INFORMATICA, 2015, Vol. 26, No. 1, 33–50

DOI: http://dx.doi.org/10.15388/Informatica.2015.37

Synchronous R-NSGA-II: An Extended Preference-Based Evolutionary Algorithm for Multi-Objective Optimization

Ernestas FILATOVAS[1]*, Olga KURASOVA[1], Karthik SINDHYA[2]

[1]*Institute of Informatics and Mathematics, Vilnius University*
Akademijos 4, LT-08663 Vilnius, Lithuania
[2]*Department of Mathematical Information Technology, University of Jyvaskyla P.O. Box 35, FI-40014 University of Jyvaskyla, Finland*
e-mail: ernest.filatov@gmail.com, olga.kurasova@mii.vu.lt, karthik.sindhya@jyu.fi

Received: July 2014; accepted: February 2015

Abstract. Classical evolutionary multi-objective optimization algorithms aim at finding an approximation of the entire set of Pareto optimal solutions. By considering the preferences of a decision maker within evolutionary multi-objective optimization algorithms, it is possible to focus the search only on those parts of the Pareto front that satisfy his/her preferences. In this paper, an extended preference-based evolutionary algorithm has been proposed for solving multi-objective optimization problems. Here, concepts from an interactive synchronous NIMBUS method are borrowed and combined with the R-NSGA-II algorithm. The proposed synchronous R-NSGA-II algorithm uses preference information provided by the decision maker to find only desirable solutions satisfying his/her preferences on the Pareto front. Several scalarizing functions are used simultaneously so the several sets of solutions are obtained from the same preference information. In this paper, the experimental-comparative investigation of the proposed synchronous R-NSGA-II and original R-NSGA-II has been carried out. The results obtained are promising.

Key words: interactive multi-objective optimization, evolutionary multi-objective optimization, preference-based evolutionary algorithms, scalarizing function.

1. Introduction

Multi-objective optimization problems often arise in different fields of engineering. The main goal while solving the problems is to minimize (or maximize) several conflicting objectives (criteria). As a rule a decision maker (DM) deals with several optimal solutions called Pareto optimal solutions, and he/she should select the most preferable one. The most popular approaches for solving multi-objective optimization problems are evolutionary multi-objective optimization (EMO) and multiple criteria decision making (MCDM). In EMO, the main target is to find a set of well-converged and well-distributed non-dominated objective vectors (equally good solutions) that approximate the

*Corresponding author.

entire Pareto front (Deb, 2001; Coello *et al.*, 2002; Talbi, 2009; Lančinskas *et al.*, 2013). This set of solutions is subsequently presented to the DM, who finally chooses one among them as his/her preferred solution. Many EMO algorithms and modifications have been developed during the past two decades. The most widely used are NSGA-II (Deb *et al.*, 2002a), SPEA2 (Zitzler *et al.*, 2001), PAES (Knowles and Corne, 2000), PESA-II (Corne *et al.*, 2001), etc. However, most of EMO algorithms are not suitable for handling a large number of objectives (especially, more than three) (Knowles and Corne, 2007). During the last decade some studies have been carried out in order to extend usability of the EMO algorithms for many objective problems (Hughes, 2005; López Jaimes and Coello Coello, 2009), and several EMO algorithms to handle many objective problems were developed – MOEA/D (Zhang and Li, 2007), its extensions (Zhang *et al.*, 2009; Ray *et al.*, 2013; Asafuddoula *et al.*, 2013), and NSGA-III (Deb and Jain, 2012; Jain and Deb, 2014). However, the increased dimensionality of Pareto-optimal front poses several challenges such as, large computational cost, difficulty in visualization of the objective space appears, exponentially more number of points are required to represent a higher-dimensional Pareto front (Kurasova *et al.*, 2013), etc. So, the DM may be unable to revise all the obtained solutions and to select the best without support of interactive methods. Usually, the more objectives are incorporated in a problem, the more preference information provided by the DM is required for the search in the interesting places of the decision space.

In MCDM, the DM's preferences are important when we deal with multi-objective optimization problems, because the main goal is to find the most satisfactory solution for the DM without exploring the whole set of Pareto optimal solutions. The MCDM approaches usually are classified into a priori, posteriori and interactive methods, depending on when the preference information is asked from the DM (Miettinen, 1999). A comprehensive overview of MCDM methods is given in Branke *et al.* (2008). Some recent approaches related to the MCDM are presented in Zeng *et al.* (2013), Hashemi *et al.* (2014), Meng and Chen (2014). In MCDM it is common for a multi-objective problem to be scalarized into a single-objective problem taking into account the DM's preference information and solving it using a suitable mathematical programming technique to find the preferred Pareto optimal solution.

When we deal with complex problems, interactive multi-objective optimization methods can assist to solve them. In the interactive methods, the DM specifies preference information progressively during the solution process. After each iteration, the DM is provided with one or some Pareto optimal solutions that reflect his/her preferences. Such a strategy allows the DM to analyze the solutions and learn about the problem during the solving process, and "move" towards the most preferable solution (Petkus *et al.*, 2009). Interactive methods can be computationally inexpensive, as only solutions that satisfy the DM's preferences are generated. However, they may require involvement of the DM extensively as compared to other MCDM methods because of the iterative solving process. The preference information of the DM can be expressed in different ways, such as reference direction, reference points or other techniques. Interactive methods can be classified into at least three groups: weight, constraint, and reference point methods (Miettinen, 1999). In general, they differ from each other in terms of scalarizing functions and preference

information used. The comprehensive surveys of the interactive methods are presented in Vira and Haimes (1983), Miettinen (1999), Branke *et al.* (2008), Luque *et al.* (2011).

In order to support the DM better, the principles of interactive and a posteriori methods are hybridized to use advantages of both these approaches (Thiele *et al.*, 2009). Recently, incorporation of DM's preference information into evolutionary approaches has been actively investigated. So-called preference-based EMO algorithms focus on only some parts of the Pareto front that are interesting for the DM and a crowded set of non-dominated objective vectors is found. So, preference-based EMO algorithms allow to reduce the computational cost and more complex multi-objective problems can be solved. Moreover, fewer efforts are required from the DM when analyzing the obtained set of solutions, because only solutions from region of interest are presented for him/her. The well-known preference-based EMO approaches are the cone-domination based EMO (Branke *et al.*, 2001), the biased niching based EMO (Branke and Deb, 2005), the reference point based EMO (Deb *et al.*, 2006b; Thiele *et al.*, 2009; Siegmund *et al.*, 2012), the light beam approach based EMO (Deb and Kumar, 2007), the weighted hypervolume based EMO (Auger *et al.*, 2009), etc. The comprehensive survey of the preference-based EMO algorithms is presented in Purshouse *et al.* (2014). Most of these approaches can be transformed into interactive multi-objective methods simply by allowing the DM to adjust preferences during the solving process interactively. In the interactive methods, the preference information requested from the DM is usually much simpler than it required by a priori methods (Purshouse *et al.*, 2014). Moreover, in interactive methods, the DM controls the search process, he/she is able to modify the preferences, learns about potential solutions of the problem during the solving process, and makes reliable decision.

