

Towards Automatic Testing of Reference Point Based Interactive Methods

Vesa Ojalehto^{1(✉)}, Dmitry Podkopaev², and Kaisa Miettinen¹

¹ University of Jyväskylä, Department of Mathematical Information Technology,
P.O. Box 35 (Agora), FI-40014 University of Jyväskylä, Finland
`{vesa.ojalehto,kaisa.miettinen}@jyu.fi`

² Systems Research Institute, Polish Academy of Sciences,
Newelska 6, 01-447 Warsaw, Poland
`dmitry.podkopaev@ibspan.waw.pl`

Abstract. In order to understand strengths and weaknesses of optimization algorithms, it is important to have access to different types of test problems, well defined performance indicators and analysis tools. Such tools are widely available for testing evolutionary multiobjective optimization algorithms.

To our knowledge, there do not exist tools for analyzing the performance of interactive multiobjective optimization methods based on the reference point approach to communicating preference information. The main barrier to such tools is the involvement of human decision makers into interactive solution processes, which makes the performance of interactive methods dependent on the performance of humans using them. In this research, we aim towards a testing framework where the human decision maker is replaced with an artificial one and which allows to repetitively test interactive methods in a controlled environment.

Keywords: Multiobjective optimization · EMO · Testing framework · Decision maker's preferences · Preference information · Aspiration level

1 Introduction

Many real-life problems of decision making and support are tackled by multiobjective optimization. A solution of a multiobjective optimization problem can be defined as a feasible solution which is the most preferred for a decision maker (DM). Therefore, multiobjective optimization methods that aim at supporting a DM rely on information about the DM's preferences (*preference information* for short) and incorporate mechanisms of communication with the DM. In the methods where such communication is organized in an interactive way (i.e. *interactive methods*), the solution process is carried out in iterations. In each iteration, the DM provides preference information and, as feedback, obtains information about Pareto optimal solutions derived based on this preference information [2, 8, 9]. Interactive methods are very suitable for solving practical problems due

to several advantages [2]. First, the DM gets the possibility to learn progressively about the set of Pareto optimal solutions of a complex problem, which reduces cognitive load. Secondly, applying interactive methods does not necessitate generating many Pareto optimal solutions, which is essential in the case of computationally complex problems. Instead, only solutions that are interesting for the DM are generated.

Many interactive methods have been developed so far, see e.g. [2,8,9]. Naturally, the problem of testing and comparing different methods arises [7,8]. Making tests and comparisons of interactive multiobjective optimization methods is hampered by the necessity of involving DMs in tests. First of all, this involvement makes method testing much more costly than testing by computational means, taking into account that many problems of industry, management, engineering, etc. require DMs being experts in corresponding fields. Secondly, it is hard to conduct good quality experiments due to various barriers related to human nature: the difficulty of creating proper motivation of DMs if tested e.g. by students using artificial problems; the inconsistency of human nature and variability among humans; difficulties of accounting for improving DM's capabilities in time due to learning¹.

As noted in [7,8], only few interactive multiobjective optimization methods have been extensively tested, which means that information about the quality of most of the methods cannot be called reliable. The main sources of such information are intuitive conclusions of the authors of the methods and results of employing the methods for solving limited numbers of real-world or hypothetical problems. In order to overcome the deficiency of tests and comparisons of interactive methods, one can use artificial DMs understood as techniques of generating preference information. Because interactive methods vary significantly in approaches to preference information modeling [9,10], different artificial DMs should be created for different preference information types.

Compared to the diversity of interactive methods, the number of approaches to creating artificial DMs is very limited. In [8], some examples of testing methods by using artificial DMs were described. Since 1999, only few new works have appeared where actions of DMs have been simulated using artificial mechanisms. Among them, a DM was represented as an additive value function in [13], and that representation was used for generating goals in a simulated goal programming problem with a discrete number of alternatives. When generating goals, judgment errors and biases of the DM were simulated and then effects on the performance of goal programming algorithms were studied. In [15], a universal mechanism of generating DM's preference information was proposed based on minimizing the distance of the corresponding Pareto optimal solution to a given "goal solution". However, that mechanism has a limited application area. The work in [7] aimed at the same goal as our research, except that a DM was modelled via a value function, which does not allow generating reference point, but

¹ Humans learn, therefore, it is not easy to employ the same DMs to test different methods, as they have learnt about the problem while solving the problem, which affects the quality of a long series of experiments.

provides preference information as rankings of given sets of alternatives. Some mechanisms of modeling the imperfection of humans' judgments was incorporated into that model and used for testing the BC-EMOA algorithm [7].

