INFORMATICA, 2001, Vol. 12, No. 1, 45–60

# Application of Computer Network to Solve the Complex Applied Multiple Criteria Optimization Problems

Gintautas DZEMYDA
*Institute of Mathematics and Informatics*
*Akademijos 4, 2600 Vilnius, Lithuania*
*e-mail: dzemyda@ktl.mii.lt*

Tomas PETKUS
*Department of Informatics, Vilnius Pedagogical University*
*Studentų 39, 2004 Vilnius, Lithuania*
*e-mail: petkai@takas.lt*

Received: October 2000

**Abstract.** The analysis of the method for multiple criteria optimization problems applying a computer network has been presented in the paper. The essence of the proposed method is the distribution of the concrete optimization problem into the network rather than the parallelization of some optimization method. The aim of the authors is to design and investigate the interactive strategies to solve complex multiple criteria problems by applying a computer network. The optimized objective function is the weight sum of the criteria. The multiple criteria problem is iterated by selecting interactively different weight coefficients of the criteria. Therefore, the process is organized by designating the computers as *the master* (that coordinates the process of other computers) and *the slaves* (that execute different tasks). In the beginning of the process the researcher allocates a certain number of optimization problems to the network. The objective function optimization problems differ only in weight coefficients of the criteria. As soon as the task of *a slave* has been executed, the result is sent to *the master*. Every computer of the network behaves in analogous way. Whenever the researcher receives an immediate result from one of the computers, he gives a decision taking into consideration the latter and all the previous results, i.e., he selects new weight coefficients for the criteria and assigns a new task to the network. Likewise the multiple criteria problem is solved until the result is acceptable for the researcher. The application of the proposed method is illustrated on the basis of the problem for the selection of the optimal nutritive value. Message Passing Interface (MPI) software has been used. The trials have been carried out with the network of computers under the operation system Windows NT.

**Key words:** multiple criteria optimization, parallel computing, message passing interface.

## 1. Introduction

The intensive development of new technologies requires solving complex problems of computer-aided design and control. Here the search for optimal solution acquires an essential significance. Investigations in this area are carried out in two directions: development of new optimization methods as well as software that would embrace various realizations of the methods developed.

Currently, computer networks have been widely spread and permit to solve complex optimization problems by using simple computers. Furthermore, the networks permit significantly more complex problems that can be solved by using the aggregate power of many computers.

The method for an interactive solving the complex multiple criteria optimization problem by using computer network has been suggested. The investigation in this paper presents two interactive strategies and the experimental analysis on the basis of them, which let us detect the effect of computer resources power supported to the user on the interactive solution of multiple criteria optimization problems applying a computer network. Problems of different calculation time have been examined using different number of computers.

## 2. Formulation of the Optimization Problem

In everyday life we often deal with problems of multiple criteria (see the collection of papers edited by Karpak and Zionts (1989) for formulations and examples of applications). In common case, the ideal solution with respect to one criterion can be absolutely unacceptable with respect to another. Thus, it is necessary to seek an optimal solution that could satisfy all the criteria.

Let us analyse a multiple criteria optimization problem:

$$\min_{X=(x_1,\dots,x_n)\in\overline{A}} f_j(X), \quad j=\overline{1,p}, \tag{1}$$

where $\overline{A}$ is a bounded domain in $n$-dimensional Euclidean space $R^n$,
$p$ is a number of criteria making problem (1),
functions $f_j(X): R^n \to R^1$ are criteria.

Let some functions $f_j(X)$, $j=\overline{1,m}$, $(m \leqslant p)$ among $f_j(X)$, $j=\overline{1,p,}$ have the following properties:

1. $f_j(X)=\min_{Y\in\overline{A}} f_j(Y)=0$ as $X\in\overline{A}_j\subset\overline{A}$.
2. $f_j(X)=f_j(\delta_j(X))$, i.e., functions $f_j(\cdot)$ are dependent on other functions $\delta_j(X)$.
3. $f_j(X)=\min_{Y\in\overline{A}} f_j(Y)$ as $\delta_j(X)\in[\delta^j_{\mathrm{min}},\delta^j_{\mathrm{max}}]$.

It follows from the last property that the dependence of $f_j(\cdot)$ on $\delta_j(X)$ has a zone of constant values as $\delta_j(X)\in[\delta^j_{\mathrm{min}},\delta^j_{\mathrm{max}}]$ (see Fig. 1 for example of the dependence).

One of the possible ways to solve the system of problems (1) is to make a single criterion problem by summing up all the criteria that are multiplied by positive weight coefficients $\lambda_j$, $j=\overline{1,p}$:

$$\min_{X=(x_1,\dots,x_n)\in\overline{A}} \sum_{j=1}^{p}\lambda_j f_j(X). \tag{2}$$

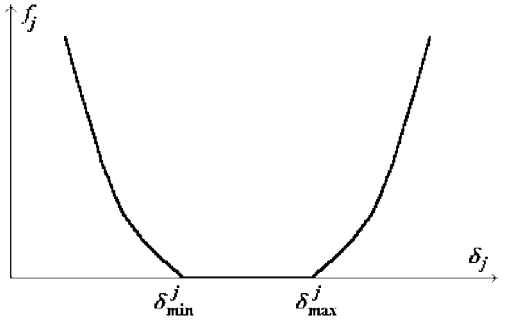


Fig. 1. Dependence of $f_j(\cdot)$ on $\delta_j(X)$.