Recently, researches have shown a big interest in developing evolutionary-based interactive methods (Deb and Chaudhur, 2007; Jaszkiewicz and Branke, 2008; Molina *et al.*, 2009; Fowler *et al.*, 2010; Greco *et al.*, 2010; Sinha *et al.*, 2014; Ruiz *et al.*, 2014). Among a variety of proposed preference-based EMO algorithms and evolutionary-based multi-objective interactive methods a considerable attention is given to incorporation of scalarizing functions into EMO approaches. López Jaimes and Coello Coello (2014) have proposed an interactive technique that uses the achievement scalarizing function (Wierzbicki, 1980) based on the Chebyshev distance (Ehrgott, 2005). Earlier Gong *et al.* (2011) proposed an interactive version of the decomposition based multi-objective evolutionary algorithm where Chebyshev approach is used, too. In the paper of Ruiz *et al.* (2014), the authors have suggested a preference-based EMO algorithm where an achievement scalarizing function is incorporated (Miettinen and Mäkelä, 2002). In the paper of Thiele *et al.* (2009), the authors have proposed preference-based evolutionary algorithm (PBEA), where the binary quality indicator of indicator-based evolutionary algorithm (IBEA) (Zitzler and Künzli, 2004) is redefined using an achievement scalarizing function and reference points. In the mentioned approaches, only one scalarizing function is used. It means that such approaches lead the DM only to one region on the Pareto front. However, in literature there exists other scalarizing functions, all of which consider the same preference information from the DM and yield comparatively different sets of Pareto optimal solutions. In practice, there is no certainty that one scalarizing function can reflect the

DM's preference. Therefore, EMO algorithms should be developed, where more than one scalarizing function should be incorporated.

In this paper, we propose a new evolutionary preference-based algorithm where several scalarizing functions are used simultaneously. Concepts from the interactive reference point method – synchronous NIMBUS (Miettinen and Mäkelä, 2006) – are borrowed and combined with the R-NSGA-II algorithm (Deb *et al.*, 2006b; Siegmund *et al.*, 2012).

The remainder of this paper is organized as follows. A description of the multi-objective problem, concepts of the multi-objective interactive synchronous NIMBUS method, as well as EMO algorithms NSGA-II and R-NSGA-II are presented in Section 2. The proposed synchronous R-NSGA-II algorithm is introduced in Section 3. Following, the results of the experimental-comparative investigations are presented in Section 4. In Section 5, we draw conclusions and give directions of the future researches.

2. Background Concepts

In this research, we extend the preference-based multi-objective evolutionary algorithm R-NSGA-II by incorporating the concepts of an interactive synchronous NIMBUS method. Fundamentals of multi-objective optimization as well as description of the involved approaches are presented in the next subsections.

2.1. *Multi-Objective Optimization Problem Statement*

At first, a formulation of multi-objective optimization problem is introduced. Let us have $k \geqslant 2$ conflicting objectives, described by the functions $f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x})$, where $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ is a vector of variables (*decision vector*), n is the number of variables. A multi-objective minimization problem is formulated as follows Miettinen (1999):

$$\text{minimize } \mathbf{f}(\mathbf{x}) = \big[f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x})\big],$$

$$\text{subject to } \mathbf{x} \in \mathbf{S},$$

where $\mathbf{S}$ is a *decision space* (feasible set, bounded domain) in the n-dimensional Euclidean space $\mathbb{R}^n$; $\mathbf{f}(\mathbf{x}) \in \mathbb{R}^k$ is a *vector of objective functions*. Each vector $\mathbf{x} \in \mathbf{S}$ is called a *feasible solution*. The vector $\mathbf{z} = \mathbf{f}(\mathbf{x}) \in \mathbb{R}^k$ for a feasible solution $\mathbf{x}$ is called an *objective vector*. A set of the objective vectors composes the so-called *feasible criterion space* $\mathbf{Z}$ (feasible region). A point $\mathbf{x}^* \in \mathbf{S}$ is (globally) *Pareto optimal* if there does not exist another point $\mathbf{x} \in \mathbf{S}$ such that $f_i(\mathbf{x}) \leqslant f_i(\mathbf{x}^*)$ for all $i = 1, \ldots, k$, and $f_j(\mathbf{x}) < f_j(\mathbf{x}^*)$ for at least one j. Objective vectors are regarded as optimal if none of their components can be improved without deterioration to at least one of the other components. An objective vector $\mathbf{z}^* = \mathbf{f}(\mathbf{x}^*)$ is Pareto optimal if the corresponding point $\mathbf{x}^*$ is Pareto optimal. The set of all the Pareto optimal solutions is called a *Pareto set*. The region defined by the value of all objectives for all the Pareto set points is called a *Pareto front*.

For two objective vectors $\mathbf{z}, \mathbf{z}' \in \mathbf{Z}$, we say that $\mathbf{z}$ *dominates* $\mathbf{z}'$ iff that $z_i \leqslant z'_i$ for all $i = 1, \ldots, k$ and there exists one j such that $z_j < z'_j$. In EMO algorithms, the subset of solutions in a population whose objective vectors are not dominated by any other objective vector is called the *non-dominated set*, and the objective vectors are called the *non-dominated objective vectors*. The main aim of the EMO algorithms is to generate well-distributed non-dominated objective vectors as close as possible to the Pareto optimal front.

When we want to solve a multi-objective optimization problem, we are interested in finding the Pareto optimal solution that best satisfies the desires of the DM, a person who is interested in solving the problem and can express preferences in some way (Sindhya *et al.*, 2011). So, two more vectors that fix the ranges of the Pareto front are used – the *ideal objective vector* $\mathbf{z}^{\mathrm{id}} \in \mathbb{R}^k$, and the *nadir objective vector* $\mathbf{z}^{\mathrm{nad}} \in \mathbb{R}^k$. Lower bounds of the Pareto optimal set are available in the ideal objective vector, its components z_i^{id} are obtained by minimizing each of the objective functions individually subject to the feasible region. The upper bounds of the Pareto optimal set are expressed by the components z_i^{nad} of the nadir objective vector. A vector $\mathbf{z}^{\mathrm{u}}$, that is strictly better than $\mathbf{z}^{\mathrm{id}}$ is called the *utopian objective vector* ($z_i^{\mathrm{u}} = z_i^{\mathrm{id}} - \epsilon$, $i = 1, \ldots, k$ where ϵ is a small positive scalar) (Miettinen, 1999). Unfortunately, there exists no constructive way to obtain the exact nadir objective vector for non-linear problems so, typically it is approximated (Deb *et al.*, 2006a, 2010). In general, the ideal, utopian and nadir objective vectors correspond to a non-existent solutions.

2.2. *Synchronous NIMBUS*

NIMBUS (Miettinen and Mäkelä, 1995; Miettinen, 1999) is a popular interactive classification-based multi-objective optimization method, which is continuously being developed and applied to different engineering problems (Hakanen *et al.*, 2005, 2006; Miettinen and Mäkelä, 2006; Laukkanen *et al.*, 2010; Miettinen *et al.*, 2014). NIMBUS method and its extensions are realized in the decision support system IND-NIMBUS (Ojalehto *et al.*, 2007) that is aimed at solving complex non-linear multi-objective optimization problems. It is also realized in WWW-NIMBUS (Miettinen and Mäkelä, 2000) that is free for academic and research use.