More approaches to creating artificial DMs have been developed for enhancing existing methods, and they may be adoptable for method testing. The approach in [1] is an example of such a study (see also references therein), where the DM's preference model based on a fuzzy inference system was trained during the interactive solution process and used for providing additional preference information on behalf of the DM.

Clearly, each artificial DM created for testing methods should be tailored to the preference information expected by these methods. A popular way of modeling DM's preferences is via value functions (often referred to as utility functions). The advantages are a theoretically proved completeness [5], and the simplicity of representation. From a value function, one can easily obtain such preference information as pairwise comparisons or rankings of given sets of alternatives (as e.g. in [7]). Note that methods where the DM can be replaced by a value function are called non ad hoc methods [8, 12, 13]. However, in many interactive methods which are popular in practice, the preference information is provided in the form of reference points [2, 9] representing desirable objective function values. Such methods are regarded as ad hoc, e.g. methods where the DM cannot be replaced by a value function [8, 12, 13]. To our knowledge, there are no artificial DMs developed for testing methods based on reference points.

In this paper, we develop an artificial DM for testing interactive methods, which involve preference information as a reference point. It is the first development of this kind. We mimic the behavior of a human DM who adjusts preferences based on obtained information about derived solutions, and demonstrates randomness in the behavior in responses to the uncertainty about the Pareto optimal set.

The paper is organized as follows. In Sect. 2, we describe the concept of an artificial DM and in Sect. 3 incorporate it into a framework for testing interactive methods. In Sect. 4, we present results of testing two methods: R-NSGA-II [4] and minimizing an achievement scalarizing function [14]. We conclude in Sect. 5.

2 Artificial Decision Maker

We propose to employ an artificial DM to replace the real DM. Our concept of an artificial DM and its interaction with an interactive method comprises the following three components:

- *Steady part*: the complexity of knowledge possessed by the DM and related to solving the considered class of problems which does not change during the solution process. This includes accumulated experience and the core preferences which do not change in time.
- *Current context*: the current situation as perceived by the DM, which may change in time. This includes: the knowledge about the problem accumulated

by the DM during the solution process, level of tiredness which can affect concentration, and the probability of making mistakes.

- *Preference information*: the method-specific information expressed by the DM during the solution process to guide the method toward solutions that are more preferred by the DM.

The artificial DM should be defined by the steady part which does not change in time, a mechanism of representing and updating the current context as the solution process continues, and the mechanism of generating the preference information based on the steady part and the current context. By varying the parameters of the steady part, one can obtain different artificial DMs for conducting multiple experiments.

It is tempting to describe the steady part as a classical model of DM's preferences (e.g. choice function, binary relation or utility function). However, as said, there are no studies describing how to generate preference information in terms of reference points from such models. Therefore, we construct the steady part in the form of some general preference information which cannot be called a preference model in the classical sense. We propose a procedure of generating the current preference information based on the steady preference information and taking into account the current context. The latter is represented by the current solution or the set of derived solutions available for the DM.

3 Testing Framework

The aim of this research is to create a framework for comparing different interactive methods with an artificial DM. The proposed framework is compatible with interactive methods where the DM provides one's preferences in each iteration of the method as a reference point. In what follows, we first give basic notions of multiobjective optimization, then describe the artificial DM used in the framework, and finally proceed with details on how the artificial DM is utilized.

Multiobjective optimization problems are formulated as follows:

$$\begin{aligned} & \text{minimize} && \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_k(\mathbf{x}))^T \\ & \text{subject to} && \mathbf{x} = (x_1, \dots, x_n)^T \in S, \end{aligned}$$

meaning that the DM wishes to simultaneously minimize k ($k \geq 2$) objective functions $f_i : S \rightarrow \mathbb{R}$ on the set S of *feasible solutions* (*decision vectors*) which is a nonempty compact subset of \mathbb{R}^n . The image of S is denoted by $\mathbf{f}(S)$. Its elements $\mathbf{z} = \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_k(\mathbf{x}))^T$ in the objective space \mathbb{R}^k consisting of objective (function) values are called *objective vectors*.

