSSAS algorithm manages to obtain some solutions in the region of the DM's interest unlike the original R-NSGA-II algorithm. For further performance evaluation of the algorithms we will not consider the DTLZ1, DTLZ3, and DTLZ7 problems.

The mean values of Generational Distance (9), Spread (15) and Hypervolume metrics and their confidence intervals (95 % confidence level) for each test problem are presented in Tables 3, 4 and 5, respectively. The best average values (lowest for Generational Distance and Spread, highest for Hypervolume) of the calculated metrics are presented with italics. The values obtained by the proposed R-NSGA-II with MOSSAS algorithm which are better than obtained by original R-NSGA-II are marked in bold. Table 3 shows that the proposed R-NSGA-II with MOSSAS algorithm approximates the Pareto front of two-objective problems ZDT2, ZDT3, ZDT4, ZDT6, and three-

Table 3 Mean and confidence intervals of generational distance metric

Problem	R-NSGA-II		R-NSGA-II with MOSASS			
			$E_G = 2000$			
			$E_G = 1000$		$E_G = 400$	
	$E_L = 100$	$E_L = 200$	$E_L = 400$	$E_L = 100$	$E_L = 200$	$E_L = 400$
ZDT1	4.54E-04 ±5.86E-05	1.16E-03 ±9.29E-04	1.67E-03 ±2.99E-04	7.15E-04 ±5.03E-04	6.03E-04 ±4.14E-04	7.43E-04 ±6.49E-04
ZDT2	1.05E-03 ±8.89E-04	5.91E-04 ±5.20E-05	2.18E-03 ±3.58E-04	4.71E-04 ±5.12E-05	4.48E-04 ±4.83E-05	5.89E-04 ±1.53E-04
ZDT3	8.62E-03 ±3.24E-03	8.16E-04 ±2.29E-04	1.90E-03 ±1.72E-03	4.21E-03 ±3.24E-03	9.59E-04 ±3.92E-04	6.50E-04 ±1.57E-04
ZDT4	5.37E-02 ±1.63E-02	7.89E-02 ±2.06E-02	1.19E-01 ±2.55E-02	4.04E-02 ±1.46E-02	5.99E-02 ±1.77E-02	5.58E-02 ±1.69E-02
ZDT6	3.41E-04 ±3.87E-05	3.69E-04 ±4.00E-05	5.78E-04 ±1.08E-04	3.16E-04 ±4.04E-05	3.17E-04 ±3.69E-05	3.37E-04 ±3.61E-05
DTLZ2	4.92E-03 ±2.13E-04	3.59E-03 ±3.65E-04	2.99E-03 ±2.58E-04	3.49E-03 ±2.40E-04	3.25E-03 ±2.12E-04	3.31E-03 ±3.60E-04
DTLZ4	2.52E-03 ±1.65E-04	2.30E-03 ±1.51E-04	2.24E-03 ±1.53E-04	2.57E-03 ±1.85E-04	2.28E-03 ±1.60E-04	2.27E-03 ±1.49E-04
DTLZ5	5.32E-04 ±5.82E-05	6.03E-04 ±1.19E-04	7.32E-04 ±1.73E-04	6.62E-04 ±2.60E-04	5.45E-04 ±8.88E-05	6.89E-04 ±2.05E-04
DTLZ6	9.87E-04 ±2.77E-04	1.48E-03 ±6.70E-04	2.09E-02 ±3.56E-03	1.77E-03 ±1.60E-03	2.48E-03 ±1.03E-03	2.86E-03 ±8.23E-04

