

VILNIUS UNIVERSITY

Ernestas Filatovas

SOLVING MULTIPLE CRITERIA OPTIMIZATION
PROBLEMS IN AN INTERACTIVE WAY

Summary of Doctoral Dissertation

Technological Sciences, Informatics Engineering (07T)

Vilnius, 2012

Doctoral dissertation was prepared at the Institute of Mathematics and Informatics of Vilnius University in 2006–2011.

Scientific Supervisor

Dr Olga Kurasova (Institute of Mathematics and Informatics of Vilnius University, Technological Sciences, Informatics Engineering – 07T).

The dissertation will be defended at the Council of the Scientific Field of Informatics Engineering at the Institute of Mathematics and Informatics of Vilnius University:

Chairman

Prof Dr Habil Gintautas Dzemyda (Vilnius University, Technological Sciences, Informatics Engineering – 07T).

Members:

Prof Dr Eduardas Bareiša (Kaunas University of Technology, Technological Sciences, Informatics Engineering – 07T),

Prof Dr Romas Baronas (Vilnius University, Physical Sciences, Informatics – 09P),

Prof Dr Habil Antanas Čenys (Vilnius Gediminas Technical University, Technological Sciences, Informatics Engineering – 07T),

Prof Dr Saulius Gudas (Vilnius University, Technological Sciences, Informatics Engineering – 07T).

Opponents:

Prof Dr Habil Jonas Mockus (Vilnius University, Technological Sciences, Informatics Engineering – 07T),

Prof Dr Dalius Navakauskas (Vilnius Gediminas Technical University, Technological Sciences, Informatics Engineering – 07T).

The dissertation will be defended at the public meeting of the Council of the Scientific of Informatics Engineering in the auditorium number 203 at the Institute of Mathematics and Informatics of Vilnius University, at 1 p. m. on March 28 2012.

Address: Akademijos st. 4, LT-08663 Vilnius, Lithuania.

The summary of the doctoral dissertation was distributed on February 27 2012.

A copy of the doctoral dissertation is available for review at the Library of Vilnius University.

VILNIAUS UNIVERSITETAS

Ernestas Filatovas

DAUGIAKRITERINIŲ OPTIMIZAVIMO UŽDAVINIŲ
SPRENDIMAS INTERAKTYVIUOJU BŪDU

Daktaro disertacijos santrauka

Technologijos mokslai, informatikos inžinerija (07T)

Vilnius, 2012

Disertacija rengta 2006–2011 metais Vilniaus universiteto Matematikos ir informatikos institute.

Mokslinė vadovė

Dr. Olga Kurasova (Vilniaus universiteto Matematikos ir informatikos institutas, technologijos mokslai, informatikos inžinerija – 07T).

Disertacija ginama Vilniaus universiteto Matematikos ir informatikos instituto Informatikos inžinerijos mokslo krypties taryboje:

Pirmininkas

prof. habil. dr. Gintautas Dzemyda (Vilniaus universitetas, technologijos mokslai, informatikos inžinerija – 07T),

Nariai:

prof. dr. Eduardas Bareiša (Kauno technologijos universitetas, technologijos mokslai, informatikos inžinerija – 07T),

prof. dr. Romas Baronas (Vilniaus universitetas, fiziniai mokslai, informatika – 09P),

prof. habil. dr. Antanas Čenys (Vilniaus Gedimino technikos universitetas, technologijos mokslai, informatikos inžinerija – 07T),

prof. dr. Saulius Gudas (Vilniaus universitetas, technologijos mokslai, informatikos inžinerija – 07T).

Oponentai:

prof. habil. dr. Jonas Mockus (Vilniaus universitetas, technologijos mokslai, informatikos inžinerija – 07T),

prof. dr. Dalius Navakauskas (Vilniaus Gedimino technikos universitetas, technologijos mokslai, informatikos inžinerija – 07T).

Disertacija bus ginama viešame Informatikos inžinerijos mokslo krypties tarybos posėdyje 2012 m. kovo mėn. 28 d. 13 val. Vilniaus universiteto, Matematikos ir informatikos instituto 203 auditorijoje.

Adresas: Akademijos g. 4, LT-08663 Vilnius, Lietuva.

Disertacijos santrauka išsiuntinėta 2012 m. vasario 27 d.

Disertaciją galima peržiūrėti Vilniaus universiteto bibliotekoje.

1. Introduction

The relevance of the problems, the scientific novelty of the results and their practical significance as well as the aim and tasks of the work are described in this chapter.

Relevance of the Problem

In practice, optimization problems are often multiple criteria. Such problems are solved in many areas of human activities: process management, economics, aircraft construction, bridge construction, and others. We often need to choose a compromise between two or more criteria, and the criteria are usually contradictory – with a reduction in the value of one criterion, the value of the other criterion is increasing. For example, in order to increase profit, it is necessary to reduce expenses; raising the car power, it is necessary to reduce fuel consumption. In everyday life, multiple criteria problems are solved intuitively. However, these problems are successfully solved only using mathematical methods. Multiple criteria problems can have several criteria and all the criteria must be harmonized in a decision-making process. Often there is a case where these problems do not have a single optimal solution. Then only Pareto optimal solutions, also called non-dominant, are searched.

Many different Pareto optimal solutions are obtained while solving multiple criteria optimization problems. The final decision depends on a decision maker, who participates in the decision-making process. One of the simplest ways to solve a multiple criteria optimization problem is based on its transformation to a single criterion problem using scalarization methods. The well-known weighted sum method is frequently used. A single criterion objective function is made by summing up the objective functions of all the criteria, multiplied by weighting coefficients. When the problem is solved in an interactive way, the decision maker selects the values of the weighting coefficients interactively, with regard to specifics of the problem, to the available experience and expertise. This enables the decision maker to participate not only in decision-making, but also in the decision process. Moreover, these single criterion problems can be solved independently and the decision process can be accelerated using multi-processor computers or computer clusters. Solving strategies must be developed with a view to use computing resources efficiently, taking into account the abilities of the decision maker to form the tasks.

Disadvantages of the weighted sum method are as follows: the solutions are not uniformly distributed on the Pareto front, and if the objective functions are non-convex, not all Pareto optimal solutions are obtained. It is necessary to look for methods that do not have these disadvantages, but keep the interaction.

A decision support system is essential for solving multiple criteria optimization problems interactively. The system must ensure the convenience of use, speed, and visualization of intermediate and final solutions. The decision support system helps the decision maker to go deep into the problem and to understand its specificity. It is necessary to develop a new system that would integrate optimization methods, visualization and parallelization of the decision process.

Two main problems are solved here: (1) development and analysis of an interactive way for solving multiple criteria optimization problems which ensures a uniform distribution of solutions on the Pareto front; (2) implementation and analysis of an interactive decision support system for solving multiple criteria optimization problems.

The Aim and Tasks

The key aim of the dissertation is to develop an interactive way for solving multiple criteria optimization problems with the help of which a decision maker obtains alternative solutions uniformly distributed on the Pareto front, using the interactive decision support system.

To achieve the aim, it was necessary to solve the following tasks:

- to analyze multiple criteria optimization methods and interactive decision support systems for solving multiple criteria optimization problems;
- to create an interactive way for solving multicriteria optimization problems, which finds alternative solutions uniformly distributed on the Pareto front;
- to develop an interactive decision support system which integrates the created interactive solving way, the decision process visualization and parallelization for multiple criteria optimization;
- to develop and compare the solving strategies, when a multiple criteria optimization problem is solved interactively, using a computer cluster;
- to investigate the time required for a decision maker to learn to solve a multiple criteria optimization problem by the interactive decision support system.

The Objects of Research

The objects of research of the dissertation are multiple criteria optimization problems, interactive methods for solving these problems, interactive decision support systems, and application of parallel computing in decision support systems.

Scientific Novelty

1. The way proposed for solving multiple criteria optimization problems, which integrates a weighted sum method and an adaptive weighted sum method, allows us to solve multiple criteria optimization problems interactively and to find the solutions uniformly distributed on the Pareto front.
2. The strategies for solving multiple criteria optimization problems interactively using a computer cluster are compared experimentally.
3. The time required for a decision maker to learn to solve a multiple criteria optimization problem by the interactive decision support is investigated experimentally.

Practical Significance

The interactive decision support system is adapted to a multiple criteria optimization problem in selecting the optimal nutrition values. The system can be adapted to other similar problems, for example, diet formation, and menu formation for the pupils, etc.

The research was partly supported by the Lithuanian State Science and Studies Foundation project: “The research of human factors in multiple criteria optimization problems applying parallel computing (No. T-07134), 2007”.

The Defended Statements

1. The way proposed for solving multiple criteria optimization problems finds the solutions uniformly distributed on the Pareto front and allows solving multiple criteria optimization problems interactively.
2. The integration of the created interactive way for solving multiple criteria optimization problems, the decision process visualization and parallelization into the interactive decision support system ensures an effective decision process which helps a decision maker to find a preferable solution.
3. The solving strategy for solving multiple criteria optimization problems in which the computer helps a decision maker to form the tasks allows finding a preferable solution faster using more computers.
4. A decision maker learns to solve a multiple criteria optimization problem faster by the interactive decision support system when the computer cluster is used, but the number of computers should be such that a decision maker were able to evaluate the solutions obtained in time.

Approbation and Publications of the Research

The main results of the dissertation were published in 5 scientific papers: 4 articles in the periodical scientific publications; 1 article in the proceedings of scientific conference. The main results of the work have been presented and discussed at 8 national and international conferences.

The Scope of the Scientific Work

The dissertation is written in Lithuanian. It consists of 5 chapters and the list of references. There are 139 pages of the text, 29 figures, 5 tables, and 193 bibliographical sources.