The set of Pareto optimal solutions of the problem (the *Pareto optimal set*) is defined by $E = \{\mathbf{x} \in S : \text{there is no } \mathbf{x}' \in S \text{ such that } f_i(\mathbf{x}') \leq f_i(\mathbf{x}) \text{ for all } i = 1, \dots, k \text{ and } \mathbf{f}(\mathbf{x}') \neq \mathbf{f}(\mathbf{x})\}$.

Let us also introduce an *ideal objective vector* and a *nadir objective vector* defined, respectively, as $\mathbf{z}^* = (z_1^*, \dots, z_k^*)^T$ where $z_i^* = \min_{\mathbf{x} \in E} f_i(\mathbf{x})$ for $i = 1, \dots, k$, and $\mathbf{z}^{\text{nad}} = (z_1^{\text{nad}}, \dots, z_k^{\text{nad}})^T$ where $z_i^{\text{nad}} = \max_{\mathbf{x} \in E} f_i(\mathbf{x})$ for

$i = 1, \dots, k$. Note that the nadir objective vector is, in general, more difficult to obtain than the ideal objective vector and, therefore, approximations are often used (see e.g. [3, 8] and references therein).

The *current context* is defined as follows. We assume that the artificial DM is aware of the bounds of the objective functions, that is, the objective vectors \mathbf{z}^* and \mathbf{z}^{nad} . These vectors can either be known or estimated [8]. Their components give bounds on the aspiration levels which constitute a reference point. In addition, the set of derived Pareto optimal (or non-dominated) solutions, which is updated during the solution process, provides the DM with information about what combinations of objective function values are achievable.

As for the *steady part* of preference representation, for each objective f_i , $i = 1, \dots, k$, we introduce a “ranking coefficient” w_i which determines the priority of the objective function f_i over the other objective functions. That is, the DM prefers more to obtain smaller values for those objective functions whose $w_i \in (0, 1]$ is larger. We assume that all objective functions are relevant to the problem and, therefore, each of them should have a ranking coefficient.

In addition to ranking coefficients, we utilize *initial aspiration levels*, $\text{asp}_i \in (z_i^*, z_i^{\text{nad}}]$, $i = 1, \dots, k$, that is, objective values the artificial DM would like to achieve. If $\text{asp}_i = z_i^{\text{nad}}$, we assume that the artificial DM initially does not have any preferences regarding objective f_i .

The probability $p \in (0, 1]$ determines how willing the artificial DM is to give up on the initial preferences. With larger p and larger w_i values, the artificial DM is more probable to consider the f_i objective relevant, i.e., to use asp_i as the reference point component (otherwise, it uses the component of the nadir vector). Alongside with the constant probability p , we introduce the varying probability p_λ which is initialized with p and decreased in the process of consecutive consideration of objective functions in the order defined by their priority (for details, see the scheme of the decision process below). Finally, the *preference information* is represented as a reference point $\text{ref} = (\text{ref}_1, \dots, \text{ref}_k)$.

Now we can describe how the artificial DM interplays with a method. This process has the following parameters: θ – tolerance value which controls when an objective function value is considered to be acceptable; t_{\max} – maximum number of iterations; t – iteration counter. Furthermore, we denote a uniformly distributed random number in the interval $[0, 1]$ by rand .

At the beginning of the solution process, when the set of derived solutions P is empty, we generate preference information as described below.

1. For each objective function f_i :
 - (a) if asp_i is not defined, set $\text{asp}_i = z_i^*$;
 - (b) Set initial components of the reference point ref_i :
 - i if $\text{rand} < p \cdot w_i$, set $\text{ref}_i = \text{asp}_i$,
 - ii else, set $\text{ref}_i = z_i^{\text{nad}}$.

Here, if the aspiration level for an objective function is not defined, we set it to the ideal value of this objective in step (a), as each objective should have the opportunity to be improved, even if the artificial DM does not have a notion

of what the values should be. Otherwise, the component of the initial reference point for each objective function is set at step (b) either as the aspiration level or the component of the nadir objective vector. The latter choice depends on the ranking coefficient of the objective function as well as the probability, if the aspiration level of the objective function should be used, in order to increase the priority of improving values of those objective functions.

After the artificial DM has been initialized with preference information, it is utilized with an interactive multiobjective optimization method as follows.