In the NIMBUS method, new Pareto optimal solutions are generated by solving a scalarized problem which includes preference information given by the DM. Scalarization plays a very important role in multi-objective optimization (Miettinen, 1999; Miettinen and Mäkelä, 2006). By the usage of the scalarizing function, the multiple objectives and some preference information provided by a DM are transformed into a single objective function and a subproblem is formed. In Miettinen and Mäkelä (2002), Branke and Gardiner (2003) it was shown that by incorporating more scalarizing functions the more different optimal solutions can be obtained even in the case when exactly the same preference information is provided by the DM.

In synchronous NIMBUS approach (Miettinen and Mäkelä, 2006), the NIMBUS was extended by incorporating three scalarizing functions that are based on the same pref-

erence information. It should be noted, that the scalarizing functions involved in synchronous NIMBUS are based on reference points. As a rule, in the reference point methods, the DM is asked to specify a *reference point* $\bar{\mathbf{z}} \in \mathbb{R}^k$, that is consisted of desirable aspiration levels $\bar{z}_i$ for each objective function. The involved scalarizing functions in the synchronous NIMBUS approach are as follows Miettinen and Mäkelä (2002, 2006):

$$\text{STOM: minimize} \quad \max_{i=1,\dots,k}\left[\frac{f_i(\mathbf{x}) - z_i^{\text{u}}}{\bar{z}_i - z_i^{\text{u}}}\right] + \rho\sum_{i=1}^{k}\frac{f_i(\mathbf{x})}{\bar{z}_i - z_i^{\text{u}}}, \tag{1}$$

$$\text{subject to} \quad \mathbf{x} \in \mathbf{S},$$

$$\text{ASF: minimize} \quad \max_{i=1,\dots,k}\left[\frac{f_i(\mathbf{x}) - \bar{z}_i}{z_i^{\text{nad}} - z_i^{\text{u}}}\right] + \rho\sum_{i=1}^{k}\frac{f_i(\mathbf{x})}{z_i^{\text{nad}} - z_i^{\text{u}}}, \tag{2}$$

$$\text{subject to} \quad \mathbf{x} \in \mathbf{S},$$

$$\text{GUESS: minimize} \quad \max_{i=1,\dots,k}\left[\frac{f_i(\mathbf{x}) - z_i^{\text{nad}}}{z_i^{\text{nad}} - \bar{z}_i}\right] + \rho\sum_{i=1}^{k}\frac{f_i(\mathbf{x})}{z_i^{\text{nad}} - \bar{z}_i}, \tag{3}$$

$$\text{subject to} \quad \mathbf{x} \in \mathbf{S},$$

where a so-called augmentation coefficient $\rho > 0$ is a relatively small scalar. Using such an approach the DM may obtain several Pareto optimal solutions, where each of them reflects the DM's preferences and gives variety of different solutions. After that the DM can select the most preferred one as a final solution or use it for the further improvement.

2.3. *Evolutionary Multi-Objective Optimization Algorithms NSGA-II and R-NSGA-II*

Among the EMO algorithms NSGA-II is commonly used (Deb *et al.*, 2002a). It is an elitist algorithm, where the best solutions found are preserved. In addition, NSGA-II also uses a crowding comparison procedure, an explicit diversity preservation mechanism to obtain a well-distributed approximation of the Pareto optimal front. The steps of NSGA-II are described in Algorithm 1.

NSGA-II uses two types of fitness functions – the primary fitness function is the Pareto optimality, the secondary one is a crowding distance. In the paper of Deb *et al.* (2006b), the reference point-based NSGA-II algorithm (called R-NSGA-II) has been proposed, where the DM's preferences are expressed by reference points. In order to speed up and control of diversity, an extension of R-NSGA-II has been provided in Siegmund *et al.* (2012). The crowding distance used in NSGA-II is changed to the Euclidean distance from the reference points. The Step 3 of NSGA-II algorithm is changed so, that the parents with smaller Euclidean distances are preferred. The Step 5 is modified by such issues: each front of the population R is clustered, and only the representatives of the clusters are saved to the population P; if not all representatives of the clusters could be picked, only the representatives with smaller Euclidean distance are added into the population P.

Algorithm 1 NSGA-II

Step 1: Generate a random initial population P_0 of size N.

Step 2: Sort the population to different non-domination levels (fronts), and assign each individual a fitness equal to its non-domination level (1 is the best level).

Step 3: Create an offspring population of size N using binary tournament selection, recombination and mutation operations (parents with larger crowding distance are preferred if their non-domination levels are the same).

Step 4: Combine the parent and the offspring populations and create a population R.

Step 5: Reduce the population R to the population P of size N: sort the population R into different non-dominated fronts; fill the population P with individuals from population R starting from the best non-dominated front until the size of P is equal to N; if all the individuals in a front cannot be picked fully, calculate a crowding distance and add individuals with the largest distances into the population P.

Step 6: Check if the termination criterion is satisfied. If yes, go to Step 7, else return to Step 2.

Step 7: Stop.

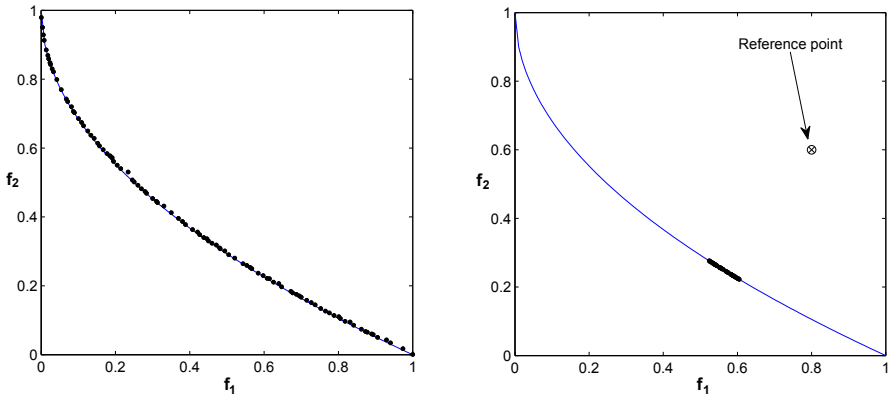

Fig. 1. Non-dominated objective vectors of ZDT1 problem, obtained by: (a) NSGA-II, (b) R-NSGA-II.

NSGA-II algorithm tries to approximate the whole Pareto front and to distribute the obtained non-dominated objective vectors evenly. On contrary, the obtained objective vectors by R-NSGA-II are concentrated on a particular region of the Pareto front approximated according to a reference point expressed by the DM. The differences of results, obtained by the NSGA-II and R-NSGA-II algorithms, are demonstrated using the popular test problems ZDT1 (2 objectives) and DTLZ2 (3 objectives) in Figs. 1 and 2.