Then the solving process of problem (2) is reiterated by selecting different combinations of coefficient values $\lambda_j, j = \overline{1,p}$. As a result, we get the points of Pareto set (Zionts, 1989) and the most acceptable of them are selected.

The problem of this type is solved by Dzemyda and Petkus (1998). This is a problem of selection of the optimal nutritive value. The criteria groups are as follows:

1) feed cost price $\varphi(x_1,\dots,x_n) = \sum\limits_{i=1}^{n} x_i k_i$,
2) recommended permissible maximal and minimal violation of the requirements $\Psi_j(x_1,\dots,x_n), j = \overline{1,m}$:

$$\Psi_j(x_1,\dots,x_n) = \\ = \begin{cases} 0, & \text{if } \quad R^j_{\min} \leqslant \sum\limits_{i=1}^{n} x_i A_{ij}(x_l,\dots,x_n) \leqslant R^j_{\max} \\ \sum\limits_{i=1}^{n} x_i A_{ij}(x_l,\dots,x_n) - R^j_{\max}, & \text{if } \quad \sum\limits_{i=1}^{n} x_i A_{ij}(x_l,\dots,x_n) - R^j_{\max} > 0 \\ R^j_{\min} - \sum\limits_{i=1}^{n} x_i A_{ij}(x_l,\dots,x_n), & \text{if } \quad R^j_{\min} - \sum\limits_{i=1}^{n} x_i A_{ij}(x_l,\dots,x_n) > 0 \end{cases}$$

where

$x_i$ is the constituent part of the *i*-th ingredient in feed;

$R^j_{\min}$ is the recommended permissible minimal amount of the $j$-th nutritive characteristics in feed;

$R^j_{\max}$ is the recommended permissible maximal amount of the $j$-th nutritive characteristics in feed;

$A_{ij}$ is a non-linear function, expressing the value of the $j$-th nutritive characteristics of the $i$-th ingredient (some ingredients, starting from the $l$-th one, make an influence on the nutritive characteristics of other ingredients);

$k_i$ is the price of the $i$-th ingredient for a weight unit.

The criteria of the first and the second groups are contradictory – when the violation of permissible amount of nutritive characteristics increases, the price of feed is falling.

Taking into consideration, that the sum of ingredient constituent parts $x_i$ must be equal to 1 and the ratio of the nutritive characteristics affect the feed intake, Dzemyda and Petkus (1998) propose to consider the following equations in the definition domain $\overline{A}$:

$$1)\sum_{i=1}^{n} x_i = 1, \tag{3}$$

$$2)\sum_{i=1}^{n} x_i(A_{it_1^v}(x_l,\dots,x_n) - c_v A_{it_2^v}(x_l,\dots,x_n) = 0, \quad v = \overline{1,w}, \tag{4}$$

where

$c_v$ is the required *v*-th ratio of the nutritive characteristics;

$(t_1^v,\ t_2^v)$ is the *v*-th combination of the nutritive characteristics that are restricted in pairs;

$w$ is the number of combinations of the nutritive characteristics that are restricted in pairs.

The following formulation of the selection of the optimal nutritive value problem is considered by Dzemyda and Petkus (1998):

$$\min_{x_1,\dots,x_n} f(x_1,\dots,x_n) = \min_{x_1,\dots,x_n} \left\{ \sum_{j=1}^{m} r_j \Psi_j^2(x_1,\dots,x_n) + \sum_{i=1}^{n} x_i k_i \right.$$
$$\left. + s\left(\sum_{i=1}^{n} x_i - 1\right)^2 + \sum_{v=1}^{w} s_v \left(\sum_{i=1}^{n} x_i\big(A_{it_1^v}(x_l,\dots,x_n) - c_v A_{it_2^v}(x_l,\dots,x_n)\big)\right)^2 \right\},$$
$$z_{\min}^i \leqslant x_i \leqslant z_{\max}^i,\ i = \overline{1,n}. \tag{5}$$

Here constraints (3) ir (4) are included into the objective function with positive weight coefficients $s$ and $s_v$, $v = \overline{1,w}$. Thus, the problem has been simplified as the definition domain of problem (5) transferred into the rectangular.

In comparison with problem (2),

- the coefficients $r_j$ of problem (5) correspond to the coefficients $\lambda_j$ of problem (2);
- the coefficient at the criterion $\varphi(x_1,\dots,x_n)$ is equal to 1, i.e., $\lambda_{m+1} = 1$,
- $\delta_j(X) = \sum\limits_{i=1}^{n} x_i A_{ij}(x_l,\dots,x_n)$,
- $\delta_{\min}^j$ corresponds to $R_{\min}^j$, and $\delta_{\max}^j$ corresponds to $R_{\max}^j$.