2. Multiple Criteria Optimization and Decision Support Systems

In this chapter, the classification of the multiple criteria optimization (MO) methods is presented and the main methods are described. The methods for a uniform distribution of solutions on the Pareto front are investigated, too. The measures for estimating of the solutions are described. Decision support systems are reviewed.

The general form of a MO (minimization) problem is:

$$\min_{X \in \mathbf{D}} \mathbf{F}(X) = [f_1(X), f_2(X), \dots, f_m(X)]^T \quad (2.1)$$

subject to

$$g_j(X) \leq 0, \quad j = 1, 2, \dots, k,$$

$$h_l(X) = 0, \quad l = 1, 2, \dots, p,$$

where \mathbf{D} is a bounded domain in the n -dimensional Euclidean space R^n ; m is the number of criteria; k is the number of inequality constraints; p is the number of equality constraints; $X = (x_1, x_2, \dots, x_n)$ is a vector of variables; $\mathbf{F}(X) \in R^m$ is a vector of objective functions; and the functions $f_j(X): R^n \rightarrow R^1$ are criteria. The feasible criterion space (feasible region) \mathbf{Z} is defined as the set $\{\mathbf{F}(X) \mid X \in \mathbf{D}\}$.

A point $X^* \in \mathbf{D}$ is *Pareto optimal*, iff there does not exist another point $X \in \mathbf{D}$, such that $\mathbf{F}(X) \leq \mathbf{F}(X^*)$, and $f_i(X) < f_i(X^*)$ for at least one function. The set of all Pareto optimal points is called a *Pareto front*.

The MO methods can be classified into four classes according to the role of a decision maker (DM) in the decision process: no preference, posteriori, priori, and interactive methods. *No preference* methods do not require the priority of objectives. No information on DM's preferences is used. A particular method yields only one Pareto optimal solution, so a few Pareto optimal solutions can be obtained by using different no preference methods or different metrics. In the *posteriori* methods, many Pareto optimal solutions are generated and DM selects the most preferable one. Generation of many Pareto optimal solutions is computationally expensive. Moreover, it is difficult for DM to analyze a large amount of solutions. The DM must specify his/her preference information before the decision process in the *priori* methods. If the solution obtained is satisfactory, the DM does not have to waste too much time in the decision process. However, before the decision process, the DM does not necessarily know how realistic his expectations are. If DM does not get a satisfactory solution, he must change his preference information. The *interactive* methods require the interaction with the DM during the decision process. The decision process is iterative: the DM changes preference information during the decision process according to the solutions obtained previously. These methods allow us to solve complicated MO problems with a lot of criteria and constraints.

The most wide used MO method is the weighted sum (WS) method that transforms a multiple criteria optimization problem (MOP) into a single criteria optimization problem (SOP) by summing up all the criteria multiplied by weighting coefficients. The formulation of the SOP obtained is:

$$\min_{X \in \mathbf{D}} \sum_{i=1}^m w_i f_i(X), \quad (2.2)$$

where w_i is a weighting coefficient of the i -th criterion, $0 < w_i \leq 1$, $\sum_{i=1}^m w_i = 1$. The method finds all the Pareto optimal points if the functions $f_i(X)$ are convex. This method can be applied as a posteriori, priori, or interactive approach according to the way of DM's participation in the decision process.

The adaptive weighted sum (AWS) method belongs to a group of methods for uniform distribution of the solutions on the Pareto front. The method is based on the WS method and it is a posteriori as all the methods of this group. The AWS method searches the points of the Pareto front in the regions where the Pareto front points have not been discovered by the WS method, by introducing some additional constraints on these regions. The advantages of this method are that it produces well-distributed solutions, finds Pareto optimal solutions in non-convex regions. Moreover, the AWS method adapts to the shape of the Pareto front and ensures the uniform distribution of Pareto points. This feature distinguishes the AWS method from the other methods of this group. In the case of two criteria, the AWS method consists of the following procedures:

1. The objective functions of the criteria are normalized.
2. A MOP (2.1) is solved by the WS method and some points are obtained.

3. The Euclidean distances between the neighbouring points are computed. The coincident points are eliminated.
4. The Pareto front is divided into the intervals consisting of the neighbouring points. The lengths of the intervals are computed.
5. The intervals, in which the Pareto optimal points will be searched, are identified. The points will be searched in the interval, if it is longer than the defined maximal length and this interval has not been eliminated before. If there are no such intervals, the decision process is terminated.
6. The Pareto front points are searched in the intervals identified by the WS method with additional constraints (2.3), (2.4) introduced which restrict a feasible region in each identified interval. The constraints are:

$$\hat{f}_1(X) \leq P_1^x - \delta_1, \quad (2.3)$$

$$\hat{f}_2(X) \leq P_2^y - \delta_2, \quad (2.4)$$

where δ_1 and δ_2 are offset distances selected by the DM, P_1^x and P_2^y are x and y positions of the i -th endpoint in each interval; $\hat{f}_1(X)$ and $\hat{f}_2(X)$ are the normalized functions of criteria.

7. If no Pareto front points are found in the interval, it will be eliminated. Go to step 3.

In the case of more than two criteria, the principle of the method remains: the above-mentioned intervals must be changed to patches (hyperpatches) and the length of the interval to the area of patches (hypervolume of hyperpatches).

In the three or more criteria cases, the Pareto filter is required. Moreover, it is difficult to impose inequality constraints. So, the authors of the AWS method have adopted equality constraints to define feasible regions for further divisions. However, the equality constraints strongly restrict a feasible region, therefore it is difficult to find solutions which satisfy these constraints.

When solving MOPs it is important to estimate optimization methods and compare them. Various authors proposed some measures (generational distance, hypervolume, error ratio, spread, etc.). In this research, the *spread* is used in order to estimate the distribution of obtained points on the Pareto front. In the case of two criteria, the spread is computed by this formula:

$$\varphi = \sqrt{\frac{1}{n' - 1} \sum_{i=1}^{n'} (\bar{d} - d_i)^2}, \quad (2.5)$$

where $d_i = \min_j (|f_1^i(X) - f_1^j(X)| + |f_2^i(X) - f_2^j(X)|)$, $i, j = 1, 2, \dots, n'$; \bar{d} is the mean of all d_i ; n' is the number of Pareto points obtained. $f_1^i(X)$ and $f_2^i(X)$ are values of the first and second criterion of the i -th point. The zero value of this measure indicates that all the points are uniformly distributed on the Pareto front.

In the case of interactive methods a decision support system (DSS) is necessary. DSSs for multiple criteria optimization can be classified into two groups: systems of general purpose and problem-oriented systems. The first group of DSSs serves to aid the solution of different problems of multiple criteria optimization. The second group of

DSSs serves to solve one or several types of complicated practical multiple criteria optimization problems and a problem-oriented graphical user interface (GUI) is developed for them.

After a comprehensive review, some conclusions are made. Disadvantages of the WS method are as follows: the solutions are not uniformly distributed on the Pareto front and if the objective functions are non-convex, not all Pareto optimal solutions are obtained. A demerit of the AWS method is that it is not interactive. It is necessary to develop such a solving way that integrates these two methods and does not have these disadvantages, but keeps the advantages. In order to solve MOPs interactively, a decision support system must be developed.

3. An Interactive Way for Solving Multiple Criteria Optimization Problems and a Decision Support System

In this chapter, an interactive way for solving MOPs that integrates the WS and AWS methods is proposed. The model of interactive DSS for solving MOPs is described. A DSS is developed according to this model. The interactive way for solving MOPs proposed is implemented in this DSS.

The procedures of the solving way proposed are as follows:

1. A MOP is solved by the WS method:
 - 1.1. A DM selects values of weighting coefficients $w = (w_1, w_2, \dots, w_m)$;
 - 1.2. The SOP is optimized and a solution is found;
 - 1.3. The DM evaluates the solution obtained. If the solution satisfies the DM's preferences, the decision process is terminated. Otherwise, two alternatives are possible: (a) the DM changes the values of the weighting coefficients $w = (w_1, w_2, \dots, w_m)$ according to the solution obtained and afterwards goes to Step 1.2.; (b) the DM goes to Step 2.
2. The MOP is solved by the AWS method.
3. All solutions, obtained by the AWS method, are saved.
4. The DM reviews and evaluates the obtained solutions. If the DM finds a solution that satisfies his preferences, the decision process ends up. Otherwise, the DM selects one or several solutions obtained, changes the values of the weighting coefficients $w = (w_1, w_2, \dots, w_m)$, and goes to Step 1.2.

As mentioned before, the AWS method is based on the WS method. The integration of the AWS method into the proposed solving way ensures the uniform distribution of the solutions obtained on the Pareto front. So, the DM has more different alternative solutions, which may not have been obtained, when the DM selects the values of the weighting coefficients $w = (w_1, w_2, \dots, w_m)$. However, the number of solutions, obtained by the AWS method, should not be too large. Otherwise, the DM would not be able to evaluate the solutions within a reasonable time and to select the most appropriate ones, so the method would lose its interactivity.

In the proposed interactive way for solving MOPs, the DM can participate not only in the decision-making, but also in the decision process, selecting the values of the weighting coefficients. Thus, the DM uses his expertise experience, knowledge of the specifics of the problem and directs the decision process to the desired way. The proposed solving way ensures interactivity.

When the MOPs are solved interactively, usually the decision process is iterative: a lot of SOPs are solved, many solutions are obtained, the DM selects the most appropriate solution. A SOP can be solved by a computer independently of the others. So, MOPs are solved using computer clusters. Master/slaves model is used for the decision process parallelization. A SOP with a set of values of the weighting coefficients $w = (w_1, w_2, \dots, w_m)$ is called a *task*.