3. Synchronous R-NSGA-II Algorithm

The original R-NSGA-II algorithm provides the concentrated solutions on a region of interest according to the DM's preferences. However, the preference information expressed

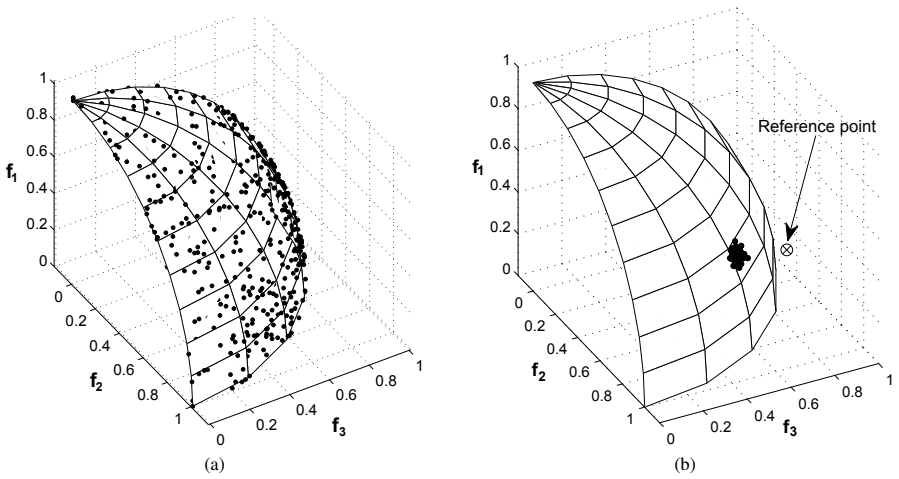

Fig. 2. Non-dominated objective vectors of DTLZ2 problem, obtained by: (a) NSGA-II, (b) R-NSGA-II.

by the DM is treated only in one way. Incorporating several scalarizing functions to the algorithm will enable to focus the search on several regions of interest using the same preference information. Usage of the various functions in the same approach simultaneously will increase a variety of the obtained non-dominated objective vectors, but still obeying the preference information of the DM. In this way, the DM will have access to multiple solutions that reflect his/her preference information and thereby can find his/her preferred solution. The DM will have more freedom and flexibility in the decision process. Here we propose the synchronous R-NSGA-II algorithm that is based on the concept of the R-NSGA-II algorithm involving three different scalarizing functions used simultaneously. The steps of the proposed synchronous R-NSGA-II are described in Algorithm 2, which involves a synchronous preference operator.

3.1. *Synchronous Preference Operator*

In order to coordinate the use of all three scalarizing functions in the synchronous R-NSGA-II algorithm, the synchronous preference operator is proposed. Here for every individual in each front $\bar{F}_i$ the values of the scalarizing functions (1), (2), (3) are evaluated. Next, all the individuals in $\bar{F}_i$ are ranked in ascending order according to the values of scalarizing functions, i.e. rank 1 being the individual with minimum value of the corresponding scalarizing function. Hence, for every individual up to 3 different ranks are available. The global rank of every individual being the minimum of all the ranks.

In Fig. 3, the Pareto front approximations, obtained the synchronous R-NSGA-II algorithm, are presented for ZDT1 and DTLZ2 problems. The non-dominated objective vectors are concentrated to three regions, although only one reference point is used. These concentration depends on the clustering parameter value δ ($0 < \delta \leqslant 1$) which must be set carefully. Too small value compress the solutions to three very crowded groups. Too high value does not allow formation of separated groups, therefore preference-based method loses its sense.

Algorithm 2 Synchronous R-NSGA-II

Step 1: Generate a random initial population P_0 of size N and set a generation count $t = 0$. Set values of recombination r and mutation m operations. Set a clustering parameter value δ ($0 < \delta \leqslant 1$).

Step 2: Sort the population to different non-domination levels (fronts) and assign for each objective vector a fitness equal to its non-domination level (1 is the best level). Calculate values of the three scalarizing functions and the global rank for each individual (see Section 3.1).

Step 3: Create an offspring population Q'_t using binary tournament selection (parents with smaller global rank are preferred), recombination and mutation operations.

Step 4: Combine the parent and offspring populations and create a population $R_t = P_t \cup Q'_t$.

Step 5: Perform non-dominated sorting to the population R_t and identify different fronts $\bar{F}_i$, $i = 1, 2, \ldots, p$.

Step 6: Set a new population $P_{t+1} = \emptyset$.

Step 7: Cluster each front $\bar{F}_i$, $i = 1, 2, \ldots, p$: the individuals are assigned to one cluster, if the smallest value of the three scalarizing functions differs less than δ (as in R-NSGA-II algorithm); $\bar{R}_t$ ($\bar{R}_t \subset R_t$) is a set of representatives of each cluster, $\bar{R}_{ti}$ ($\bar{R}_{ti} \in \bar{R}_t$) are representatives in i-th front.

Step 8: Set a count $i = 1$. As long as $|P_{t+1}| + |\bar{R}_{ti}| \leqslant N$, perform $P_{t+1} = P_{t+1} \cup \bar{R}_{ti}$, $i = i + 1$, if ($i > p$ and $|P_{t+1}| < N$) then set $\bar{F}_i = \bar{F}_i \setminus \bar{R}_{ti}$, $i = 1, 2, \ldots, p$ and go to Step 7.

Step 9: Add $(N - |P_{t+1}|)$ representatives of $\bar{F}_i$, $i = 1, 2, \ldots, p$ with the smallest global rank to P_{t+1}.

Step 10: Check if the termination criterion is satisfied. If yes, go to Step 11, else set $t = t + 1$ and return to Step 2.

Step 11: Stop.

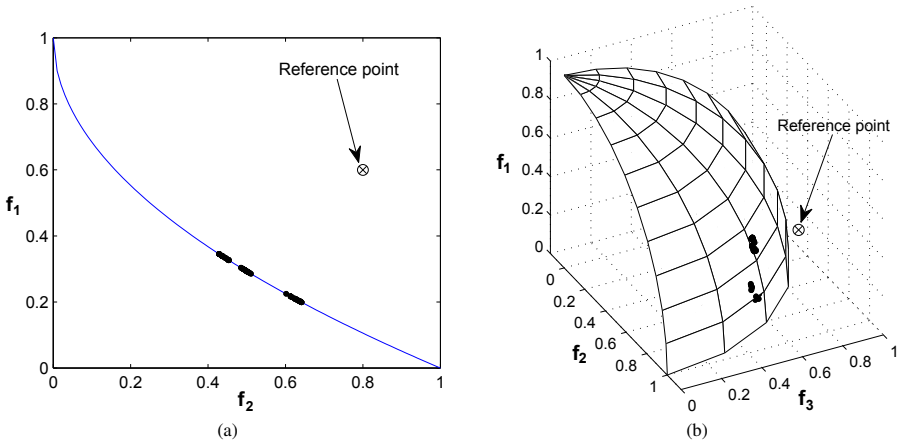

Fig. 3. Non-dominated objective vectors, obtained by the synchronous R-NSGA-II, when solving: (a) ZDT1, (b) DTLZ2 problems.