The peculiarity of problem (5) is the use of squared criteria $\Psi_j$ in the objective function. Likewise the transformation of definition domain to rectangular one, it broadens the scale of potential local optimization algorithms.

Dzemyda and Petkus (1998) suggested to solve problem (5) by using variable metric algorithm realized by Tiešis (1975) and presented in the optimization software MINIMUM (Dzemyda, 1985). The values of the coefficients $s$ and $s_v$, $v = \overline{1,w}$, have been fixed relatively large. Selecting different values of the coefficients $r_j$ as well as the initial values of the argument $X = (x_1, \ldots, x_n)$ gives different solutions. The interactive selection of the coefficients $r_j$ enabled Dzemyda and Petkus (1998) to find the solution that satisfies the technological needs.

## 3. Interactive Use of the Computer Network

### 3.1. *The Idea of Interactive Multiple Criteria Optimization*

Computer network offers new methods and tools to solve problem (1). Two possible ways are presented bellow:

1. Parallelization of the optimization algorithm (e.g. variable metric) that is used to solve problem (2) when the values of the weight coefficients $\lambda_j$ are fixed.
2. Interactive decision making on the basis of several solutions of problem (2) obtained by using local optimization by different computers with various values of the coefficients $\lambda_j$.

The shortcoming of the first way – the optimization algorithms are very complicated and there is a great part of individually acquired unpublished authors' knowledge in realization software. Thus, the second way to solve the problem is more effective and hence, it is the basis of this paper, which deals with a concept of interactive multiple criteria optimization. The following interactive optimization method is proposed:

1. Solving the optimization problem with different values of the weight coefficients $\lambda_j$ in parallel: different computers solve the same optimization problem (2) – the values of coefficients $\lambda_j$ differ only. The process is organized by designating the computers as the *master* and the *slaves*. The *master* (P0) controls the process of computer network members, i.e., *slaves* (P1, P2, P3, ...) that execute the *master*'s tasks (see Fig. 2).
2. Visual comparison of the obtained solutions and allocation of new tasks for the computer network. The investigator communicates with the *master* and selects a new combination of weight coefficients $\lambda_j$ for a single optimization problem (2) that will be allocated by the *master* to one of the *slaves*.

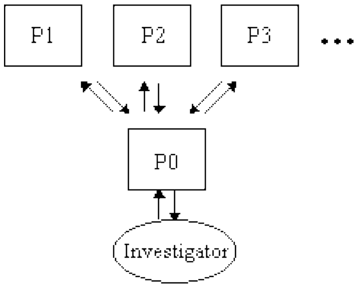


Fig. 2. The structure of interactive analysis.

The investigator starts up by organizing all the computers to solve the problem with various initial data. The computing time of single optimization problem (2) is $t_v = t_f + t_s + t_\varepsilon$; where $t_f$ – task formation time, $t_s$ – task computing time, $t_\varepsilon$ – data transfer and other expenditure.

Let $t_\varepsilon << t_f$ and $t_\varepsilon << t_s$. Hence, $t_\varepsilon$ may be ignored and $t_v = t_f + t_s$ is analyzed. Here the inequality $t_f < t_s$ should be valid, otherwise, the problem may be computed by one computer. Task computing time $t_s$ is constant, i.e., we assume that it is not affected by the problem parameters.

Task formation time $t_f$ is variable. The formation of the first task takes a lot of time for the investigator at the initial point of the work. The following tasks are formed quickly until the first result from the network is received. Subsequently $t_f$ settles upon a certain value depending on the computer number. The settled value also depends on the investigator's experience and $t_s$.

Let us analyze the settled regime to define a sufficient number of computers. In this case, we will follow an assumption that $t_f$ and $t_s$ are constant. Let $K$ be a sufficient number of computers. This number may be defined as follows:

$$K = \frac{t_s}{t_f}. \tag{6}$$

Let us analyze an example in Fig. 3. Consider the settled case. The sufficient number of computers for the problem solving is three – the computers will not be idle and the investigator will not stand idle, too. In common case, the sufficient minimal computer number should not cause idle time for the investigator, and computers may be idle. The idle time is equal to $Kt_f - t_s$:

- if $Kt_f - t_s = 0$, the idle time (both of the computers and the investigator) is equal to zero (this situation is illustrated in Fig. 3);
- if $Kt_f - t_s < 0$, the investigator will stand idle;
- if $Kt_f - t_s > 0$, the computers will be idle.

Hence, if $t_f \geqslant t_s$, then $K = 1$, i.e., one computer is sufficient for the calculations.