The DSS is needed to implement the interactive way for solving MOPs. The model of interactive DSS for solving MOPs, when the problem is transformed to a SOP using scalarization, is described here. The model integrates optimization, visualization and parallelization of the decision process.

The main parts of the model are as follows: graphical user interface, a list of tasks, a module for the connection with computer-slaves, module for the connection with the computer-master, optimization toolbox, a list of unreviewed solutions, and a list of the saved solutions and database.

Matlab (ver. R2009a) is selected for implementing the DSS according to this model. It is a high-level programming language and the interactive environment that allow us to perform computationally intensive tasks. Since Matlab is adjusted to matrix computation, the calculations with arrays (vectors and matrices) are fast. Solution of MOPs is implemented in the DSS developed; therefore the most part of the calculations is namely with the arrays. Many specialized toolboxes are integrated into Matlab. An optimization toolbox provides wide used algorithms for the standard and large-scale optimization. Constrained and unconstrained continuous and discrete optimization problems can be solved by these algorithms. The toolbox includes functions for linear programming, quadratic programming, binary integer programming, nonlinear optimization, and MO. A parallel computing toolbox is implemented in Matlab. It allows us to solve computationally and data intensive problems using multicore processors, computer clusters, and grids. Message passing interface (MPI) is applied here. Moreover, parallel for-loops, special array types, and parallelized numerical algorithms enable us to parallelize algorithms without MPI programming. A Matlab GUIDE allows us to create GUI. A Matlab compiler lets us create application as an executable or a shared library.

In this dissertation, a new strategy (called second) for solving MOPs interactively using a computer cluster is proposed and the strategy is compared with another strategy (called first), proposed by another author.

First strategy. Only a DM forms all tasks and evaluates the solutions obtained. Using the first strategy, it is difficult for the DM to form the tasks such that all computer-slaves were busy. The selection of values of the weighting coefficients lasts rather long. The DM delays more time forming tasks especially at the beginning of the decision process. The more computer-slaves are used, the lower business of computers is. So, the usage of many computers is not effective in solving MOPs by the first strategy.

Second strategy. The DM forms the tasks, but when he is late to do that, the computer-master generates the tasks. The computer generates values of the weighting coefficients that are close to the values, with which the DM had obtained the best solution, in his opinion. The values of the weighting coefficients are generated randomly adding or subtracting random numbers that do not exceed 10 % of the values, with which the best solution was obtained before. In the second strategy, the problem of idle

computer-slaves is solved. The computer-slaves become helpers to the DM in task formation.

The MOP of optimal selection of the nutritional values is chosen for the analysis of the proposed interactive way for solving MOPs and testing the DSS developed. The objective of the problem is to minimize farmers' expenditure on nutrition by the optimal selection of feed ingredients in cattle-breeding. It is taken into consideration that cattle diets consist of different ingredients (e.g., maize, corn, peas, fish oil, etc.) on the one hand, and each ingredient differs by different nutritional characteristics (e.g., protein, calcium, natrium, etc.), on the other hand. The cost price as well as violations of the requirements to the values of nutritional characteristics must be minimized. Thus, the MOP is solved.

Solving the problem, two groups of contradictory criteria are as follows:

1. The feed cost price

$$f_1(X) = \sum_{i=1}^n x_i \psi_i. \quad (3.1)$$

2. Violations of the requirements of the values of nutrition characteristics

$$f_j(X) = \begin{cases} 0, & \text{if } R_j^{\min} \leq R_j \leq R_j^{\max} \\ R_j - R_j^{\max}, & \text{if } R_j - R_j^{\max} > 0 \\ R_j^{\min} - R_j, & \text{if } R_j^{\min} - R_j > 0 \end{cases} \quad (3.2)$$

where $R_j = \sum_{i=1}^n x_i A_{ij}(X)$, $j = 2, \dots, m$; x_i is a constituent part of the i -th ingredient in feed; n is the number of ingredients; $(m - 1)$ is the number of nutritional characteristics; R_j^{\min} and R_j^{\max} are the recommended permissible minimal and maximal amount of the j -th nutritional characteristic; $A_{ij}(X)$ is a nonlinear function that expresses the value of the j -th nutritional characteristic of the i -th ingredient; ψ_i is the price of the i -th ingredient for a weight unit; and the ingredients $X = (x_1, \dots, x_n)$ influence the nutritional characteristics of the other ingredients.

Criterion (3.1) is contradictory to the group of criteria (3.2). With an increase in violations of the permissible amount of nutritional characteristics, the price of feed is falling. Furthermore, it is necessary to obtain a solution in which a combination of violations and the cost price appropriates the DM.

In accordance with the fact that the sum of the constituent parts x_i of ingredients must be equal to one and the sum of the values of the weighting coefficients must be equal to one, the following formulation of the problem has been proposed:

$$\min_{X \in D} \left(w_1 f_1(X) + \sum_{j=2}^m w_j f_j^2(X) \right), \quad (3.3)$$

subject to: $\sum_{i=1}^n x_i = 1$, $\sum_{j=1}^m w_j = 1$, $x_i^{\min} \leq x_i \leq x_i^{\max}$,

where x_i^{\min} and x_i^{\max} are the minimal and maximal value of the constituent part of the i -th ingredient; n is the number of variables; and m is the number of criteria. In experimental investigations, the values $n = 50$, $m = 15$ are used.

A peculiarity of problem (3.3) is the use of squared criteria $f_j(X)$, $j = 2, \dots, m$ in the objective function. Similarly the transformation of the definition domain into a rectangular one, it broadens the scale of potential local optimization algorithms. Furthermore, the objective function (3.3) is convex.

When solving a MOP, the graphical representation plays an important role in decision-making. The decision support system is designed with GUI that facilitates the DM work and permits us to review the results and to plan the decision process. The developed DSS is slightly changed according to the specificity of the problem of the optimal selection of nutrition values.

The main window of the DSS is presented in Fig. 1. The top right corner of the window displays the last obtained or edited solution. Fourteen horizontal bars present deviations from the norm of values of the corresponding nutritional characteristics, i.e., violations of the requirements, when the values of the weighting coefficients $w = (w_1, w_2, \dots, w_m)$, $m = 15$, presented on the left, are selected. The numerical values of violations are presented on the right. On the top of fourteen horizontal bars, the value of the cost price (*Cost Price*) is located. At the bottom, the sum of violations of the requirements (*Sum of Violations*) is presented.

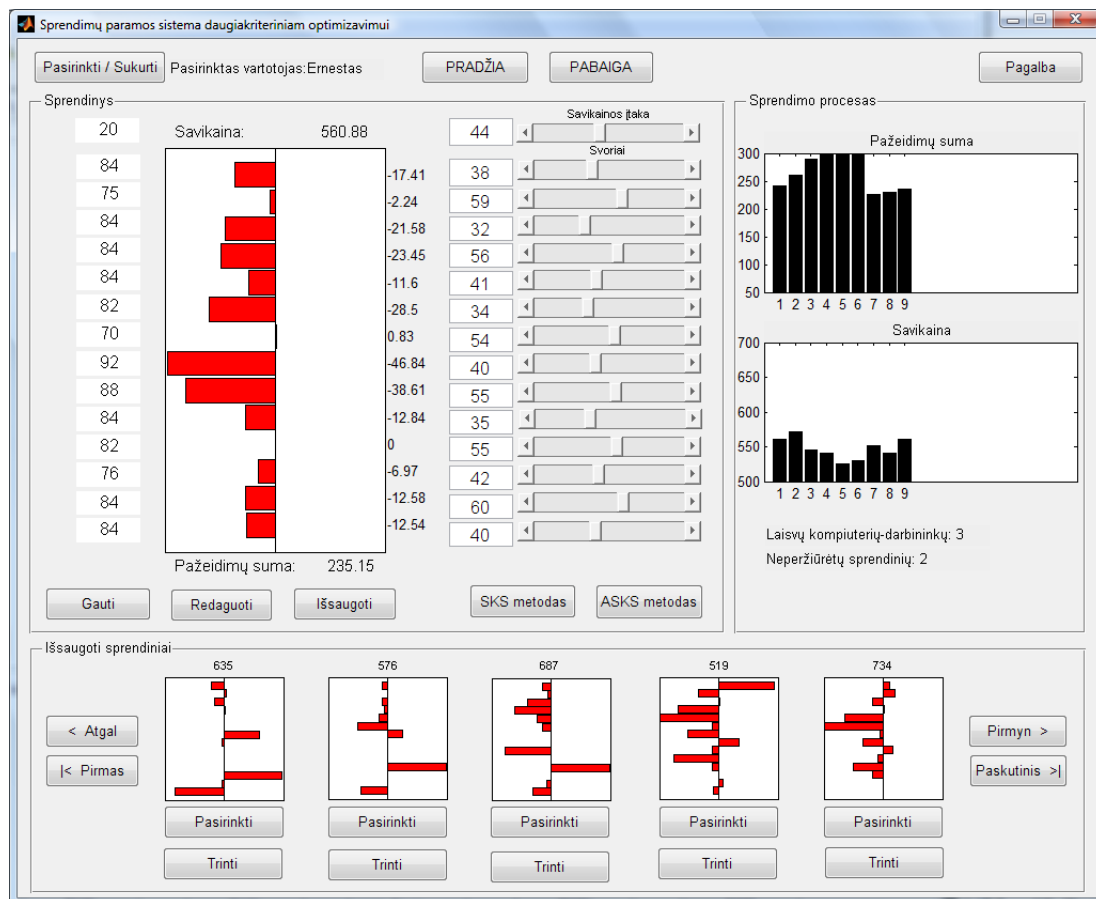


Fig. 1. Graphic user interface of the decision support system

A block in the middle of the DSS is designed for changing the weighting coefficients. On the right of the window, the column chart of the sums of violations and the values of cost prices, obtained in each iteration, are presented in order to observe the

solving process. The bottom of the window presents the solutions (5 blocks) that have been obtained and saved up to the moment. The blocks display only five saved solutions, nevertheless, it is possible to review and use any other saved solution for further editing.