Table 4 Mean and confidence intervals of Spread metric

Problem	R-NSGA-II with MOSASS														
	R-NSGA-II				$E_G = 2000$										
	$E_G = 1000$				$E_L = 200$		$E_L = 400$		$E_L = 100$		$E_L = 200$		$E_L = 400$		
ZDT1	2.96E-02 ±1.05E-02	2.17E-02 ±9.69E-03	2.09E-02 ±1.09E-02	1.05E-02 ±3.12E-03	2.09E-02 ±1.09E-02	2.62E-02 ±8.18E-03	2.62E-02 ±8.18E-03	2.62E-02 ±8.18E-03	2.62E-02 ±8.18E-03	2.62E-02 ±8.18E-03	2.92E-02 ±1.02E-02	2.92E-02 ±1.02E-02	2.92E-02 ±1.02E-02	2.92E-02 ±1.02E-02	1.80E-02 ±5.95E-03
ZDT2	1.81E-02 ±4.31E-03	2.07E-02 ±6.72E-03	1.53E-02 ±6.95E-03	1.28E-02 ±2.91E-03	1.53E-02 ±6.95E-03	2.17E-02 ±6.23E-03	2.17E-02 ±6.23E-03	2.17E-02 ±6.23E-03	2.17E-02 ±6.23E-03	2.17E-02 ±6.23E-03	2.00E-02 ±4.37E-03	2.00E-02 ±4.37E-03	2.00E-02 ±4.37E-03	2.00E-02 ±4.37E-03	1.94E-02 ±7.07E-03
ZDT3	2.43E-02 ±7.52E-03	2.67E-02 ±6.70E-03	2.47E-02 ±7.07E-03	1.76E-02 ±8.48E-03	2.47E-02 ±7.07E-03	2.09E-02 ±5.61E-03	2.09E-02 ±5.61E-03	2.09E-02 ±5.61E-03	2.09E-02 ±5.61E-03	2.09E-02 ±5.61E-03	2.34E-02 ±6.39E-03	2.34E-02 ±6.39E-03	2.34E-02 ±6.39E-03	2.34E-02 ±6.39E-03	2.34E-02 ±5.75E-03
ZDT4	2.63E-02 ±7.81E-03	2.49E-02 ±6.94E-03	2.61E-02 ±6.18E-03	2.94E-02 ±8.03E-03	2.61E-02 ±6.18E-03	2.58E-02 ±9.38E-03	2.58E-02 ±9.38E-03	2.58E-02 ±9.38E-03	2.58E-02 ±9.38E-03	2.58E-02 ±9.38E-03	2.61E-02 ±7.46E-03	2.61E-02 ±7.46E-03	2.61E-02 ±7.46E-03	2.61E-02 ±7.46E-03	2.32E-02 ±7.28E-03
ZDT6	2.41E-02 ±6.79E-03	1.83E-02 ±3.84E-03	2.51E-02 ±5.23E-03	1.93E-02 ±4.51E-03	2.51E-02 ±5.23E-03	2.58E-02 ±5.84E-03	2.58E-02 ±5.84E-03	2.58E-02 ±5.84E-03	2.58E-02 ±5.84E-03	2.58E-02 ±5.84E-03	2.40E-02 ±6.81E-03	2.40E-02 ±6.81E-03	2.40E-02 ±6.81E-03	2.40E-02 ±6.81E-03	2.59E-02 ±7.45E-03
DTLZ2	4.85E-03 ±1.42E-03	5.32E-03 ±1.38E-03	4.30E-03 ±7.82E-04	5.83E-03 ±1.82E-03	4.30E-03 ±7.82E-04	4.63E-03 ±9.89E-04	4.63E-03 ±9.89E-04	4.63E-03 ±9.89E-04	4.63E-03 ±9.89E-04	4.63E-03 ±9.89E-04	3.60E-03 ±6.01E-04	3.60E-03 ±6.01E-04	3.60E-03 ±6.01E-04	3.60E-03 ±6.01E-04	5.66E-03 ±1.47E-03
DTLZ4	9.64E-03 ±2.39E-03	1.05E-02 ±2.34E-03	1.11E-02 ±2.02E-03	1.13E-02 ±2.24E-03	1.11E-02 ±2.02E-03	6.64E-03 ±1.52E-03	6.64E-03 ±1.52E-03	6.64E-03 ±1.52E-03	6.64E-03 ±1.52E-03	6.64E-03 ±1.52E-03	7.03E-03 ±1.41E-02	7.03E-03 ±1.41E-02	7.03E-03 ±1.41E-02	7.03E-03 ±1.41E-02	1.41E-02 ±2.39E-03
DTLZ5	9.48E-03 ±4.44E-03	1.36E-02 ±3.55E-03	1.61E-02 ±6.05E-03	1.43E-02 ±4.99E-03	1.61E-02 ±6.05E-03	1.69E-02 ±6.38E-03	1.69E-02 ±6.38E-03	1.69E-02 ±6.38E-03	1.69E-02 ±6.38E-03	1.69E-02 ±6.38E-03	1.33E-02 ±5.54E-03	1.33E-02 ±5.54E-03	1.33E-02 ±5.54E-03	1.33E-02 ±5.54E-03	1.55E-02 ±5.72E-03
DTLZ6	9.53E-03 ±2.64E-03	1.04E-02 ±1.99E-03	1.11E-02 ±3.03E-03	6.91E-03 ±1.06E-03	1.11E-02 ±3.03E-03	1.06E-02 ±3.19E-03	1.06E-02 ±3.19E-03	1.06E-02 ±3.19E-03	1.06E-02 ±3.19E-03	1.06E-02 ±3.19E-03	1.28E-02 ±3.12E-03	1.28E-02 ±3.12E-03	1.28E-02 ±3.12E-03	1.28E-02 ±3.12E-03	1.13E-02 ±2.91E-03