1. Set
 - $p_\lambda = p$ – the varying probability of using the aspiration level as the component of the reference point,
 - $F = \emptyset$ – the current index set of relevant objective functions,
 - $t = 0$ – the iteration counter,
 - P – the set of derived Pareto optimal (non-dominated) solutions,
 - and generate initial preferences (reference point ref) as described above.
2. $t++$
3. Provide the interactive method with the current reference point ref to generate a new Pareto optimal solution x , and add this solution to the set P .
4. For each objective function f_i ordered by ranking coefficients w_i in a decreasing order:
 - a. if $\text{asp}_i - f_i(x) < \theta$ and $\text{rand} < p$, add i to the set F ,
else, if $\text{rand} < p_\lambda$, add i to the set F .
 - b. Set $p_\lambda = p_\lambda - \frac{p_\lambda}{|F|} \cdot |F|$.
5. If $|F| = k$, go to step 10.
6. For each objective function f_i with $i \in F$:
 - a. Set the new component of the reference point ref_i :
 $\text{ref}_i = \text{asp}_i - (\text{asp}_i - f_i(x))/2$.
7. For each objective function f_l with $l \notin F$:
 - a. Construct predict_l using a decision tree trained with previously obtained Pareto optimal solutions;
 - b. set $\text{ref}_l = \min(\text{predict}_l, z_l^{\text{nad}})$.
8. If the new reference point is identical to the previous one, go to step 10.
9. If $t < t_{\max}$ go to step 2.
10. **STOP**. Select x as the solution to the problem.

In the beginning of the solution process, the interactive method is used to generate a new objective vector using the current reference point ref . Then we select and add to set F those objective functions which are considered to be relevant during this iteration. Firstly, if the objective vector value has achieved the desired aspiration level, it is selected with a high probability (but not equal to one), as we are assuming that the DM is not certain that the aspiration level is the best possible which could be achieved.

On the other hand, if the component of the objective vector has not achieved the desirable value, it is selected with a lower probability. This selection probability is then decreased based on the number of objective functions selected so

far. The described scheme gives a strong preference on selecting most preferred objectives that have achieved desired values, while decreasing chances to select less desired objective functions. When all objective functions are selected to be relevant, the current Pareto optimal solution is considered to be the final solution of the problem. This means that it is possible that the artificial DM will end the solution process prematurely.

Next, in step 4, a new reference point ref is created. In order to take into account that the aspiration level might not be reachable, the reference point components for the selected objectives are set between the current objective function value and the aspiration level. Then the remaining reference point components are set either to the nadir value or to the value predicted by a decision tree [11] trained with previously obtained Pareto optimal solutions. A decision tree is built for each objective, using the values of the other objectives as a training data for predicting which values should be selected for other objectives in order to obtain the preferred value for the considered objective function.

The solution is finally accepted either when all initial aspiration levels have been achieved, artificial DM could not create a new reference point or after the maximum number of iterations has been conducted.

4 Numerical Experiments

Next we give some computational results to demonstrate application of the artificial DM. For this demonstration, we use two different methods for generating new Pareto optimal solutions: R-NSGA-II algorithm [4] and minimizing the achievement scalarizing function (ASF) of a reference point method [14] to project a reference point to the Pareto optimal set, where the differential evolution algorithm is used to minimize the ASF. As the latter method produces only a single Pareto optimal solution, while R-NSGA-II produces several ones, among the Pareto optimal (nondominated) solutions generated by R-NSGA-II, the one nearest to the reference point is selected. The R-NSGA-II algorithm had the population size 100 and was allowed to have maximum of 200 generations totaling to maximum of 20000 evaluations. The differential evolution method had the stopping criterion of maximum of 20000 evaluations.

Each method was used to solve four different problems: DTLZ1 – DTLZ4 [6] with the number of objectives (k) ranging from 2 to 6, totaling 24 different problems. Each problem was solved ten times using both methods, with ten different, randomly generated sets of initial preference information. The maximum number of iterations was set to 11.