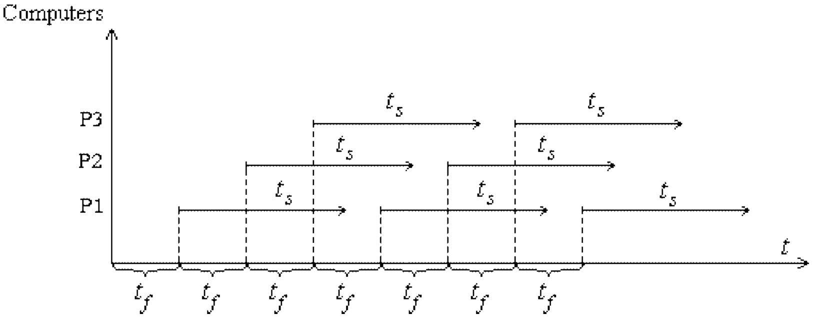


Fig. 3. The efficiency conditions of the computer network application.

We need to decide what to visualize for the interactive decision making. As it was mentioned in Section 2, some functions $f_j(X), j = \overline{1,m}$, $(m \leqslant p)$ are dependent on other functions $\delta_j(X)$. The additional information, that can assist in decision making, is the location of $\delta_j(X)$ with respect to the centre of interval $[\delta^j_{\min}, \delta^j_{\max}]$. When the investigator receives this information, he can intuitively draw a relation between the selected values of coefficients $\lambda_j, j = \overline{1,m}$, and the obtained values of $\delta_j(X) j = \overline{1,m}$. However, the problem of visualization of the data on remaining criteria $f_j(X)$ and their weight coefficients $\lambda_j$, $j = \overline{m+1,p}$, needs an additional analysis when $m < p$. In application on selection of the optimal nutritive value below, we had $p = m + 1$. In this case we fixed $\lambda_p = 1$ and presented visually the numerical values of $f_p(X)$. But such an approach requires deeper investigations for the case when $p > m + 1$. These investigations are out of the scope of our paper.

Several strategies of interactive multiple criteria optimization applying a computer network have been investigated below. The strategies differ in possibility spectrum for the investigator. The analysis should answer the question, how the possibilities obtained affect the investigator's decisions.

*Basic strategy.* Multiple criteria optimization problems are solved interactively providing the investigator with the following means:

- "Dialogue with the investigator module",
- "Connection with the computer network module",
- Complementary memory: lists of tasks, not reviewed results, and the investigator's memorized results.

Fig. 4 presents the strategy scheme in detail. The strategy partially merges with the software used for realization peculiarities. P0 denotes the investigator's computer – *the master*, and P1, P2, P3,. . . label other computers of the network designated as *the slaves*. PS labels the special process that is described bellow and that is used for synchronization of master's performance. This process may be allocated to a separate computer, the master's computer or the slave's computer. A separate computer was used for PS in our experiments.

"Dialogue with the investigator module" communicates with the investigator. The module in its working area allows the investigator to be supplemented by: the number of computers that is necessary for the problem calculation, the number of tasks that have not been started to solve, the number of not reviewed solutions, which the investigator has already received, visual presentation of solutions, the investigator's working means, comfortable controlling by a keyboard. Taking into the consideration the data presented on the screen, the investigator may plan the process of problem solving. At the initial stage of the work, when the investigator is aware of the number of computers applied, he/she may form tasks. The algorithm for solving a single optimization problem (2) is known by the slaves beforehand. Thus, it is enough for the investigator to select the weight coefficients $\lambda_j$. The formed task is allocated to the slaves.

"Connection with the network module" mediates between "Dialogue with the investigator module" and the network. It corresponds to a dispatcher of the task performance.

In the presence of free slaves, “Connection with the network module” assigns weight coefficients to one of the free computers using the procedure of MPI (Message Passing Interface) package software for the data passing. In the absence of free slaves, the task is enlisted in “List of tasks”. After the task assignment, “Connection with the network module” has been waiting for messages both from “Dialogue with the investigator module” and the computer network. However, “Connection with the network module” is set up to hold up its process, until a message from any of the computers is received. In the process holding of “Connection with the computer network module” no data of “Dialogue with the investigator module” is verified and the investigator is disable to carry out any of the operations that are being passed using “Connection with the network module”. In this case, in order to avoid time expenditure, a special – the first process is applied (in Fig. 4 the process is denoted by P1).

Before master’s “Connection with the computer network module” starts waiting for the results from computer network, it passes a message to the first process P1. The process P1 is waiting for any message since beginning of the work. It returns the message back to the master if the message denotes no end of the work. Henceforth, the master defines the process from which the data are received. In the case of receiving data from the first process, “Connection with the network module” verifies whether the investigator is allocating a new task. In the absence of a new task, “Connection with the network