3.2. *Performance Metrics*

The comparison of the results of EMO algorithms is one of the main problems in multi-objective optimization. It is obvious, that performance metrics should be used in order to evaluate EMO algorithms. During the past few decades, many metrics have been introduced (Deb, 2001; Zhou *et al.*, 2011). The most popular performance metrics are as follows:

- The generational distance (GD) (Deb, 2001; Van Veldhuizen, 1999) measures the convergence of EMO algorithms.
- The inverted generational distance (IGD) (Zitzler *et al.*, 2003) measures both convergence and diversity of the obtained objective vectors simultaneously.
- The spacing (Schott, 1995) and spread (Deb, 2001; Deb *et al.*, 2002a) measure the distribution of the obtained objective vectors.
- The hypervolume metric (HV) (Zitzler and Thiele, 1998; Van Veldhuizen, 1999) is used to compare the non-dominated sets of objective vectors and performance of the algorithms, estimating convergence as well as diversity.

Summarizing, EMO algorithms are measured on two aspects: the convergence (how close the obtained non-dominated objective vectors are to the Pareto front), and the distribution of the obtained objective vectors.

Recently, some performance metrics were proposed for preference-based EMO algorithms. In Wickramasinghe *et al.* (2010), the authors have adapted HV metric for such algorithms, but this metric can be misleading depending on the location of the reference point. In Mohammadi *et al.* (2013), the authors have improved the metric by incorporating IGD in it. However, the proposed metric does not suit when evaluating performance of the preference-based algorithm with more than one crowded set of non-dominated objective vectors.

In this investigation, generational distance (GD) and spread-based performance metrics were used to compare two preference-based algorithms – original R-NSGA-II and proposed synchronous R-NSGA-II.

The generational distance for a set of non-dominated objective vectors Q is calculated by the formula (Deb, 2001):

$$GD = \frac{1}{|Q|}\sum_{j=1}^{|Q|} g_j, \tag{4}$$

where

$$g_j = \min_{l=1,\ldots,|Q'|}\left(\sum_{i=1}^{k}\left(f_i^{(j)} - f_i'^{(l)}\right)\right). \tag{5}$$

Here g_j is a distance between the j-th objective vector of Q and the nearest objective vector of a set of members of the Pareto optimal front Q'. $f_i^{(j)}$ and $f_i'^{(l)}$ are the i-th

objective function values of the j-th objective vector of the set Q and that of the l-th objective vector of the set Q', respectively.

Spread for a set of non-dominated objective vectors Q is calculated by the formula (Deb, 2001):

$$SP = \frac{\sum_{i=1}^{k} d_i^e + \sum_{j=1}^{|Q|} |d_j - \hat{d}|}{\sum_{i=1}^{k} d_i^e + |Q|\hat{d}}, \tag{6}$$

where d_j is a distance between neighboring objective vectors, $\hat{d}$ is an average of these distances, d_i^e is a distance between the extreme objective vectors of Q' and Q corresponding to the i-th objective function. In preference-based EMO methods non-dominated objective vectors are obtained from a region of the Pareto front, but not from the entire Pareto front as in classic EMO methods. So, incorporation of d_i^e distances in the metric 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 formula:

$$SP_p = \frac{\sum_{j=1}^{|Q|} |d_j - \hat{d}|}{|Q|\hat{d}}. \tag{7}$$

Moreover, as it shown in Fig. 3, some gaps between the obtained non-dominated objective vectors and some sets of the crowded objective vectors can appear, when we solve an optimization problem using the synchronous R-NSGA-II algorithm. In order to evaluate the diversity of the obtained objective vectors more precisely, the spread metric value should be calculated for each set of the crowded objective vectors individually, and the average of all spread values should be evaluated. However, the problem of the clustering of the obtained objective vectors arises. In this paper, we propose such a clustering way:

- Three Pareto optimal solutions and corresponding objective vectors (Solution1, Solution2, and Solution3) are obtained by optimizing scalarizing functions (1), (2), (3) using any exact single-objective optimization method (mathematical programming).
- Distances between the objective vectors corresponding to the optimal solutions and the non-dominated objective vectors, obtained by preference-based EMO, are calculated.
- The non-dominated objective vectors are assigned to different clusters (Cluster1, Cluster2, and Cluster3). The non-dominated objective vectors nearest to Solution1 are assigned to Cluster1, nearest to Solution2 – to Cluster2, and nearest to Solution3 – to Cluster3.

As it is shown in Miettinen and Mäkelä (2002) that in some cases depending on the selected reference point different scalarizing functions may provide the same Pareto solution. So, the number of clusters is equal to the number of different Pareto solutions. It is important to note that a non-dominated objective vector can be assigned only to one cluster (nearest one). So, it can happen that some clusters are empty. The spread should

be calculated for the non-dominated objective vectors from non-empty clusters. We propose to evaluate spread metric, when more than one crowded set of the non-dominated objective vectors are obtained, by such a formula:

$$SP_s = \frac{1}{s}\sum_{r=1}^{s}\left(\frac{1}{|Q_r|\hat{d}}\sum_{j=1}^{|Q_r|}|d_j - \hat{d}|\right), \tag{8}$$

where s is the number of non-empty clusters, Q_r is a set of the obtained non-dominated objective vectors, assigned to r-th cluster. In the case when $s = 1$ the formula (8) is identical to the formula (7).

4. Numerical Experiments and Discussion

In order to evaluate the proposed synchronous R-NSGA-II algorithm and to compare it with the original R-NSGA-II, computational experiments have been carried out. The well-known test problems for evaluation of multi-objective optimization algorithms have been considered: two-objective problems ZDT1–ZDT4, ZDT6, and three-objectives problems DTLZ1–DTLZ6 (Huband *et al.*, 2006). Both investigated algorithms require preference information provided by the DM that is expressed as a reference point, therefore various reference points were selected. The used reference points, the number of objectives and variables for each problem considered are presented in Table 1. All the reference points are achievable – it means that all objectives values can be improved at the same time, without having to impair any of them.

We have used 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. The clustering parameter δ controls the diversity of the obtained non-dominated objective vectors. As it is shown in Siegmund *et al.* (2012) the R-NSGA-II algorithm finds a concentrated set of non-dominated objective vectors when the value of

Table 1
Problems and reference points used in R-NSGA-II and synchronous R-NSGA-II algorithms.

Problem	Number of objectives	Number of variables	Reference point
ZDT1	2	30	(0.80, 0.60)
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)

Table 2
Average and confidence intervals of GD metric.

Problem	R-NSGA-II		Synchronous R-NSGA-II	
	Average	Confidence interval	Average	Confidence interval
ZDT1	2.85E–06	(2.36E–06, 3.34E–06)	**1.96E–06**	(1.73E–06, 2.18E–06)
ZDT2	**2.00E–06**	(1.77E–06, 2.24E–06)	2.24E–06	(1.94E–06, 2.54E–06)
ZDT3	3.90E–03	(3.45E–03, 4.35E–03)	**1.92E–03**	(1.66E–03, 2.18E–03)
ZDT4	**1.39E–02**	(7.21E–03, 2.06E–02)	1.74E–02	(8.73E–03, 2.60E–02)
ZDT6	**1.94E–06**	(1.71E–06, 2.16E–06)	2.07E–06	(1.83E–06, 2.31E–06)
DTLZ1	4.38E+00	(3.87E+00, 4.89E+00)	**1.89E+00**	(1.71E+00, 2.07E+00)
DTLZ2	1.47E–03	(9.50E–04, 1.99E–03)	**4.28E–05**	(3.27E–05, 5.29E–05)
DTLZ3	3.06E+01	(2.85E+01, 3.28E+01)	**2.20E+01**	(2.06E+01, 2.34E+01)
DTLZ4	8.83E–04	(4.04E–04, 1.36E–03)	**2.85E–04**	(2.35E–04, 3.35E–04)
DTLZ5	1.49E–05	(1.17E–05, 1.81E–05)	**1.02E–05**	(8.36E–06, 1.20E–05)
DTLZ6	**5.68E–02**	(2.07E–02, 9.28E–02)	5.87E–02	(2.14E–02, 9.59E–02)

Table 3
Average and confidence intervals of Spread metric.