Moreover, some different DMs can solve the same problem independently. Their solutions obtained are saved in the database and later on a comparative analysis can be made and the best result achieved can be selected.

MOP (3.3) is 15-criteria problem. However, the implementation of the AWS method is very complicated in the case of more than three criteria. Moreover, the computations are time-consuming. According to the fact that criterion (3.1) is contradictory to the group of criteria (3.2), problem (3.3) is aggregated to a two-criteria problem:

$$\min_{X \in D} (w_1' f_1(X) + w_2' f_2'(X)), \quad (3.4)$$

where $f_1(X)$ is the cost price, $f_2'(X) = \sum_{j=2}^{15} f_j(X)$ is the sum of violations of the requirements, and $w_1' + w_2' = 1$.

The general schema of solving MOP (3.3) using the DSS is described as follows:

1. MOP (3.3) of 15 criteria is solved by the WS method. A DM selects values of the weighting coefficients $w = (w_1, w_2, \dots, w_m), m = 15$. SOPs with various sets of values of the weighting coefficients are solved. The DM evaluates the solutions obtained and saves the most preferable ones.
2. If the DM fails to obtain preferable solutions by the WS method, then the two-criteria problem (3.4) is solved by the AWS method. The cost prices and all 14 violations of the requirements of the solutions are computed. The graphical view of violations is shown to the DM.
3. The DM evaluates the solutions obtained by the AWS method and selects the most preferable ones. If no preferable solutions are obtained, the DM can try to improve the solutions by selecting other values of the weighting coefficients $w = (w_1, w_2, \dots, w_m), m = 15$ and solving problem (3.3) by the WS method.
4. The DM stops the decision process when the preferable solutions are found.

The Matlab function *fmincon* is used for solving a SOP. This function includes some optimization algorithms: Trust-Region-Reflective Optimization, Active-Set Optimization, Interior-Point Optimization, SQP Optimization. The function *fmincon* selects the most suitable method or a combination of methods for solving a problem.

4. Experimental Investigations

In this chapter, the results of experimental investigations are presented. Two solving strategies for solving MOPs interactively are compared. The DM's learning to solve MOP using the DSS is investigated. The interactive way for solving MOPs is explored experimentally.

In the case of MOPs, it is difficult to compare solutions numerically, because the DM must evaluate preferences of the values of all criteria and the DM selects the most preferable solution in his opinion. But, in order to estimate the solving strategies, we need the numerical estimation. To this end, the so-called combined criterion is used. The values of the combined criterion are calculated by the formula:

$$K_t = \sqrt{(\hat{f}_{1t})^2 + (\hat{f}'_{2t})^2}, \quad (4.1)$$

where t is the time moment; \hat{f}_{1t} is the normalized cost price, obtained at the moment t ; \hat{f}'_{2t} is the normalized sum of violations of the requirements, obtained at the moment t ; and $f'_{2t} = \sum_{j=2}^m f_{jt}(x_1, \dots, x_n)$. The values \hat{f}_{1t} and \hat{f}'_{2t} are arranged in the interval $[0, 1]$, according to the minimal and maximal values obtained during all the experiments.

In order to compare the solving strategies, a percentage estimator v is introduced:

$$v = \sum_{t=1}^T \frac{K_{t2}}{K_{t1}} \times 100 \%, \quad (4.2)$$

where t is a time moment ($t = 1, \dots, T$); T is the time of solving the problem; K_{t1} and K_{t2} are the values of the combined criterion for the first and second cases compared at the moment t . The smaller value v , the greater the difference between two cases is.

The objectives of comparison of the strategies are to detect the effect of the number of computers used in the process and to compare the efficiency of different strategies, depending on the number of computers applied in the calculations.

MOP (3.3) has been solved in a computer cluster by one, six, 12, 24 computer-slaves where time expenditure for a SOP is one minute. The problem was solved at least 30 minutes. Each experiment has been iterated for 10 times. The averages of the combined criterion values are computed.

The results obtained in solving problem (3.3) by the first strategy, when one and six computer-slaves were used, are presented in Fig. 2. As expected, when solving the problem by six computer-slaves, better results are reached in less time ($v = 51 \%$). Usage of six computers allows the DM to achieve almost twice better results in the sense of the combined criterion. A further increase in the number of computer-slaves is senseless – the DM cannot afford to form new tasks for more than six computer-slaves without idleness of the computer-slaves.

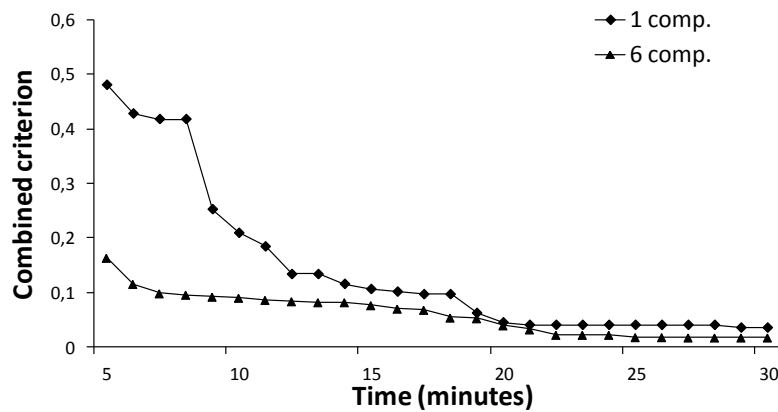


Fig. 2. Dependence of efficiency of the first strategy on the number of computer-slaves

Fig. 3 illustrates the results obtained by the second strategy using six, 12, and 24 computer-slaves. An increase in the number of computer-slaves provides better results faster. The minimal value $K_t = 0,014$ is achieved using six computer-slaves. Applying 12 computer-slaves, the minimal value $K_t = 0,017$. In the case of

24 computer-slaves, the minimal value $K_t = 0,016$. When comparing the results achieved using six and 12 computer-slaves, $v = 67 \%$, and the results achieved using 12 and 24 computer-slaves, $v = 77 \%$. The results obtained show that usage of a large number of computer-slaves allows the DM to achieve better results in less time. However, it is not necessary to increase the number of computer-slaves, because the difference between the results obtained by 12 and 24 computer-slaves is less than the results obtained by six and 12 computer-slaves. Moreover, the DM would not be able to evaluate the solutions obtained.

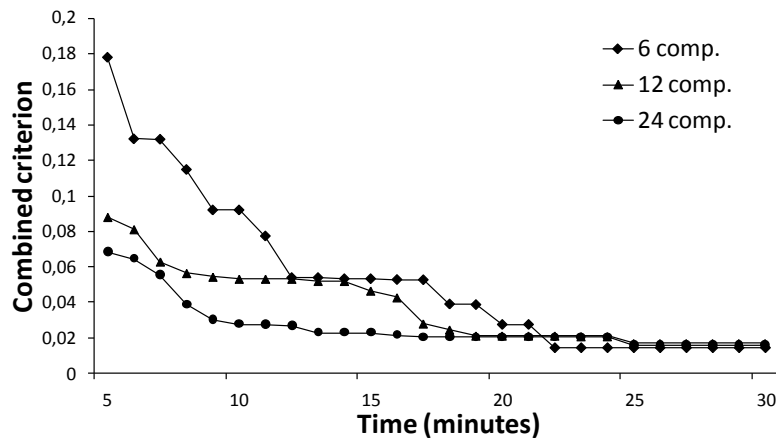


Fig. 3. Dependence of efficiency of the second strategy on the number of computer-slaves

The results of the first and second strategies are shown in Fig. 4.

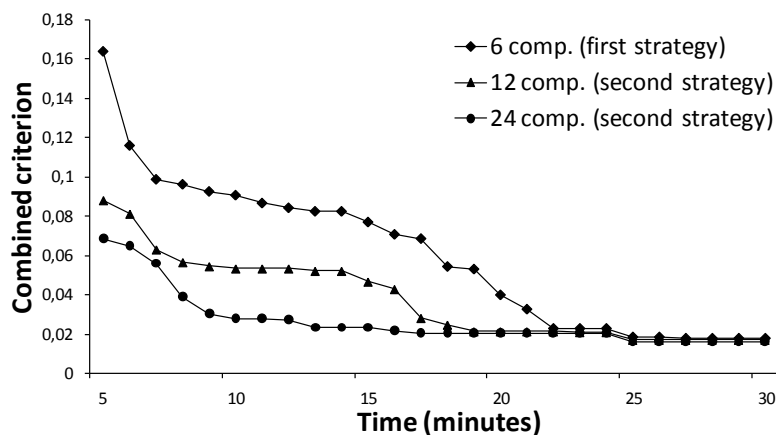


Fig. 4. Comparison of the first and second strategies

When comparing the results obtained by six and 12 computer-slaves, achieved until the 10-th and 20-th minute $v = 56 \%$; until the 30-th minute $v = 67 \%$. When comparing the results obtained by six and 24 computer-slaves, achieved until the 10-th minute $v = 40 \%$; until the 20-th minute $v = 37 \%$; until the 30-th minute $v = 53 \%$. A great advantage of the second strategy is obvious in comparison with the first strategy, especially at the beginning of the decision process.