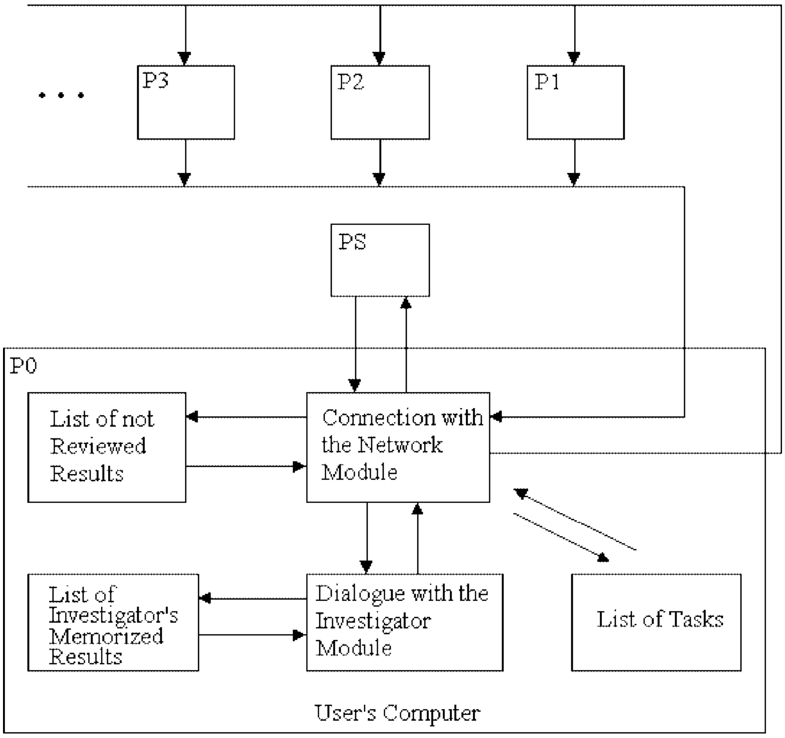

. . .
P3
P2
P1
PS
P0
List of not Reviewed Results
Connection with the Network Module
List of Investigator's Memorized Results
Dialogue with the Investigator Module
List of Tasks
User's Computer

Fig. 4. Flowchart of the basic strategy.

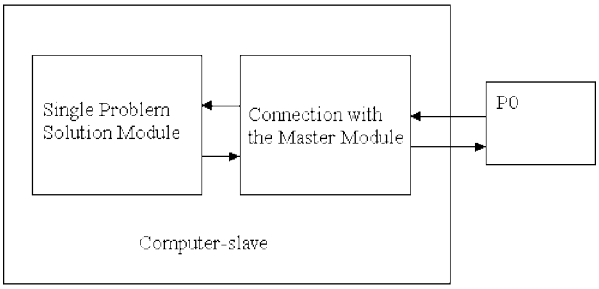

Single Problem Solution Module
Connection with the Master Module
P0
Computer-slave

Fig. 5. Performance of computer-slave.

module" passes a message to the first process and the procedure is iterated till results from the slaves are updated.

As soon as weight coefficients are received by a slave's "Connection with the master module", an operation of "Single problem solution module" starts (see Fig. 5). This module solves the optimization problem (2) with the weight coefficients that have been received. In consequence of the solved problem, it transfers the result to "Connection with the master module" that passes the results to the master.

The results, after returning to "Connection with the network module", are enlisted in "List of not reviewed results". In the presence of any formed task in "Task list", "Connection with the network module" transfers it to a free computer. Then "Dialogue with investigator module" informs about the number of tasks in "Task list" and the number of solutions in "List of not reviewed results". At any moment the investigator may take any of the results out of "List of not reviewed results". Then he/she is able to appraise the solution and to record it into "List of the investigator's memorized results". The memorized solutions may be reviewed by the investigator, selected and used for the further editing.

The problem computation strategy designed in this way enables the multiple criteria optimization problem (1) to be solved interactively applying any number of computers.

*Simplified strategy.* This and the basic strategies differ in possibilities supplied for the investigator. The essential difference is based on the limited list of memorized results presented for the investigator. The simplified strategy suggests the list restricted by a single solution and hence, the investigator is permitted to memorize only one solution that is the best, as far as he/she is concerned.

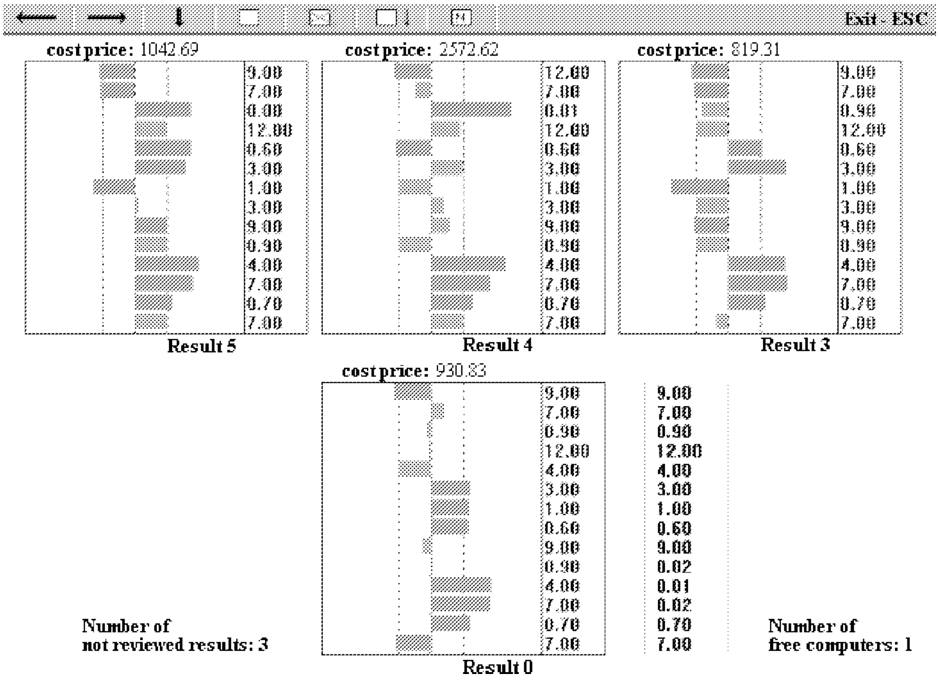
### 3.2. *Computer Software and Hardware*