Problem	R-NSGA-II		Synchronous R-NSGA-II	
	Average	Confidence interval	Average	Confidence interval
ZDT1	**1.18E–02**	(1.01E–02, 1.35E–02)	8.33E–02	(7.11E–02, 9.55E–02)
ZDT2	**1.02E–02**	(9.39E–03, 1.10E–02)	7.75E–02	(6.65E–02, 8.85E–02)
ZDT3	**1.44E–02**	(1.20E–02, 1.69E–02)	5.38E–02	(4.67E–02, 6.10E–02)
ZDT4	7.60E–02	(6.60E–02, 8.60E–02)	**3.70E–02**	(3.05E–02, 4.35E–02)
ZDT6	**8.14E–03**	(7.54E–03, 8.75E–03)	6.98E–02	(5.94E–02, 8.02E–02)
DTLZ1	**3.75E–02**	(3.11E–02, 4.40E–02)	4.17E–02	(3.62E–02, 4.73E–02)
DTLZ2	**7.80E–03**	(6.57E–03, 9.02E–03)	2.35E–02	(1.86E–02, 2.84E–02)
DTLZ3	5.89E–02	(5.16E–02, 6.63E–02)	**4.37E–02**	(3.79E–02, 4.94E–02)
DTLZ4	**8.53E–03**	(6.94E–03, 1.01E–02)	4.67E–02	(3.98E–02, 5.36E–02)
DTLZ5	**7.10E–03**	(6.03E–03, 8.17E–03)	8.43E–02	(7.19E–02, 9.66E–02)
DTLZ6	**4.69E–03**	(4.22E–03, 5.17E–03)	3.29E–02	(2.83E–02, 3.75E–02)

the parameter δ is relatively small. In this investigation, we have fixed the value of the parameter $\delta = 0.0001$.

The evaluated algorithms have been implemented in Matlab environment. The same computer (Intel Core i7-3632QM @2.2 GHz, 8.00 GB) has been used for experimental investigations.

Approximations of the Pareto front obtained by both algorithms are measured by the generational distance (GD) performance metric (4). Diversity of the obtained objective vectors is measured by spread metric (8) (in the R-NSGA-II case $s = 1$). We have carried out 100 independent runs with each test problem with various initial populations. The average values of the used metrics and the confidence intervals (95% confidence level) have been calculated and presented in Tables 2 and 3. The lowest average values of the calculated metrics are written in bold.

Table 2 shows that in more cases the synchronous R-NSGA-II algorithm approximates the Pareto front better, but the differences comparing with R-NSGA-II are not essential. Moreover, the advantage of synchronous R-NSGA-II is that in almost all the cases when

the average values are lower, the confidence intervals do not overlap with the corresponding intervals calculated for R-NSGA-II. On the contrary, for those test problems when the average values calculated for R-NSGA-II are lower the confidence intervals overlap with ones calculated for the synchronous R-NSGA-II. Both the algorithms have difficulties to approximate the complex test problems (DTLZ1 and DTLZ3) with many local Pareto fronts, and to converge to the global Pareto front in the limited number of generations. Such difficulties commonly appear with evolutionary algorithms including original NSGA-II (Deb *et al.*, 2002b; Zitzler *et al.*, 2003).

The average values of spread metric presented in Table 3 are slightly higher in the most cases of the synchronous R-NSGA-II algorithm. It means that the non-dominated objective vectors, obtained by the R-NSGA-II algorithm, are more evenly distributed on the part of the Pareto front. However, in both algorithms, the spread values are small, therefore sufficient distribution of the obtained objective vectors is ensured.

5. Conclusions and Future Works

The synchronous R-NSGA-II algorithm has been proposed in this paper. It is an extension of the evolutionary reference-based R-NSGA-II algorithm. The idea of synchronization is adapted from the synchronous NIMBUS method. The main difference of the proposed algorithm from original R-NSGA-II is that three scalarizing functions are used instead of the Euclidean distance. The advantage of the proposed synchronous R-NSGA-II algorithm is that it provides several crowded groups of the non-dominated objective vectors for the DM according the same preference information. Scalarizing functions are used in the selection of the parents for recombinations, and in the generation of the new population.

The proposed synchronous R-NSGA-II algorithm has been compared with the original R-NSGA-II solving the widely-used test problems. Two performance metrics – the generational distance and spread – are selected for the comparison of the algorithms. Moreover, the modified spread metric for performance evaluation of the preference-based EMO algorithms is proposed. The proposed metric is adapted for evaluation of the algorithms where the several sets of grouped objective vectors are obtained.

Experimental investigation has been showed that the synchronous R-NSGA-II algorithm approximates the Pareto front better than the original R-NSGA-II algorithm, and higher spread value is insignificant in the case of the synchronous R-NSGA-II algorithm.

In the future, a new interactive method for multi-objective optimization shall be developed on basis of the proposed synchronous R-NSGA-II algorithm. As the algorithm allows to obtain non-dominated objective vectors from several regions of interests, the variety of the obtained solutions, that are interesting for the DM is increasing. The interactive method would enable the DM to achieve the desired solution in few steps.

Acknowledgements. Ernestas Filatovas is supported by the postdoctoral fellowship funded by European Union Structural Funds project "Postdoctoral Fellowship Implementation in Lithuania" and Karthik Sindhya was funded by the TEKES – the Finnish Funding Agency for Technology and Innovation (the SIMPRO project).

References

Asafuddoula, M., Ray, T., Sarker, R. (2013). A decomposition based evolutionary algorithm for many objective optimization with systematic sampling and adaptive epsilon control. In: *Evolutionary Multi-Criterion Optimization*. Springer, Berlin, pp. 413–427.

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.

Branke, J., Deb, K. (2005). Integrating user preferences into evolutionary multi-objective optimization. In: *Knowledge Incorporation in Evolutionary Computation*. Springer, Berlin, pp. 461–477.

Branke, J., Kaußler, T., Schmeck, H. (2001). Guidance in evolutionary multi-objective optimization. *Advances in Engineering Software*, 32(6), 499–507.

Branke, J., Deb, K., Miettinen, K., Slowinski, R. (2008). *Multiobjective Optimization: Interactive and Evolutionary Approaches. Lecture Notes in Computer Science*, Vol. 5252. Springer, Berlin.

Buchanan, J., Gardiner, L. (2003). A comparison of two reference point methods in multiple objective mathematical programming. *European Journal of Operational Research*, 149(1), 17–34.

Coello, C.A.C., Van Veldhuizen, D.A.. Lamont, G.B. (2002). *Evolutionary Algorithms for Solving Multi-objective Problems*, Vol. 242. Springer, Berlin.