Table 5 Mean and confidence intervals of HV metric

Problem	R-NSGA-II with MOSASS											
	R-NSGA-II			$E_G = 1000$								
	$E_L = 100$	$E_L = 200$	$E_L = 400$	$E_G = 2000$			$E_G = 4000$			$E_G = 6000$		
ZDT1	1.13E-01 ±6.67E-02	1.37E-01 ±7.34E-02	1.49E-01 ±7.66E-02	1.72E-01 ±8.22E-02	1.17E-01 ±6.80E-02	1.22E-01 ±6.94E-02	1.34E-01 ±7.27E-02	1.43E-01 ±7.51E-02	1.23E-01 ±6.95E-02	1.27E-01 ±7.07E-02	1.18E-01 ±6.80E-02	1.28E-01 ±7.10E-02
ZDT2	2.74E-02	3.43E-02 ±3.68E-02	3.92E-02 ±3.93E-02	4.71E-02 ±4.31E-02	2.89E-02 ±3.37E-02	3.03E-02 ±3.45E-02	3.40E-02 ±3.66E-02	4.26E-02 ±4.09E-02	8.68E-02 ±5.85E-02	3.95E-02 ±3.94E-02	4.06E-02 ±4.00E-02	4.34E-02 ±4.13E-02
ZDT3	3.61E-02	4.26E-02 ±4.09E-02	4.84E-02 ±4.37E-02	8.68E-02 ±5.85E-02	3.95E-02 ±3.94E-02	4.06E-02 ±4.00E-02	4.34E-02 ±4.13E-02	1.43E-01 ±7.51E-02	1.23E-01 ±6.95E-02	1.27E-01 ±7.07E-02	1.18E-01 ±6.80E-02	1.28E-01 ±7.10E-02
ZDT4	1.13E-01	1.37E-01 ±6.67E-02	1.49E-01 ±7.66E-02	1.72E-01 ±8.22E-02	1.17E-01 ±6.80E-02	1.22E-01 ±6.94E-02	1.34E-01 ±7.27E-02	1.43E-01 ±7.51E-02	1.23E-01 ±6.95E-02	1.27E-01 ±7.07E-02	1.18E-01 ±6.80E-02	1.28E-01 ±7.10E-02
ZDT6	7.28E-02	7.40E-02 ±5.40E-02	7.50E-02 ±5.43E-02	7.73E-02 ±5.52E-02	7.30E-02 ±5.36E-02	7.31E-02 ±5.36E-02	7.34E-02 ±5.37E-02	7.40E-02 ±5.40E-02	7.50E-02 ±5.43E-02	7.30E-02 ±5.36E-02	7.31E-02 ±5.36E-02	7.34E-02 ±5.37E-02
DTLZ2	3.61E-03	3.86E-03 ±1.19E-02	3.73E-03 ±1.21E-02	3.84E-03 ±1.23E-02	3.62E-03 ±1.19E-02	3.64E-03 ±1.20E-02	3.89E-03 ±1.24E-02	3.86E-03 ±1.23E-02	3.73E-03 ±1.21E-02	3.62E-03 ±1.19E-02	3.64E-03 ±1.20E-02	3.89E-03 ±1.24E-02
DTLZ4	4.43E-04	5.38E-04 ±4.18E-03	5.27E-04 ±4.55E-03	5.76E-04 ±4.76E-03	4.45E-04 ±4.19E-03	4.32E-04	6.45E-04 ±5.04E-03	5.38E-04 ±4.60E-03	5.27E-04 ±4.55E-03	4.45E-04 ±4.19E-03	4.32E-04	6.45E-04 ±5.04E-03
DTLZ5	1.58E-03	2.17E-03 ±7.89E-03	1.99E-03 ±8.85E-03	2.31E-03 ±9.53E-03	1.93E-03 ±8.72E-03	1.77E-03 ±8.35E-03	2.28E-03 ±9.47E-03	2.17E-03 ±7.89E-03	1.99E-03 ±8.85E-03	1.93E-03 ±8.72E-03	1.77E-03 ±8.35E-03	2.28E-03 ±9.47E-03
DTLZ6	2.98E-02	2.99E-02 ±3.43E-02	2.99E-02 ±3.43E-02	2.72E-02	2.99E-02 ±3.43E-02	3.00E-02 ±3.44E-02	3.01E-02 ±3.44E-02	2.99E-02 ±3.43E-02	2.99E-02 ±3.43E-02	2.99E-02 ±3.43E-02	3.00E-02 ±3.44E-02	3.01E-02 ±3.44E-02