The multiple criteria optimization problem (1) has been solved by using computer network under a software package MPI (Message Passing Interface) (see Internet publications PHPCR (2000) and MPI (1998), and the book by Snir and co-authors (1996)). The package permits separate computers to compose a single parallel computer. MPI communication and subprogram library executes application programs written in FORTRAN, C and C++. The software package supplies the functions to automatically start up tasks on the virtual machine and allows the tasks to communicate and synchronize with each other. In our case, the virtual machine is composed of Pentium II computers connected to the local Windows NT network. The optimization problem (2) has been solved for different values of weights of criteria using a variable metrics algorithm from the optimization software package MINIMUM (Dzemyda, 1985; Tiešis 1975).

A special graphic interface has been designed for the solution of multiple criteria optimization problem in accordance with the selected calculation strategy. It has been designed applying Microsoft Visual C++ version 6.0 integrated design interface. The method of parallel problem computation and data interchange package MPI caused the program means to design a graphic interface. In our case, the data interchange is performed by MPI 1.3 package that permits to compute programs written in Microsoft Visual C++ version 6.0.

### 3.3. *Special Software*

The investigator's participation is essential in solving interactively a multiple criteria optimization problem. His/her working skills along with formal and informal information obtained about the computed problem affect the calculation process. Thus, application of "MFC Wizard Application" project has enabled us to design user's interface that facilitates his/her work and permits to review the results and to plan the process of calculation (see Fig. 6).



costprice: 1042.69
9.00
7.00
0.00
12.00
0.60
3.00
1.00
3.00
9.00
0.90
4.00
7.00
0.70
7.00
Result 5

costprice: 2572.62
12.00
7.00
0.01
12.00
0.60
3.00
1.00
3.00
9.00
0.90
4.00
7.00
0.70
7.00
Result 4

costprice: 819.31
9.00
7.00
0.90
12.00
0.60
3.00
1.00
3.00
9.00
0.90
4.00
7.00
0.70
7.00
Result 3

costprice: 930.83
9.00
7.00
0.90
12.00
4.00
3.00
1.00
0.60
9.00
0.90
4.00
7.00
0.70
7.00
Result 0

9.00
7.00
0.90
12.00
4.00
3.00
1.00
0.60
9.00
0.02
0.01
0.02
0.70
7.00

Number of
not reviewed results: 3

Number of
free computers: 1

Fig. 6. User's environment.

Fig. 6 presents the graphic interface of a concrete problem solving (it is designed according to the specificity of the problem). The problem of the optimal selection of the nutritive value (Dzemyda and Petkus, 1998) has been presented. The objective of the problem is to minimize farmers' expenditure on nutrition by the optimal selection of feed ingredients in cattle-breeding. The cost price must be minimized in order to meet the necessary requirements of the nutritive value. It is taken into the consideration that animal diets consist of different ingredients on the one hand, and every ingredient varies in different nutritive characteristics on the other hand. Here the cost price is one of major criteria. The rest of 14 criteria are the squared levels of permissible minimal and maximal norm violations. In Fig. 6 the windows show results of a single problem solving in the interface. The left dotted line denotes permissible minimal level and the right one – permissible maximal level. 14 horizontal columns present middle values of the corresponding nutritive characteristic violations of the interval of the requirements. In the right side of the window, weight coefficients of the criteria are located.

The top of the screen presents the solutions that have been received and memorized up to the moment. The windows display only three memorized solutions, nevertheless, it is possible to review and use for further editing any other memorized solution with the help of toolbar buttons or the cursor keys. It has been foreseen the possibility to delete any of the memorized solution out of a solution cluster that is improper and therefore, needless of editing as far as the investigator is concerned.