Examples of two test runs when solving the ZDLT2 problem with three objective functions with both R-NSGA-II and the ASF methods can be seen in Figs. 1 and 2, respectively. In these figures, the search path taken by the artificial DM is shown as a continuous line, with x marking as each reference point constructed by the artificial DM. The diamond represents the initial reference point and the square represents the final reference point of the solution process. In Fig. 1, it can be seen that the artificial DM constructed seven reference points in the case of

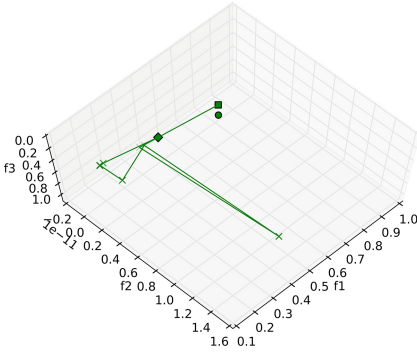


Fig. 1. R-NSGA-II: search path with the artificial DM

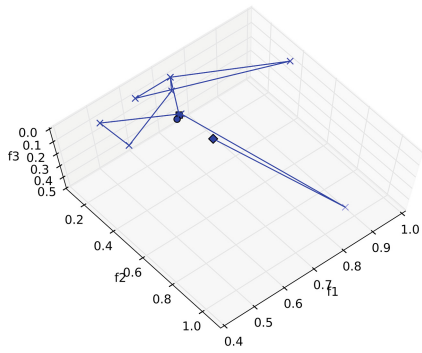


Fig. 2. ASF method: search path with the artificial DM

R-NSGA-II and nine for the ASF method (Fig. 2). The circle outside the search path shows the aspiration level, asp_i , $i = 1, \dots, k$, that the artificial DM was aiming at. The Pareto optimal solutions generated during the test runs are not shown here.

At the beginning of each test run, the artificial DM does not have any knowledge of the Pareto optimal set of the problem being solved and as described in Sect. 3, the first steps taken are random. This means that all search paths are different for each test run, and the first steps can lead away from the aspiration levels. But as can be seen in figures, after the artificial DM has accumulated enough knowledge of the problem, the solution process converges towards the aspiration levels.

The obtained results are detailed in Table 1 for R-NSGA-II and in Table 2 for the ASF method, where for brevity we limit to the cases $k \in \{2, 4, 6\}$. In the tables, we give the name of the problem, the number of objectives (k), the mean and minimum distances to the initial aspiration levels and the standard deviation of the distances. Finally, the tables report how many iterations the artificial DM used the on average during the test runs.

As can be seen, a distinction between the two methods can be drawn, while both methods behave in a somewhat similar manner. The ASF method could find final solutions that are consistently closer to the initial preferences, i.e., mean values and deviations of distances are smaller than with R-NSGA-II. The ASF method was also able to achieve the initial aspiration levels, even though this did not happen in all runs. For problems with four and more objectives, the differences in the performance were slightly smaller, as the ASF method did not achieve initial aspiration levels consistently, but it should be noted that the performance of R-NSGA-II also deteriorated.

The latter result can be considered as somewhat surprising, taking into account that the NSGA-II algorithm underlying the R-NSGA-II algorithm does not typically perform well with problems having more than three objective functions, and it could be expected that the performance of R-NSGA-II would

Table 1. R-NSGA-II results

Problem	k	mean	dev	min	iter
DTLZ1	2	0.709	0.447	0.000	10
DTLZ2	2	0.040	0.044	0.002	11
DTLZ3	2	2.427	1.600	0.863	5
DTLZ4	2	0.237	0.278	0.006	9
DTLZ1	4	0.224	0.185	0.012	11
DTLZ2	4	0.273	0.185	0.061	11
DTLZ3	4	3.579	0.954	1.671	5
DTLZ4	4	0.439	0.355	0.038	11
DTLZ1	6	0.432	0.270	0.148	11
DTLZ2	6	0.365	0.192	0.024	11
DTLZ3	6	6.000	2.238	3.865	11
DTLZ4	6	0.411	0.269	0.026	11

Table 2. ASF method results

Problem	k	mean	dev	min	iter
DTLZ1	2	0.606	0.327	0.000	10
DTLZ2	2	0.002	0.002	0.000	11
DTLZ3	2	2.112	1.870	0.000	4
DTLZ4	2	0.002	0.002	0.000	11
DTLZ1	4	0.239	0.257	0.000	11
DTLZ2	4	0.004	0.002	0.001	11
DTLZ3	4	1.819	1.340	0.000	6
DTLZ4	4	0.069	0.147	0.001	11
DTLZ1	6	0.386	0.247	0.001	10
DTLZ2	6	0.102	0.207	0.001	10
DTLZ3	6	1.725	1.287	0.005	8
DTLZ4	6	0.115	0.155	0.001	11

deteriorate more. However, it should be noted that the aim of the interactive solution processes is not to obtain best possible coverage of the Pareto frontier, but to concentrate on the area which is the most interesting for the DM. As R-NSGA-II generates several solutions in that area in contrast to a single solution obtained by the ASF method, the former algorithm provides the artificial DM more information, i.e., Pareto optimal solutions to construct more suitable reference points. This implies that the comparison of population based and non-population based methods should be made fair by paying attention to the amount of information the artificial DM is trained with.