objective problems DTLZ2, DTLZ4 better. The proposed algorithm is superior in cases of almost all combinations of parameters E_G , E_L for problems ZDT2, ZDT3, DTLZ2, DTLZ4. Moreover, for problems ZDT2, ZDT3, ZDT6, DTLZ2, confidence intervals are not overlapped comparing with the original R-NSGA-II algorithm. Although the results obtained by the original algorithm are better in the case of ZDT1 and DTLZ5 problems, however for these problems the confidence intervals for some combinations of parameters E_G , E_L are overlapping, hence the difference between the obtained results is not essential.

The mean values of Spread metric (see Table 4), obtained by the proposed algorithm are lower for all problems except of DTLZ5. It means that the non-dominated objective vectors are more evenly distributed on the part of the Pareto front covered by the region of the DM's interest. For problem DTLZ5 the original algorithm has better performance considering the Spread metric, however Spread values are small enough in the investigated algorithms, hence sufficient distribution of the obtained objective vectors is ensured in both cases.

HV metric estimates both precision of the Pareto front approximation and distribution of the objective vectors. Reference points are selected as the points R for HV estimation (see Table 1). Table 5 shows that in almost all cases the proposed algorithm has better performance considering HV value, and higher E_L value enables obtaining a higher HV value.

5 Conclusions

The memetic preference-based evolutionary multi-objective algorithm has been proposed in this paper. It is an extension of the R-NSGA-II algorithm where the local search strategy based on single agent stochastic search MOSASS is incorporated. The proposed memetic algorithm enables improving of quality of the Pareto front approximation and obtaining more even distribution of the approximating objective vectors.