The bottom of the screen displays the last received or edited solution. A small window nearby is designed for the substitution of weight coefficients $\lambda_j$. On the bottom right side of the screen the number of free computers or the number of formed, but uncalculated tasks in the absence of free computers has been updated at any moment of the process. The number of received and not reviewed solutions has been displayed on the left side of the screen. In the case, when the recent solution more or less satisfies the investigator, it may be memorized and, in the case of necessity, compared to others after some time. One of the calculated criteria of the problem has been displayed above every window with the results. This criterion – the cost price along with the diagram that represents the results – allows the investigator to predict about acceptability of the solution.

Simpler example of a special graphic environment that provides rapid and visually acceptable results is given in Fig. 7. Simultaneous interface of multiple results (Fig. 7, top windows) and an interactive replacement of the initial data (Fig. 7, bottom window) is permissible. Thus, the process of solving the problem is facilitated. The example bellow is an obvious illustration of relatively great violations of the recommended permissible minimal and maximal amounts of the nutritive characteristics in feed at a lower price. In this case, the violations of the recommendations decrease when the price is rising.

## 4. Efficiency Investigation

The experimental investigation has been carried out on the basis of designed basic and simplified strategies for multiple criteria optimization problems to be solved interactively

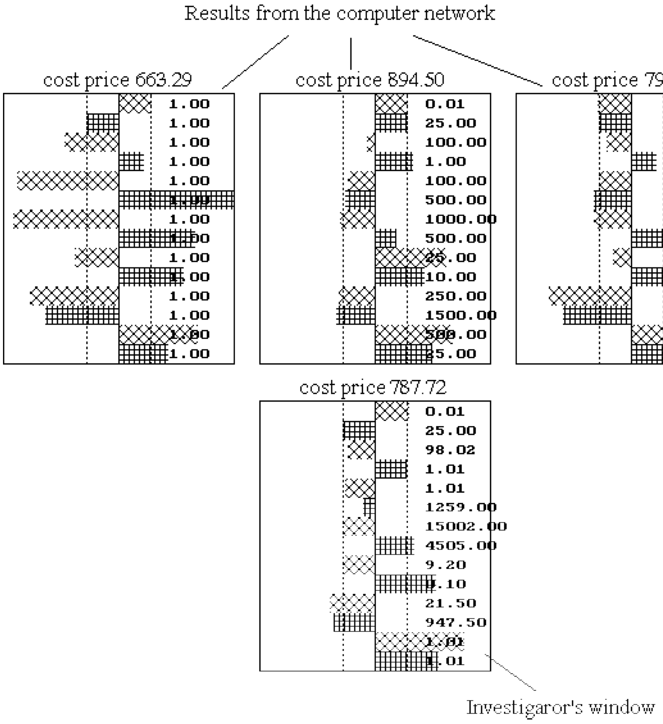


Fig. 7. The window of an interactive comparison of the solutions.

by using computer network. A selection of optimal nutritive value problem has been investigated and a random permissible minimal and maximal norm violation levels have been generated. Different multiple criteria optimization problems applying two, four and six computers-slaves have been calculated for the cases of one or two minutes for a single problem (2) solution (time expenditure has been artificially prolonged until the desired duration). It has been sought to detect experimentally the time spent for calculation from the initial stage of the process up to the best solution to the investigator's mind, and the effect of the number of computers on computing time.

Every trial of the investigation has been iterated for ten times. A problem from the trial has been solved at least 30 minutes. Every next iterated experiment has been recorded: the values of combined criterion, that includes requirement violations and the cost price, have been fixed every minute since the zero time moment. Then ten realizations have made the criterion values mean. Values of the combined criterion are obtained from the formula $V_i = \sqrt{K_i^2 + S_i^2}$, $i$ – the time moment, $K_i$ – normalized cost price, $S_i$ – normalized sum violations of the requirements. The values of $S_i$ are located to the interval [0; 1] by the formula $\frac{S_i - a}{b - a}$, $a$ – the least sum violation obtained, $b$ – the greatest sum violation obtained, $S_i$ – sum violations of the requirements. The values of $K_i$ are calculated in analogous way with $S_i$.

Fig. 8 represents the integral results obtained by using the basic strategy when the

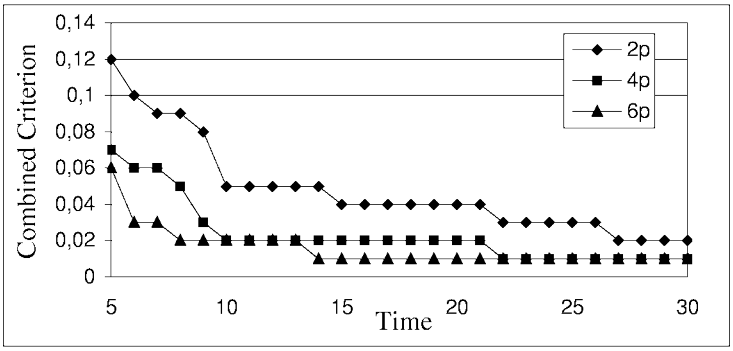


Fig. 8. Dependence of efficiency of the basic strategy on the number of computers $p$.