Corne, D.W., Jerram, N.R., Knowles, J.D., Oates, M.J. (2001). PESA-II: Region-based selection in evolutionary multiobjective optimization. In: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO'2001*, pp. 283–290.

Deb, K. (2001). *Multi-objective Optimization using Evolutionary Algorithms*. Wiley, Hoboken.

Deb, K., Chaudhuri, S. (2007). I-MODE: an interactive multi-objective optimization and decision-making using evolutionary methods. In: *Lecture Notes in Computer Science*, Vol. 4403, pp 788–802.

Deb, K., Jain, H. (2012). An improved NSGA-II procedure for many-objective optimization. Part I: Solving problems with box constraints. *KanGAL report* 2012009.

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. (2002a). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2), 182–197.

Deb, K., Thiele, L., Laumanns, M., Zitzler, E. (2002b). Scalable multi-objective optimization test problems. In: *Proceedings of the World on Congress on Computational Intelligence*, pp. 825–830.

Deb, K., Chaudhuri, S., Miettinen, K. (2006a). Towards estimating nadir objective vector using evolutionary approaches. In: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO-2006, New York*, pp. 643–650.

Deb, K., Sundar, J., Udaya Bhaskara Rao, N., Chaudhuri, S. (2006b). Reference point based multi-objective optimization using evolutionary algorithms. *International Journal of Computational Intelligence Research*, 2(3), 273–286.

Deb, K., Miettinen, K., Chaudhuri, S. (2010). Toward an estimation of nadir objective vector using a hybrid of evolutionary and local search approaches. *IEEE Transactions on Evolutionary Computation*, 14(6), 821–841.

Ehrgott, M. (2005). *Multicriteria Optimization*. Springer, Berlin.

Fowler, J.W., Gel, E.S., Köksalan, M.M., Korhonen, P., Marquis, J.L., Wallenius, J. (2010). Interactive evolutionary multi-objective optimization for quasi-concave preference functions. *European Journal of Operational Research*, 206(2), 417–425.

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, New York, pp. 721–728.

Greco, S., Matarazzo, B., Slowinski, R. (2010). Interactive evolutionary multiobjective optimization using dominance-based rough set approach. In: *2010 IEEE Congress on Evolutionary Computation, CEC*, pp. 1–8.

Hakanen, J., Miettinen, K., Mäkelä, M.M., Manninen, J. (2005). On interactive multiobjective optimization with NIMBUS® in chemical process design. *Journal of Multi-criteria Decision Analysis*, 13(2–3), 125–134.

Hashemi, S.S., Hajiagha, S.H.R., Amiri, M. (2014). Decision making with unknown data: development of ELECTRE method based on black numbers. *Informatica*, 25(1), 21–36.

Heikkola, E., Miettinen, K., Nieminen, P. (2006). Multiobjective optimization of an ultrasonic transducer using NIMBUS. *Ultrasonics*, 44(4), 368–380.

Huband, S., Hingston, P., Barone, L., While, L. (2006). A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation*, 10(5), 477–506.

Hughes, E.J. (2005). Evolutionary many-objective optimisation: many once or one many? In: *2005 IEEE Congress on Evolutionary Computation, CEC*, Vol. 1, pp. 222–227.

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 Transactions on Evolutionary Computation*, 18(4), 602–622.

Jaszkiewicz, A., Branke, J. (2008). Interactive multiobjective evolutionary algorithms. In: *Multiobjective Optimization. Lecture Notes in Computer Science*, Vol. 5252. Springer, Berlin, pp. 179–193.

Knowles, J.D., Corne, D.W. (2000). Approximating the nondominated front using the Pareto archived evolution strategy. *Evolutionary Computation*, 8(2), 149–172.

Knowles, J., Corne, D. (2007). Quantifying the effects of objective space dimension in evolutionary multiobjective optimization. In: *Evolutionary Multi-Criterion Optimization. Lecture Notes in Computer Science*, Vol. 4403. Springer, Berlin, pp. 757–771.

Kurasova, O., Petkus, T., Filatovas, E. (2013). Visualization of Pareto front points when solving multi-objective optimization problems. *Information Technology and Control*, 42(4), 353–361.

Lančinskas, A. Martinez Ortigosa, P., Žilinskas, J. (2013). Multi-objective single agent stochastic search in non-dominated sorting genetic algorithm. *Nonlinear Analysis: Modelling and Control*, 18(3), 293–313.

Laukkanen, T. Tveit, T.-M., Ojalehto, V., Miettinen, K., Fogelholm, C.-J. (2010). An interactive multi-objective approach to heat exchanger network synthesis. *Computers & Chemical Engineering*, 34(6), 943–952.

López Jaimes, A., Coello Coello, C.A. (2009). Some techniques to deal with many-objective problems. In: *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*. ACM, New York, pp. 2693–2696.

López-Jaimes, A., Coello Coello, C.A., (2014). Including preferences into a multiobjective evolutionary algorithm to deal with many-objective engineering optimization problems. *Information Sciences*, 277, 1–20.

Luque, M., Ruiz, F., Miettinen, K. (2011). Global formulation for interactive multiobjective optimization. *OR Spectrum*, 33(1), 27–48.

Meng, F., Chen, X. (2014). An approach to interval-valued hesitant fuzzy multi-attribute decision making with incomplete weight information based on hybrid Shapley operators. *Informatica*, 25(4), 617–642.

Miettinen, K. (1999). *Nonlinear Multiobjective Optimization*. Springer, Boston, 1999.

Miettinen, K., Mäkelä, M.M. (1995). Interactive bundle-based method for nondifferentiable multiobjeective optimization: NIMBUS. *Optimization*, 34(3), 231–246.

Miettinen, K., Mäkelä, M.M. (2000). Interactive multiobjective optimization system WWW-NIMBUS on the internet. *Computers & Operations Research*, 27(7), 709–723.

Miettinen, K., Mäkelä, M.M. (2002). On scalarizing functions in multiobjective optimization. *OR Spectrum*, 24(2), 193–213.

Miettinen, K., Mäkelä, M.M. (2006). Synchronous approach in interactive multiobjective optimization. *European Journal of Operational Research*, 170(3), 909–922.

Miettinen, K., Mustajoki, J., Stewart, T.J. (2014). Interactive multiobjective optimization with NIMBUS for decision making under uncertainty. *OR Spectrum*, 36(1), 39–56.

Mohammadi, A., Omidvar, M.N., Li, X. (2013). A new performance metric for user-preference based multi-objective evolutionary algorithms. In: *2013 IEEE Congress on Evolutionary Computation, CEC*, pp. 2825–2832.

Molina, J., Santana, L.V., Hernández-Díaz, A.G., Coello Coello, C.A., Caballero, R. (2009). g-Dominance: reference point based dominance for multiobjective metaheuristics. *European Journal of Operational Research*, 197(2), 685–692.

Ojalehto, V., Miettinen, K., Mäkelä, M.M. (2007). Interactive software for multiobjective optimization: IND-NIMBUS. *WSEAS Transactions on Computers*, 6(1), 87–94.