The proposed memetic R-NSGA-II with MOSASS algorithm has been experimentally compared with the original R-NSGA-II algorithm by solving the widely-used benchmark test problems with the limited number of function evaluations. To evaluate the preference-based EMO algorithms we have proposed a new PR metric and adapted the well-known performance metrics—Generational Distance, Spread and Hypervolume, where only solutions from the region of the DM's interest are considered.

Experimental investigation has shown that, in most of the analysed cases, the proposed algorithm approximates the Pareto front better than the original R-NSGA-II algorithm by measuring the approximation quality using the Generational Distance. The values of Spread metric has shown that the solutions obtained by the proposed memetic algorithm are more evenly distributed. In the sense of Hypervolume metric, that estimates both approximation quality and distribution, the proposed algorithm is superior for all analysed test problems.

Acknowledgements This research is funded by a Grant (No. MIP-051/2014) from the Research Council of Lithuania.

References

- Auger A, Bader J, Brockhoff D, Zitzler E (2009) Articulating user preferences in many-objective problems by sampling the weighted hypervolume. In: Proceedings of the 11th annual conference on genetic and evolutionary computation, ACM, pp 555–562
- Caponio A, Neri F (2009) Integrating cross-dominance adaptation in multi-objective memetic algorithms. Springer, Berlin
- Deb K (1999) Multi-objective genetic algorithms: problem difficulties and construction of test problems. *Evolut Comput* 7:205–230
- Deb K (2001) Multi-objective optimization using evolutionary algorithms. Wiley, Hoboken
- Deb K, Kumar A (2007) Light beam search based multi-objective optimization using evolutionary algorithms. In: 2007 IEEE congress on evolutionary computation (CEC), pp 2125–2132
- Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evolut Comput* 6(2):182–197
- Deb K, Sundar J, Udaya Bhaskara Rao N, Chaudhuri S (2006) Reference point based multi-objective optimization using evolutionary algorithms. *Int J Comput Intell Res* 2(3):273–286
- Deb K, Thiele L, Laumanns M, Zitzler E (2002) Scalable multi-objective optimization test problems. In: Proceedings of the world on congress on computational intelligence, pp 825–830
- Filatovas E, Kurasova O, Sindhya K (2015) Synchronous R-NSGA-II: an extended preference-based evolutionary algorithm for multi-objective optimization. *Informatica* 26(1):33–50
- Goel T, Deb K (2002) Hybrid methods for multi-objective evolutionary algorithms. In: Proceedings of the fourth Asia-Pacific conference on simulated evolution and learning (SEAL02), pp 188–192
- Gong M, Liu F, Zhang W, Jiao L, Zhang Q (2011) Interactive MOEA/D for multi-objective decision making. In: Proceedings of the 13th annual conference on genetic and evolutionary computation, ACM, pp 721–728
- Ishibuchi H, Murata T (1998) A multi-objective genetic local search algorithm and its application to flowshop scheduling. *IEEE Trans Syst Man Cybern Part C Appl Rev* 28(3):392–403
- Jain H, Deb K (2014) An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part II: handling constraints and extending to an adaptive approach. *IEEE Trans Evolut Comput* 18(4):602–622
- Knowles JD, Corne DW (2000) Approximating the nondominated front using the Pareto archived evolution strategy. *Evolut Comput* 8(2):149–172
- Lančinskas A, Ortigosa PM, Žilinskas J (2013) Multi-objective single agent stochastic search in non-dominated sorting genetic algorithm. *Nonlinear Anal Model Control* 18(3):293–313
- Lančinskas A, Žilinskas J, Ortigosa PM (2011) Local optimization in global multi-objective optimization algorithms. In: 2011 third world congress on nature and biologically inspired computing (NaBIC), pp 323–328
- López-Jaimes A, Coello Coello CA (2014) Including preferences into a multiobjective evolutionary algorithm to deal with many-objective engineering optimization problems. *Inf Sci* 277:1–20
- Mejía JAH, Schütze O, Deb K (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, Springer, pp 247–260
- Miettinen K (1999) Nonlinear multiobjective optimization. Springer, Berlin
- Mohammadpour A, Dehghani A, Byagowi Z (2013) Using R-NSGA-II in the transmission expansion planning problem for a deregulated power system with wind farms. *Int J Eng Pract Res* 2(4):201–204
- Murata T, Nozawa H, Tsujimura Y, Gen M, Ishibuchi H (2002) Effect of local search on the performance of cellular multiobjective genetic algorithms for designing fuzzy rule-based classification systems. In: Proceedings of the world on congress on computational intelligence, vol 1, IEEE, pp 663–668
- Purshouse RC, Deb K, Mansor MM, Mostaghim S, Wang R (2014) A review of hybrid evolutionary multiple criteria decision making methods. In: 2014 IEEE congress on evolutionary computation (CEC), pp 1147–1154
- Ray T, Asafuddoula M, Isaacs A (2013) A steady state decomposition based quantum genetic algorithm for many objective optimization. In: 2013 IEEE congress on evolutionary computation (CEC), pp 2817–2824