It is important to investigate how DMs learn to solve MOPs using the DSS. Five different cases are analyzed: the first strategy using one computer-slave (denote it as *first (1 comp.)*) and six computer-slaves (*first (6 comp.)*), the second strategy using six

computer-slaves (*second (6 comp.)*), 12 computer-slaves (*second (12 comp.)*), and 24 computer-slaves (*second (24 comp.)*). 50 decision makers took part in this investigation, i.e., solving MOP (3.3) (10 DMs in each of the five cases). Each DM has iterated the experiment for ten times. The average values of the combined criterion, obtained by all the DMs in all the experiments, have been calculated and presented in Fig. 5.

The best results (the minimal values of the combined criterion) are obtained when the MOP is solved by the second strategy using 24 computer-slaves, and the worst results are obtained, when the problem is solved by the first strategy using one computer-slave. Since the averaged results are presented here, we can state that the second strategy is superior to the first one, indeed.

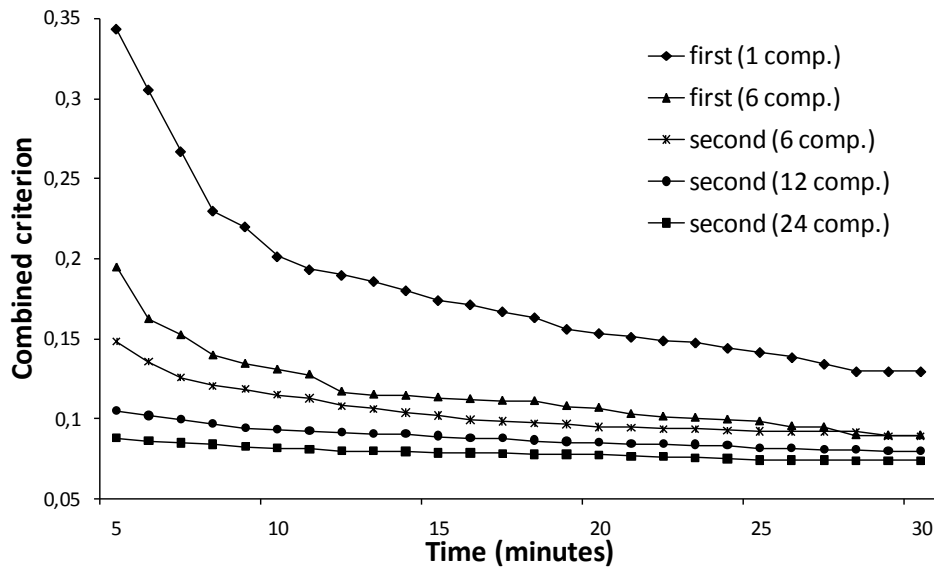


Fig. 5. Average values of the combined criterion

It is important to compare how efficiently the experienced DM and non-experienced DMs manage to use the computer-slaves. The values v , obtained solving MOP by the first and second strategies are presented in Table 1.

Table 1. The values v in the cases of the experienced DM (E) and non-experienced DMs (N)

Time moment (minutes)	<i>first (1 comp.)</i>		<i>first (6 comp.)</i>		<i>first (6 comp.)</i>		<i>second (6 comp.)</i>		<i>second (12 comp.)</i>	
	<i>second (6 comp.)</i>		<i>second (12 comp.)</i>		<i>second (24 comp.)</i>		<i>second (12 comp.)</i>		<i>second (24 comp.)</i>	
	E	N	E	N	E	N	E	N	E	N
10	34 %	58 %	56 %	53 %	40 %	47 %	52 %	78 %	71 %	86 %
20	51 %	62 %	56 %	66 %	37 %	59 %	65 %	83 %	67 %	88 %
30	51 %	64 %	67 %	73 %	53 %	65 %	84 %	86 %	77 %	89 %

We see that in the cases of the experienced DM, almost all the values v are smaller. So, the experienced DM manages to use a larger number of computer-slaves more efficiently. Moreover, the experienced DM and non-experienced DMs use computer-slaves more efficiently at the beginning of the decision process.

The results of the data analysis, where the values of the combined criterion are achieved at the end of the experiments, are presented in Table 2. The duration of one

experiment was 30 minutes; therefore we analyze the best results obtained up till this moment. Three smallest values of the combined criterion are written in bold type for each analyzed case. When the problem is solved by the first strategy using only one computer-slave, the best results are obtained during the last experiment. It is supposed that the reason is the small number of intermediate solutions, therefore more experiments are necessary for the DM to learn to solve the problem.

When the DM is solving the problem by the first strategy using six computers, he learns faster (Table 2, experiments No. 5-7), because he obtains and estimates much more intermediate solutions. When a computer assists the DM to form the tasks (the second strategy, six computer-slaves), much better results are obtained. The DMs learn faster, if they are solving the problem by the second strategy using 12 computer-slaves. The reason is that the DM has a chance to analyze many intermediate solutions. However, if the DM solves the problem using 24 computer-slaves, the learning process is slower and the results up to the fifth experiment are similar to that obtained by applying 12 computer-slaves. Starting from the sixth experiment the results “exceed” other cases. We conclude that if more computers are used, a DM learns to find a preferable solution slower, however better results are obtained last of all.

Table 2. Average values of the combined criterion obtained up till the end of each experiment

Number of experiment	Case					Average values
	<i>first</i> (1 comp.)	<i>first</i> (6 comp.)	<i>second</i> (6 comp.)	<i>second</i> (12 comp.)	<i>second</i> (24 comp.)	
1	0.1547	0.0920	0.0943	0.0770	0.0778	0.0991
2	0.1441	0.0923	0.0801	0.0758	0.0742	0.0933
3	0.1436	0.0935	0.0857	0.0793	0.0747	0.0954
4	0.1246	0.0988	0.0939	0.0832	0.0740	0.0949
5	0.1144	0.0822	0.0872	0.0776	0.0742	0.0871
6	0.1779	0.0842	0.0790	0.0835	0.0736	0.0996
7	0.1209	0.0762	0.0876	0.0806	0.0732	0.0877
8	0.1045	0.0974	0.1022	0.0853	0.0752	0.0929
9	0.1289	0.0915	0.0940	0.0828	0.0717	0.0938
10	0.0987	0.0880	0.0912	0.0835	0.0730	0.0869

The experimental results, obtained by the interactive way proposed for MOPs, are presented here. At first, the distribution of solutions on the Pareto front, when solving problem (3.4) by the WS method, is investigated. The values of the weighting coefficient w'_1 are selected from 0 to 1 at different steps s , and the values of the weighting coefficient w'_2 are calculated by the formula $w'_2 = 1 - w'_1$. Three different steps $s = 10, 50, 100$ are selected and the obtained solutions are presented in Fig. 6 (black points). We see that the solutions are not uniformly distributed on the Pareto front. With an increase of the step value s , the number of solutions increases too, but some solutions are coincident and they are crowded together in some groups. No solutions are obtained in some parts of the Pareto front.

The goal of the following experimental investigation is to show the necessity to apply the AWS method to distribute the solutions on the Pareto front uniformly in

feasible regions. The AWS method finds the solutions in those parts of the Pareto front in which the WS method was unable to find them (Fig. 6, white points). We see that the solutions by the AWS method uniformly fill in the parts of the Pareto front, which were not filled in by the WS method. However, some parts are not filled, in because no feasible points were found there. It depends on the specificity of problem (3.4). Moreover, the experimental results show that a small step value s is enough for the AWS method to obtain non-coincident solutions and a uniform distribution on the Pareto front.

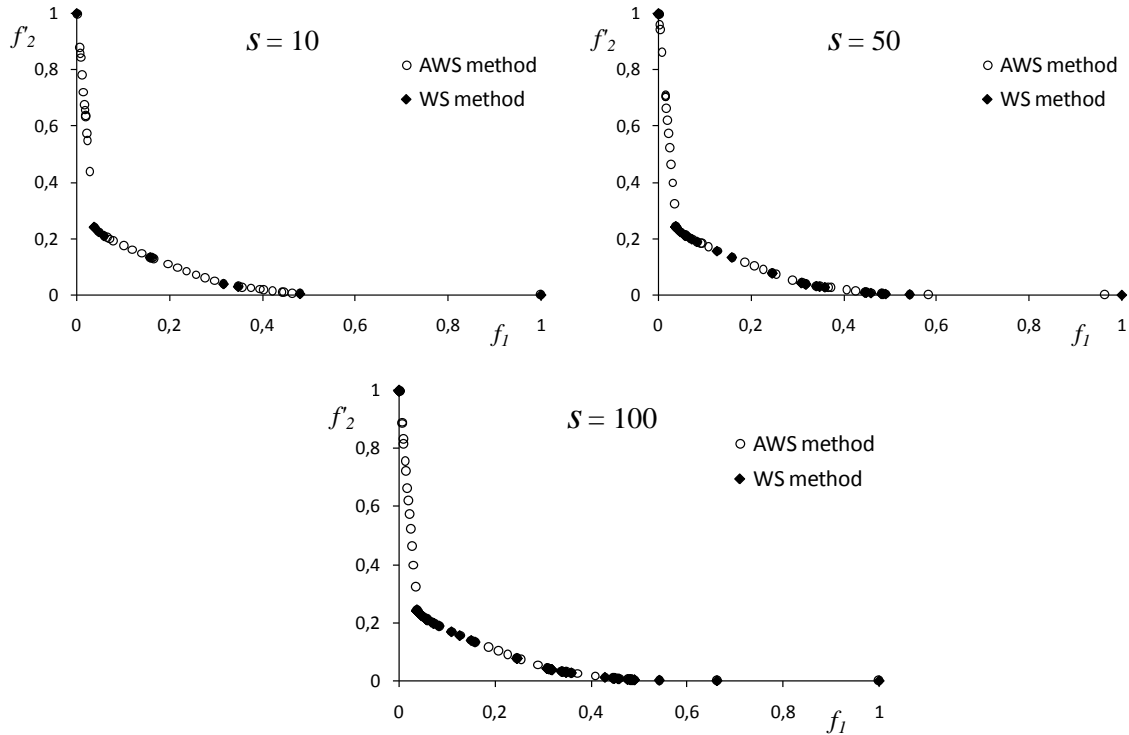


Fig. 6. Solutions obtained by the WS and AWS methods with various values s

The spread φ (2.5) is computed for the solutions obtained in order to show how the solutions are distributed on the Pareto front. The results are presented in Table 3. Here n' is the number of the solutions on the Pareto front. The number n' is smaller than the value s , because some solutions are coincident. When the value s is the same, the spread values φ are significantly smaller for the solutions obtained by the AWS method. The value φ of the solutions, obtained by the WS method, is higher even when the value n' is high ($n' = 66$) comparing with the value φ of the solutions, obtained by the AWS method, when the value n' is smaller ($n' = 45$).