5 Conclusions

In this research, we proposed to build an automated framework for testing interactive multiobjective optimization methods, without utilizing a value function to represent the DM's preferences. This was achieved by replacing the human DM with an artificial DM constructed from two distinct parts: the steady part and the current context. With the steady part the artificial DM tries to maintain the search towards its preferences, while at the same time the current context allows changing the direction as well as ending the solution process prematurely, mimicking actions of a human DM. With the proposed framework, it is possible to carry out repeatable tests of interactive methods in a controlled environment.

The numerical experiments performed with the proposed testing framework indicate that the algorithm can identify differences between different interactive methods. In the experiments, two interactive methods were compared solely based on the distance between the final solution and the steady preference information. In addition to this distance, it would be interesting to construct new performance metrics specific for interactive methods, such as characteristics of the trajectory of the solution process in the objective space.

Acknowledgments. This work was supported on the part of Vesa Ojalehto by the Academy of Finland (grant number 287496).

References

1. Babbar-Sebens, M., Minsker, B.S.: Interactive genetic algorithm with mixed initiative interaction for multi-criteria ground water monitoring design. *Appl. Soft Comput.* **12**(1), 182–195 (2012)
2. Branke, J., Deb, K., Miettinen, K., Słowiński, R. (eds.): *Multiobjective Optimization: Interactive and Evolutionary Approaches*. Springer, Heidelberg (2008)
3. Deb, K., Miettinen, K., Chaudhuri, S.: Towards an estimation of nadir objective vector using a hybrid of evolutionary and local search approaches. *IEEE Trans. Evol. Comput.* **14**(6), 821–841 (2010)
4. Deb, K., Sundar, J., Udaya Bhaskara Rao, N., Chaudhuri, S.: Reference point based multi-objective optimization using evolutionary algorithms. *Int. J. Comput. Intell. Res.* **2**(3), 273–286 (2006)
5. Debreu, G.: *Theory of Value: An Axiomatic Analysis of Economic Equilibrium*. Cowles Foundation for Research in Economics at Yale University, New Haven (1959). Monograph 17
6. Huband, S., Hingston, P., Barone, L., While, L.: A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Trans. Evol. Comput.* **10**(5), 477–506 (2006)
7. López-Ibáñez, M., Knowles, J.: Machine decision makers as a laboratory for interactive EMO. In: Gaspar-Cunha, A., Henggeler Antunes, C., Coello, C.C. (eds.) *EMO 2015*. LNCS, vol. 9019, pp. 295–309. Springer, Heidelberg (2015)
8. Miettinen, K.: *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston (1999)
9. Miettinen, K., Hakanen, J., Podkopaev, D.: Interactive nonlinear multiobjective optimization methods. In: Greco, S., Ehrgott, M., Figueira, J. (eds.) *Multiple Criteria Decision Analysis: State of the Art Surveys*, pp. 931–980. Springer, New York (2016)
10. Purshouse, R., Deb, K., Mansor, M., Mostaghim, S., Wang, R.: A review of hybrid evolutionary multiple criteria decision making methods. In: *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, pp. 1147–1154 (2014)
11. Quinlan, J.R.: Induction of decision trees. *Mach. Learn.* **1**(1), 81–106 (1986)
12. Steuer, R.E.: *Multiple Criteria Optimization: Theory, Computation, and Application*. Wiley, New York (1986)
13. Stewart, T.J.: Goal programming and cognitive biases in decision-making. *J. Oper. Res. Soc.* **56**(10), 1166–1175 (2005)
14. Wierzbicki, A.: A mathematical basis for satisficing decision making. *Math. Model.* **3**, 391–405 (1982)
15. Zujevs, A., Eiduks, J.: New decision maker model for multiobjective optimization interactive methods. In: *Proceedings of the Information Technologies*, pp. 51–58. Kaunas: Technologija (2011)