Petkus, T., Filatovas, E., Kurasova, O. (2009). Investigation of human factors while solving multiple criteria optimization problems in computer network. *Technological and Economic Development of Economy*, 15(3), 464–479.

Purshouse, R.C., Deb, K., Mansor, M.M., 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, A.B., Saborido, R., Luque, M. (2014). A preference-based evolutionary algorithm for multiobjective optimization: the weighting achievement scalarizing function genetic algorithm. *Journal of Global Optimization*, 1–29.

Schott, J.R. (1995). *Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization*. Technical report, DTIC Document.

Siegmund, F., Ng, A.H.C., 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., Ruiz, A.B., Miettinen, K. (2011). A preference based interactive evolutionary algorithm for multiobjective optimization: PIE. In: *Evolutionary Multi-criterion Optimization*. Springer, Berlin, pp. 212–225.

Sinha, A., Korhonen, P., Wallenius, J., Deb, K. (2014). An interactive evolutionary multi-objective optimization algorithm with a limited number of decision maker calls. *European Journal of Operational Research*, 233(3), 674–688.

Talbi, E.-G. (2009). *Metaheuristics: from Design to Implementation*, Vol., 74. John Wiley & Sons, New York.

Thiele, L., Miettinen, K., Korhonen, P.J., Molina, J. (2009). A preference-based evolutionary algorithm for multiobjective optimization. *Evolutionary Computation*, 17(3), 411–436.

Van Veldhuizen, D.A. (1999). *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. Technical report, DTIC Document.

Vira, C., Haimes, Y.Y. (1983). *Multiobjective Decision Making: Theory and Methodology*, Vol., 8. North-Holland, New York.

Wickramasinghe, U.K., Carrese, R., Li, X. (2010). Designing airfoils using a reference point based evolutionary many-objective particle swarm optimization algorithm. In: *2010 IEEE Congress on Evolutionary Computation, CEC*, pp. 1–8.

Wierzbicki, A.P. (1980). The use of reference objectives in multiobjective optimization. In: *Multiple Criteria Decision Making Theory and Application*. Springer, Berlin, pp. 468–486.

Zeng, S., Baležentis, T., Chen, J., Luo, G. (2013). A projection method for multiple attribute group decision making with intuitionistic fuzzy information. *Informatica*, 24(30), pp. 485–503.

Zhang, Q., Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6), 712–731.

Zhang, Q., Liu, W., Li, H. (2009). The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances. In: *2009 IEEE Congress on Evolutionary Computation, CEC*, pp. 203–208.

Zhou, A., Qu, B.-Y., Li, H. Zhao, S.-Z., Suganthan, P.N., Zhang, Q. (2011). Multiobjective evolutionary algorithms: a survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1), 32–49.

Zitzler, E., Künzli, S. (2004). Indicator-based selection in multiobjective search. In: *Parallel Problem Solving from Nature-PPSN VIII*. Springer, Berlin, pp. 332–842.

Zitzler, E., Thiele, L. (1998). Multiobjective optimization using evolutionary algorithms – a comparative case study. In: *Parallel Problem Solving from Nature – PPSN V*. Springer, Berlin, pp. 292–301.

Zitzler, E., Laumanns, M., Thiele, L. (2001). *SPEA2: Improving the Strength Pareto Evolutionary Algorithm*, Technical report.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., Da Fonseca, V.G. (2003). Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2), 117–132.

Zou, W., Zhu, Y., Chen, H., Zhang, B. (2003). Solving multiobjective optimization problems using artificial bee colony algorithm. *Discrete Dynamics in Nature and Society*, 2011

E. Filatovas received the PhD in informatics engineering from the Vilnius University in 2012, Lithuania. He is currently a post-doctoral researcher with the System Analysis Department at Institute of Mathematics and Informatics of Vilnius University, and a lecturer at Faculty of Fundamental Sciences of Vilnius Gediminas Technical University. His main research interests include multi-objective optimization, multi-objective evolutionary algorithms, multiple criteria decision making, parallel computing, data visualization, and image processing.

O. Kurasova received the PhD in computer science from Institute of Mathematics and Informatics jointly with Vytautas Magnus University in 2005, Lithuania. Recent employment is at Institute of Mathematics and Informatics of Vilnius University, as senior researcher, and at the Informatics Department of Lithuanian University of Educational Sciences as associate professor. Her research interests include data mining methods, optimization theory and applications, artificial intelligence, neural networks, visualization of multidimensional data, multiple criteria decision making, multi-objective evolutionary algorithms parallel computing.

K. Sindhya received the BE degree in chemical engineering from the R.V. College of Engineering, Bangalore, India, the ME degree in chemical engineering from the Birla Institute of Technology and Science (BITS), Pilani, India, and the PhD degree in mathematical information technology from the University of Jyväskylä, Finland, in 2011. He is currently a post-doctoral researcher with the Department of Mathematical Information Technology, University of Jyväskylä. He was previously a researcher with the Kanpur Genetic Algorithms Laboratory, Indian Institute of Technology Kanpur, India, and with the Aalto University School of Business (previously Helsinki School of Economics), Helsinki, Finland. His current research interests include single and multi-objective evolutionary algorithms, learning in multi-objective optimization, multiple criteria decision making, and applications to engineering design problems.

Pirmenybe grįstas evoliucinis algoritmas daugiakriteriniam optimizavimui – sinchroninis R-NSGA-II

Ernestas FILATOVAS, Olga KURASOVA, Karthik SINDHYA

Klasikinių evoliucinių daugiakriterinių algoritmų tikslas – rasti visos Pareto optimalių sprendinių aibės aproksimaciją. Evoliuciniuose daugiakriteriniuose optimizavimo algoritmuose sprendinių paiešką galima sukoncentruoti tik į tas Pareto aibės dalis, kurios tenkina sprendimų priemėjo teikiamą pirmenybę. Šiame straipsnyje pasiūlytas pirmenybe grįsto evoliucinio algoritmo praplėtimas daugiakriteriniams optimizavimo uždaviniams spręsti, pasiremiant interaktyvaus sinchroninio NIMBUS metodo įdėja ir ją pritaikant R-NSGA-II algoritmui. Pasiūlytame sinchroniniame R-NSGA-II algoritme naudojama informacija apie sprendimų priėmėjo teikiamą pirmenybę, siekiant rasti tik jį tenkinančius Pareto aibės sprendinius. Sinchroniškai naudojant kelias skaliarizacijas funkcijas, iš tos pačios pirmenybės informacijos randamos kelios sprendinių aibės. Šiame straipsnyje atliktas pasiūlyto sinchroninio R-NSGA-II ir originalaus R-NSGA-II algoritmų eksperimentinis lyginamasis tyrimas. Gauti rezultatai parodė, kad pasiūlytas algoritmas aproksimuoja Pareto aibę geriau nei originalus R-NSGA-II algoritmas, o sprendinių pasiskirstymo tolygumas yra pakankamai geras. Pagrindinis pasiūlyto algoritmo privalumas lyginant su originaliuoju yra tas, kad sinchroniškai gaunamos kelios nedominuojančių sprendinių grupės, atitinkančios skirtingus daugiakriterinio uždavinio skaliarizavimo būdus, esant tai pačiai sprendimų priemėjo teikiamos pirmenybės informacijai.