Table 3. The spread values φ applying the WS and AWS methods

s	WS method		AWS method	
	n'	φ	n'	φ
10	11	0.261	45	0.084
50	37	0.132	52	0.056
100	66	0.110	52	0.053

Moreover, this experiment shows that using the AWS method, the small value s is enough to obtain a sufficient number of the solutions uniformly distributed on the Pareto front. The difference between the values φ is not essential. Thus, we suggest to use the step value $s = 10$ in the proposed interactive way for solving MOPs.

The essence of the proposed interactive way for solving MOPs is to provide a possibility for the DM to obtain the solutions from the whole Pareto front. The DM obtains alternative solutions uniformly distributed on the Pareto front. Moreover, the DM can select the most preferable solutions, obtained by the AWS method, and try to improve the solutions, specifying his preferences to the criteria, by changing the values of the weighting coefficients.

In the investigated case, two-criteria problem (3.4) is solved by the AWS method. The cost price and 14 violations of the requirements to the nutritional characteristics are computed. The results obtained (Fig. 7) are presented to the DM. When solving the two-criteria problem, violations of the requirements to each nutritional characteristic, were not considered, only their sum. However, the DM has information on possible values of the cost price and the sum of violations. The DM can try to reach these values by changing the preferences (the values of the weighting coefficients) for each nutrition characteristic and the cost price, i.e. by solving 15-criteria problem using the WS method.

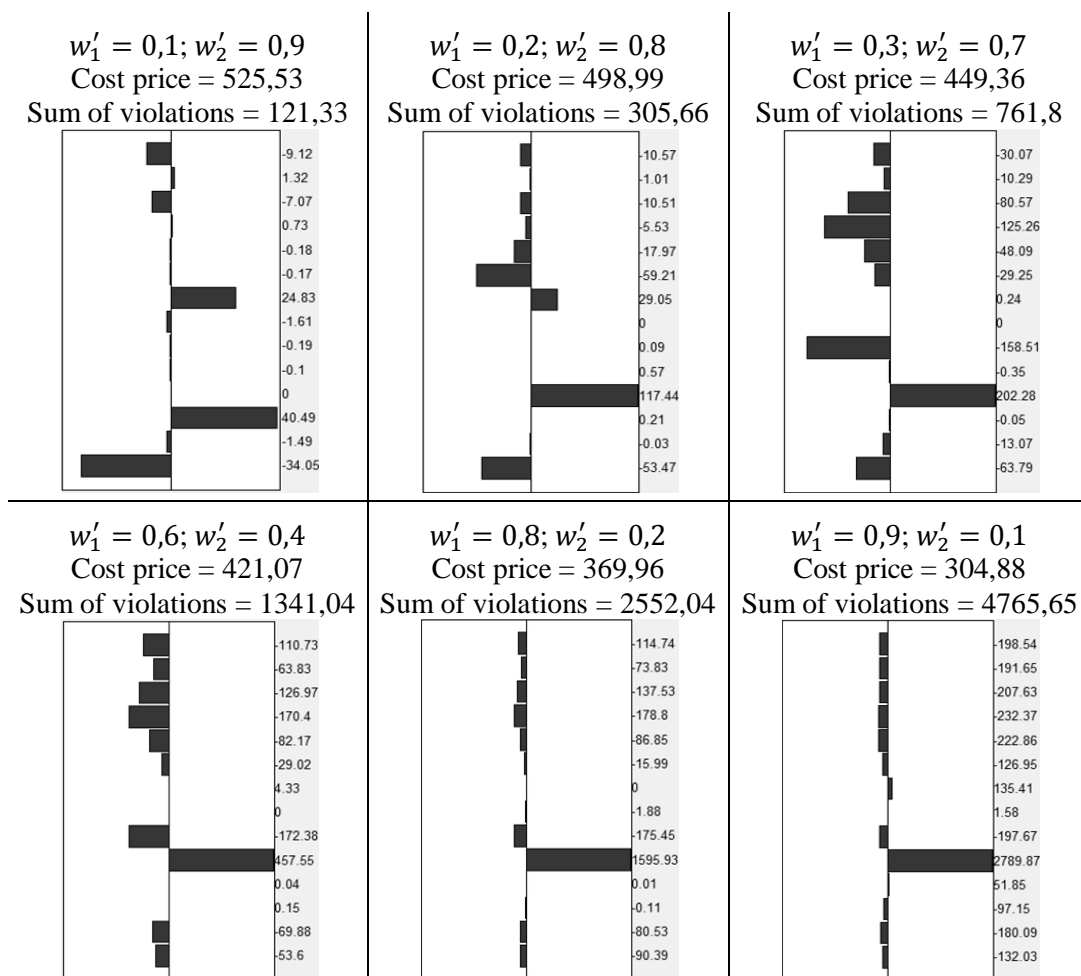


Fig. 7. Samples of solutions, obtained by the AWS method and presented to DM

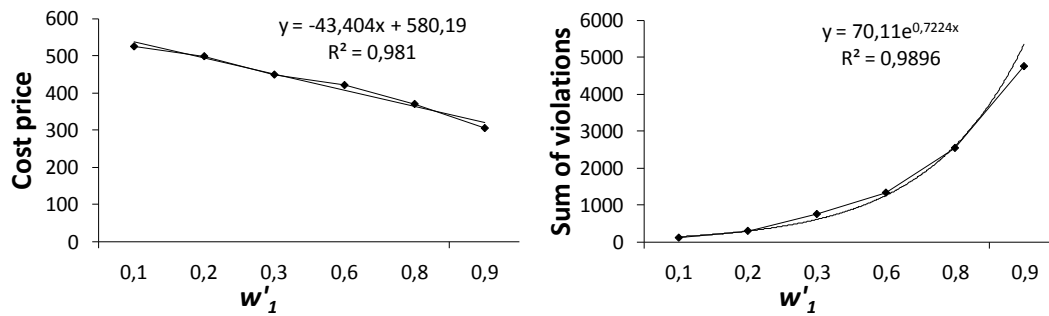


Fig. 8. Dependence of the cost price and the sum of violations on the values of the weighting coefficients

As we see in Fig. 8, with an increase in the weighting coefficient w'_1 , the cost price is decreasing linearly and the sum of violations is increasing exponentially.

It is evident that the results, obtained by the AWS method, differ from the results, obtained by the DM, when he changes the values of the weighting coefficients and the problem is solved by the WS method. On the left side of Fig. 9, the solutions obtained by the AWS method are presented. The solutions, obtained by the WS method after the correction of the values of the weighting coefficients by the DM, are presented on the right side. We see that the sum of violations increases, but some violations decrease. This fact is important to the DM.

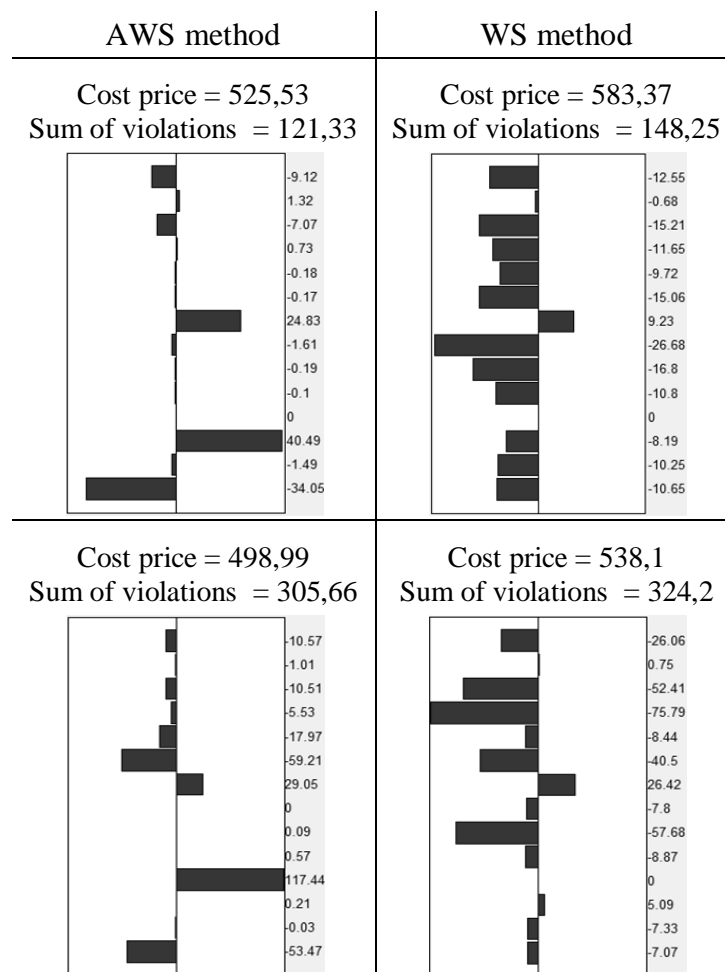


Fig. 9. Samples of solutions, obtained by the interactive way proposed

General Conclusions

The research results have shown new capabilities of the proposed interactive way for solving multiple criteria optimization problems, which integrates the weighted sum method and the adaptive weighted sum method, as well as that of the interactive decision support system.