- Ruiz AB, Saborido R, Luque M (2015) A preference-based evolutionary algorithm for multiobjective optimization: the weighting achievement scalarizing function genetic algorithm. *J Glob Optim* 62(1):101–129
- Siegmund F, Bernedixen J, Pehrsson L, Ng AH, Deb K (2012) Reference point-based evolutionary multi-objective optimization for industrial systems simulation. In: *Proceedings of the winter simulation conference*
- Siegmund F, Ng AH, Deb K (2012) Finding a preferred diverse set of Pareto-optimal solutions for a limited number of function calls. In: *2012 IEEE congress on evolutionary computation (CEC)*, pp 1–8
- Sindhya K, Miettinen K, Deb K (2013) A hybrid framework for evolutionary multi-objective optimization. *IEEE Trans Evolut Comput* 17(4):495–511
- Sindhya K, Ruiz AB, Miettinen K (2011) A preference based interactive evolutionary algorithm for multi-objective optimization: PIE. In: *6th international conference on evolutionary multi-criterion optimization, EMO 2011, Springer*, pp 212–225
- Sindhya K, Sinha A, Deb K, Miettinen K (2009) Local search based evolutionary multi-objective optimization algorithm for constrained and unconstrained problems. In: *2009 IEEE congress on evolutionary computation (CEC)*, IEEE, pp 2919–2926
- Solis F, Wets B (1981) Minimization by random search techniques. *Math Oper Res* 6(1):19–30
- Talbi EG (2009) *Metaheuristics: from design to implementation*, vol 74. Wiley, Hoboken
- Thiele L, Miettinen K, Korhonen PJ, Molina J (2009) A preference-based evolutionary algorithm for multi-objective optimization. *Evolut Comput* 17(3):411–436
- Zavala GR, Nebro AJ, Luna F, Coello CAC (2014) A survey of multi-objective metaheuristics applied to structural optimization. *Struct Multidiscip Optim* 49(4):537–558
- Zhang Q, Li H (2007) MOEA/D: a multiobjective evolutionary algorithm based on decomposition. *IEEE Trans Evolut Comput* 11(6):712–731
- Zhou A, Qu BY, Li H, Zhao SZ, Suganthan PN, Zhang Q (2011) Multiobjective evolutionary algorithms: a survey of the state of the art. *Swarm Evolut Comput* 1(1):32–49
- Zitzler E, Künzli S (2004) Indicator-based selection in multiobjective search. In: *Parallel problem solving from nature-PPSN VIII*, Springer, pp 832–842
- Zitzler E, Laumanns M, Thiele L (2001) SPEA2: improving the strength Pareto evolutionary algorithm. *Tech rep*
- Zitzler E, Thiele L (1998) Multiobjective optimization using evolutionary algorithms—a comparative case study. In: *Parallel problem solving from nature-PPSN V*, Springer, pp 292–301