The results of the experimental research allow us to draw the following conclusions:

1. The way proposed for solving multiple criteria optimization problems allows finding the solutions uniformly distributed on the Pareto front. Using the interactive decision support system developed, after reviewing the obtained solutions, the decision maker has information on possible values of the criteria that is taken into consideration when solving the problem interactively. Thus, the decision maker has initial solutions from the whole Pareto front for a further search of the preferable solution.
2. When the multiple criteria optimization problem is solved by the strategy proposed, according to the results, obtained at the beginning of the decision process (until 10-th minute), the average of the combined criterion values, obtained using 12 and 24 computer-slaves, is 52 % and 71 % (respectively) over the average of the combined criterion values, obtained using six computer-slaves. When using more computer-slaves (>24), the decision maker would not be able to evaluate the solutions obtained and idleness of computers is unavoidable.
3. When comparing the results of the strategy proposed with the other strategy, it is estimated that the strategy proposed is superior especially at the beginning of the decision process. According to the results, obtained until 10-th minute, the average of the combined criterion values, obtained using 12 and 24 computer-slaves by the strategy proposed, is 56 % and 40 % respectively higher than the average of the combined criterion values, obtained using six computer-slaves by another strategy.
4. When investigating the time required for a decision maker to learn to solve a multiple criteria optimization problem using the interactive decision support system, we have estimated, that decision makers learn faster while using 12 computer-slaves, than 24 computer-slaves, but the learning process is slower, though finally better results are obtained.
5. When comparing the results obtained by the experienced and non-experienced decision makers, the percentage averages of the combined criterion values obtained by the experienced decision maker are smaller. So, the experienced decision maker manages to use a greater number of computer-slaves more efficiently.

List of the Author's Publications on the Subject of the Dissertation

Articles in the reviewed scientific periodical publications:

1. Petkus T.; Filatovas E. 2008. Decision Making to Solve Multiple Criteria Optimization Problems in Computer Networks. *Information Technology and Control*, Vol. 37(1), 63–68. ISSN 1392-124X. (ISI Web of Science, VINITI, INSPEC). (Impact Factor 2010: 0,638).
2. 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, Vol. 15(3), 464–479. ISSN 1392-8619. (ISI Web of Science).

3. Ivanikovas S.; Filatovas E.; Žilinskas J. 2009. Experimental Investigation of Local Searches for Optimization of Grillage-Type Foundations. In: R. Čiegis, D. Henty, B.Kågström, J. Žilinskas (Eds.), *Parallel Scientific Computing and Optimization, Springer Optimization and Its Applications*, Vol. 27, 103–112. ISSN 1931-6828. (ISI Web of Science (Conference Proceedings Citation Index), SpringerLink, INSPEC, Zentralblatt MATH).
4. Filatovas E.; Kurasova O. 2011. A Decision Support System for Solving Multiple Criteria Optimization Problems. *Informatics in Education*, Vol. 12(2), 1–12. ISSN 1648-5831. (VINITI, INSPEC).

Articles in other editions:

1. Filatovas E.; Kurasova O. 2009. Decision Support System for the Optimal Selection of Feed Ingredients. *Proceedings of Euro Mini Conference “Knowledge-Based Technologies and OR methodologies for Strategic Decisions of Sustainable Development”* (KORS-2009), 58–63, Vilnius: Technika. ISBN 978-9955-28-482-6.

Short description about the author

Ernestas Filatovas received a Bachelor’s degree in mathematics from the Vilnius Pedagogical University in 2004 and Master’s degree in informatics in 2006. 2006–2011 – PhD studies at the Institute of Mathematics and Informatics, System Analysis Department. He is a member of the Lithuanian Computer Society.

DAUGIAKRITERINIŲ OPTIMIZAVIMO UŽDAVINIŲ SPRENDIMAS INTERAKTYVIUOJU BŪDU

Tyrimų sritis ir problemos aktualumas

Jau nuo senų laikų žmonės susiduria su optimizavimo problemomis ir ieško optimalių sprendimų. Žmonijai vystantis ir didėjant žmonių poreikiams, optimizavimo problemos vis sudėtingėja. Šiais laikais optimizavimo uždaviniai sprendžiami pasitelkiant optimizavimo teorines žinias bei kompiuterių resursus, kas žymiai palengvina uždavinių formalizavimą, sprendimo metodų parinkimą bei paspartina uždavinių sprendimą.

Praktikoje optimizavimo uždaviniai dažnai būna daugiakriteriniai. Tokie uždaviniai sprendžiami daugelyje žmonijos veiklos srityse: procesų valdyme, ekonomikoje, lėktuvų konstravime, tiltų statyme ir kitose. Su daugiakriteriniais optimizavimo uždaviniais susiduriama ir kasdieniniame gyvenime: perkant automobilį, renkantis poilsinę kelionę, sudarant maitinimosi racioną. Tokie uždaviniai sutinkami ten, kur reikia pasirinkti kompromisą tarp dviejų ar daugiau kriterijų, o kriterijai dažniausiai būna prieštaringi – mažinant vieno kriterijaus reikšmę, kito didėja. Pavyzdžiui, siekiant padidinti pelną, būtina sumažinti išlaidas; didinant automobilio galingumą, siekiama sumažinti kuro sąnaudas; mažinant tam tikros detalės svorį, būtina padidinti jos atsparumą. Dažnai kasdieniniame gyvenime daugiakriteriniai uždaviniai sprendžiami intuityviai. Tačiau sėkmingai tokio tipo uždaviniai sprendžiami tik matematiniais metodais. Daugiakriteriniai uždaviniai gali turėti kelis ar keliasdešimt kriterijų, ir priimant sprendimą reikia visus juos suderinti. Dažnas atvejis, kai tokio tipo uždaviniai neturi vieningo optimalaus sprendimo, tuomet ieškomi tik Pareto prasme optimalūs sprendiniai, dar vadinami nedominuojančiais.

Sprendžiant daugiakriterinį optimizavimo uždavinį, gaunama daug skirtingų Pareto optimalių sprendinių, o galutinis sprendimas priklauso nuo eksperto (sprendimo priėmėjo), kuris dalyvauja uždavinio sprendime. Vienas iš paprasčiausių daugiakriterinio optimizavimo būdų yra pagrįstas uždavinio suvedimu į vienakriterinį uždavinį naudojant skaliarizavimą. Dažnai tam taikomas gerai žinomas svertinės kriterijų sumos metodas (angl. *Weighted Sum Method*), kuriame sudaroma vienakriterinė tikslo funkcija, susumavus visų kriterijų tikslo funkcijas, padaugintas iš svertinių koeficientų. Galimi du svorinių koeficientų reikšmių parinkimo būdai. Vienu atveju koeficientų reikšmės generuojamos atsitiktinai, sudaromi keli koeficientų rinkiniai, sprendžiami vienakriteriniai uždaviniai, gaunami keli alternatyvūs sprendiniai, iš kurių sprendimų priėmėjas parenka jam tinkamiausią. Kitu atveju sprendimų priėmėjas pats interaktyviai parenka svorinių koeficientų reikšmes, atsižvelgdamas į sprendžiamo uždavinio specifiką, turimą patirtį bei kompetenciją. Tai įgalina sprendimų priėmėją dalyvauti ne tik sprendimo priėmime, bet ir sprendimo procese.

Svertinės kriterijų sumos metodo trūkumas be to, kad tik iškilomis tikslo funkcijoms randami visi Pareto sprendiniai, yra tas, kad gauti sprendiniai netolygiai pasiskirsto Pareto aibėje, todėl būtina ieškoti metodų, kurie minėto trūkumo neturi.

Jei pasirenkamas daugiakriterinio optimizavimo būdas, pagrįstas suvedimu į vienakriterinį uždavinį, tai uždavinys turi būti sprendžiamas daug kartų esant įvairiems svorinių koeficientų reikšmių rinkiniams. Kadangi šie vienakriteriniai uždaviniai gali būti sprendžiami nepriklausomai vienas nuo kito, sprendimo procesą galima paspartinti, pasitelkus daugiaprocesorinius kompiuterius ar kompiuterių klasterius. Būtina nustatyti

sprendimo strategijas, siekiant efektyviai išnaudoti skaičiavimo resursus, įvertinant sprendimo priėmėjo galimybes formuoti užduotis.

Dar viena problema, su kuria susiduriama, sprendžiant daugiakriterinį optimizavimo uždavinį interaktyviai, yra naudojamos sprendimų paramos sistemos kokybė. Sistema turi užtikrinti naudojimosi patogumą, greitį, tarpinių ir galutinių sprendinių vizualizavimą. Tas padeda sprendimų priėmėjui greičiau įsigilinti į sprendžiamą uždavinį ir perprasti jo specifiką, todėl būtina sukurti sistemą, kurioje būtų integruoti optimizavimo metodai, sprendimo proceso vizualizavimas bei jo lygiagretinimas.

Šioje disertacijoje sprendžiamos dvi problemos:

1. Interaktyvaus daugiakriterinių optimizavimo uždavinių sprendimo būdo, užtikrinančio sprendinių tolygų pasiskirstymą Pareto aibėje, sukūrimas ir analizė.
2. Interaktyvios sprendimų paramos sistemos sukūrimas ir analizė.

Darbo tikslas ir uždaviniai

Darbo tikslas – pasiūlyti interaktyvų daugiakriterinių optimizavimo uždavinių sprendimo būdą, kurio pagalba sprendimų paramos sistemoje, veikiančioje kompiuterių klasteryje, sprendimų priėmėjas gauna alternatyvius sprendinius, tolygiai pasiskirsčiusius Pareto aibėje.

Siekiant tikslo buvo sprendžiami šie uždaviniai:

- ištirti esamus daugiakriterinių optimizavimo uždavinių sprendimo metodus bei interaktyvias sprendimų paramos sistemas, skirtas daugiakriteriniams optimizavimo uždaviniams spręsti;
- pasiūlyti interaktyvų daugiakriterinių optimizavimo uždavinių sprendimo būdą, kuriuo surandami alternatyvūs sprendiniai, tolygiai pasiskirstę Pareto aibėje;
- sukurti interaktyvią daugiakriterinių optimizavimo uždavinių sprendimų paramos sistemą, apjungiančią pasiūlytą optimizavimo uždavinių sprendimo būdą, sprendimo proceso vizualizavimą ir jo lygiagretinimą;
- sukurti ir palyginti sprendimo strategijas, interaktyviai sprendžiant daugiakriterinį optimizavimo uždavinį, pasitelkus kompiuterių klasterį;
- ištirti darbo su sukurtos sprendimų paramos sistemos apsimokymo laiką, būtiną sprendimų priėmėjui perprasti sprendžiamo uždavinio specifiką, siekiant greičiau rasti tinkamą sprendinį.

Tyrimo objektas ir metodai

Disertacijos tyrimo objektas yra daugiakriteriniai optimizavimo uždaviniai, interaktyvūs jų sprendimo metodai, interaktyvios sprendimų paramos sistemos bei lygiagrečių skaičiavimų taikymas sprendimų paramos sistemose.

Analizuojant mokslinius pasiekimus daugiakriterinio optimizavimo bei sprendimų paramos sistemų kūrimo srityse buvo naudoti informacijos paieškos, sisteminimo, analizės, lyginamosios analizės ir apibendrinimo metodai. Remiantis eksperimentinio tyrimo metodu, atlikta statistinė tyrimų rezultatų analizė, kurios rezultatams įvertinti panaudotas apibendrinimo metodas.

Darbo mokslinis naujumas

1. Pasiūlytas interaktyvus daugiakriterinių optimizavimo uždavinių sprendimo būdas, apjungiantis svartinės kriterijų sumos ir adaptyvų svartinės kriterijų sumos metodus,

leidžiantis spręsti daugiakriterinius optimizavimo uždavinius interaktyviai ir užtikrinantis gaunamų sprendinių tolygų pasiskirstymą Pareto aibėje.

2. Eksperimentiškai palygintos sprendimo strategijos, pagal kurias sprendžiant daugiakriterinį optimizavimo uždavinį pasitelkiamas kompiuterių klasteris.
3. Eksperimentiškai ištirtas darbo su sprendimų paramos sistema apsimokymo laikas, būtinas sprendimų priėmėjui perprasti sprendžiamo daugiakriterinio optimizavimo uždavinio specifiką, siekiant greičiau rasti tinkamą sprendinį.

Darbo rezultatų praktinė reikšmė

Sukurta interaktyvi daugiakriterinių optimizavimo uždavinių sprendimų paramos sistema adaptuota pašarų sudėties sudarymo daugiakriteriniam uždaviniui spręsti. Ši sistema gali būti pritaikyta ir kitiems panašaus pobūdžio uždaviniams, pvz., dietos sudarymo, mokinių valgiaraščio sudarymo ir kt.

Dalis tyrimų atlikta pagal Lietuvos valstybinio mokslo ir studijų fondo remtą mokslininkų grupių projektą „Žmogiškojo faktoriaus tyrimas daugiakriteriniuose optimizavimo uždaviniuose skaičiuojant lygiagrečiai“, registracijos Nr. T-07134, vykdymo laikas 2007 metai.

Ginamieji teiginiai

1. Pasiūlytas daugiakriterinių optimizavimo uždavinių sprendimo būdas, kuriuo surandami alternatyvūs sprendiniai, tolygiai pasiskirstę Pareto aibėje, leidžiantys uždavinį spręsti interaktyviai.
2. Interaktyvaus daugiakriterinių optimizavimo uždavinių sprendimo būdo, sprendimo proceso vizualizavimo ir jo lygiagretinimo integravimas į sukurta sprendimų paramos sistemą užtikrina efektyvų sprendimo procesą, kas padeda sprendimų priėmėjui rasti jam tinkamą sprendinį.
3. Sukurta sprendimo strategija daugiakriteriniams optimizavimo uždaviniams spręsti, kurioje užduočių formavime dalyvauja kompiuteris, leidžia greičiau surasti sprendimų priėmėjui tinkamą sprendinį, naudojant didesnę kompiuterių skaičių.
4. Sprendimų priėmėjas greičiau apsimoko dirbti su sprendimų paramos sistema, kai uždavinio sprendimui naudojamas kompiuterių klasteris, tačiau kompiuterių skaičius turi būti toks, kad sprendimų priėmėjas spėtų įvertinti gaunamus sprendinius.

Darbo rezultatų aprobavimas

Tyrimų rezultatai publikuoti 5 moksliniuose leidiniuose: 4 periodiniuose recenzuojamuose mokslo žurnaluose, 1 konferencijos pranešimų medžiagoje.

Darbo apimtis

Disertaciją sudaro 5 skyriai ir literatūros sąrašas. Disertacijos skyriai: Įvadas, Daugiakriterinis optimizavimas ir sprendimų paramos sistemos, Interaktyvus daugiakriterinių optimizavimo uždavinių sprendimo būdas, Eksperimentiniai tyrimai, Bendrosios išvados. Papildomai disertacijoje pateikti naudotų žymėjimų ir santrumpų sąrašai. Bendra disertacijos apimtis 139 puslapiai, kuriuose pateikti 29 paveiksai ir 5 lentelės. Disertacijoje remtasi 193 literatūros šaltiniais.

Bendrosios išvados

Atlikti tyrimai atskleidė pasiūlyto interaktyvaus daugiakriterinių optimizavimo uždavinių sprendimo būdo, apjungiančio svertinės kriterijų sumos ir adaptyvaus svertinės kriterijų sumos metodus, ir jį įgyvendinančios sprendimų paramos sistemos galimybes. Eksperimentinių tyrimų rezultatai leido daryti šias išvadas:

1. Pasiūlytas interaktyvus daugiakriterinių optimizavimo uždavinių sprendimo būdas leidžia rasti tolygiai pasiskirsčiusius sprendinius iš visos Pareto aibės. Naudojant sukurtą sprendimų paramos sistemą, sprendimų priėmėjas peržiūrėjęs gautus sprendinius gauna informacijos apie siekiamas kriterijų reikšmes, į kurias atsižvelgia interaktyviai sprendžiamas daugiakriterinį optimizavimo uždavinį. Sprendimų priėmėjas turi eilę pradinių sprendinių iš visos Pareto aibės tolimesnei jį tenkinančio sprendinio paieškai.
2. Uždavinį sprendžiant šiame darbe pasiūlyta strategija, atsižvelgiant į rezultatus, gautus sprendimo pradžioje (iki 10-os sprendimo minutės), jungtinio kriterijaus reikšmių, gautų naudojant 12 ir 24 kompiuterius-darbininkus, vidurkis sudaro atitinkamai 52 % ir 71 % jungtinio kriterijaus reikšmių, gautų naudojant šešis kompiuterius-darbininkus, vidurkio. Naudojant didesnę kompiuterių-darbininkų skaičių (>24), sprendimų priėmėjas nespės vertinti gaunamų sprendinių ir kompiuterių prastovos neišvengiamos.
3. Lyginant tarpusavyje pasiūlytos ir pirmos strategijos, sukurtos kitų autorių, rezultatus, nustatyta, kad pasiūlyta strategija yra pranašesnė už pirmą ypač uždavinio sprendimo pradžioje, kadangi atsižvelgiant į rezultatus iki 10-os sprendimo minutės, jungtinio kriterijaus reikšmių, gautų naudojant 12 ir 24 kompiuterius-darbininkus pasiūlyta strategija, vidurkis sudaro atitinkamai 56 % ir 40 % jungtinio kriterijaus reikšmių, gautų naudojant šešis kompiuterius-darbininkus pirma strategija, vidurkio.
4. Tiriama sprendimų priėmėjo apsimokymą spręsti uždavinį naudojant sukurtą interaktyvią sprendimų paramos sistemą, nustatyta, kad sprendimų priėmėjai greičiausiai apsimoko, kai sprendimui naudojama 12 kompiuterių-darbininkų, bet naudojant didesnę jų skaičių (24) sprendimo priėmėjo apsimokymas trunka ilgai, tačiau uždavinio sprendimo pabaigoje pasiekiami geresni rezultatai jungtinio kriterijaus prasme.
5. Lyginant patyrusio ir nepatyrusių sprendimų priėmėjų gautus rezultatus, patyrusio sprendimų priėmėjo gautų jungtinių kriterijų reikšmių procentų vidurkiai yra mažesni, negu vidurkiai, kai uždavinį sprendžia nepatyrę sprendimų priėmėjai. Vadinasi patyręs sprendimų priėmėjas sugeba geriau išnaudoti didesnę kompiuterių-darbininkų skaičių.

Ernestas Filatovas

SOLVING MULTIPLE CRITERIA OPTIMIZATION
PROBLEMS IN AN INTERACTIVE WAY

Summary of Doctoral Dissertation

Technological Sciences,
Informatics Engineering (07T)

Ernestas Filatovas

DAUGIAKRITERINIŲ OPTIMIZAVIMO UŽDAVINIŲ
SPRENDIMAS INTERAKTYVIUOJU BŪDU

Daktaro disertacijos santrauka

Technologijos mokslai,
Informatikos inžinerija (07T)