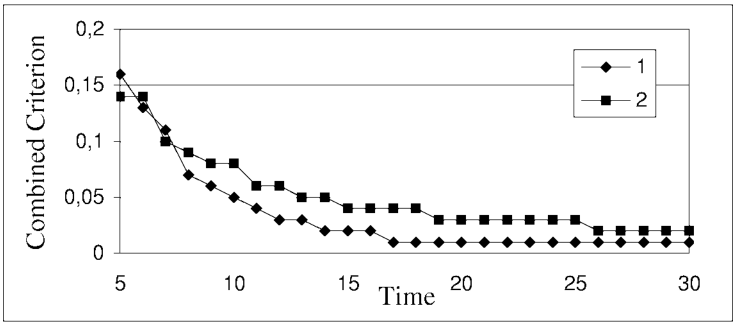


Fig. 9. Comparison of strategies: basic (1) and simplified (2).

computing time for single solving of problem (2) was one minute. Two, four and six computers-slaves were applied. The investigation has proved that transition from two computers to four ones is more effective than from four to six.

Fig. 9 represents the results of calculations based on the basic and the simplified strategies applying six computers-slaves and the computing time of one slave's problem equal to two minutes. The basic strategy is obviously prior the simplified one, though they are almost equivalent during the first minutes of the calculation.

## 5. Conclusions

An interactive optimization method by using computer network has been proposed. The advantages of the method are as follows:

1. The investigator participates actively in the optimization process.
2. The investigator discovers a new knowledge of the problem in the process of calculation. Thus, it may result a new and more efficient optimization strategy to be designed or applied.

3. The problem is solved by more than one computer. Thus, the computer time for optimal solution may be economized.

The investigation presents two strategies and the experimental analysis on the basis of them, which let us detect the effect of computer resources power supported to the user on the interactive solution of multiple criteria optimization problems applying a computer network. Single problems of different calculation time in the computer network, applying different number of computers, have been investigated. The numerous data of the investigation allow to draw the conclusions:

- the increase in number of computers applied for interactive calculation of multiple criteria optimization problems improves the effect of the problem solution;
- it is inexpedient to apply a great number of computers in solution of a single problem that requires little expenditure of computing time, because most of computers-slaves will be waiting for tasks, and the main charge of time will take the investigator for his/her decisions;
- it should be taken into account the calculation time for a single problem along with personal features and working skills of the investigator when selecting the number of computers;
- the designed strategies may be applied for large multiple criteria optimization problems to be solved interactively;
- in comparison with the simplified strategy, the basic strategy provides more possibilities for the investigator, and that enables the problem to be solved with less expenditure of computing time.

In further investigations we will estimate experimentally the efficiency of the computer network application depending on the simultaneous communication of multiple investigators by using a common computer network.

## References

Dzemyda, G. (Ed.) (1985). *The Package of Applied Programs for Dialogue Solving of Multiextremal Problems MINIMUM: A Description of Using*. The State Fund of Algorithms and Programs (reg.No50860000112), Inst.Math.Cybern., Vilnius (in Russian).

Dzemyda, G., T. Petkus (1998). Selection of the optimal nutritive value. In *Proceedings of the Special IFORS Conference "Organizational Structures, Management, Simulation of Business Sectors and Systems"*. Technologija, Kaunas. pp. 73–77.

Karpak, B., S. Zionts (Eds.) (1989). *Multiple Criteria Decision Making and Risk Analysis Using Microcomputers. NATO ASI Series, Series F: Computer and Systems Sciences*, Vol. 56. Springer.

MPI (1998). *MPI for Windows NT – Documentation*. http://www-jics.cs.utk.edu/mpint/.

PHPCR (2000). *Parallel and High Performance Computing Resources*.
http://www-jics.cs.utk.edu/parallel.html.

Snir, M., S. Otto, S. Huss-Lederman, D. Walker, J. Dongarra (1996). *MPI: The Complete Reference*. Massachusetts Institute of Technology.

Tiešis, V. (1975). The variable metric method for local minimization of multivariate functions subject to simple bounds. In R.Chomskis (Ed.), *Vyczislitelnaja Technika, Proceedings of the Conference*. Kaunas Polytechnic Institute, Kaunas. pp. 111–114 (in Russian).

Zionts, S. (1989). Multiple criteria mathematical programming: an updated overview and several approaches. In B. Karpak and S. Zionts (Eds.), *Multiple Criteria Decision Making and Risk Analysis Using Microcomputers. NATO ASI Series, Series F: Computer and Systems Sciences*, Vol. 56. Springer. pp. 7–60.

**G. Dzemyda** graduated from Kaunas University of Technology, Lithuania, in 1980, and in 1984 received there the doctoral degree in technical sciences (Ph.D.) after post-graduate studies at the Institute of Mathematics and Informatics, Vilnius, Lithuania. In 1997 he received the degree of Doctor Habilius from Kaunas University of Technology. He was conferred the title of Professor (1998) at Kaunas University of Technology. Recent employment is at the Optimization Department of the Institute of Mathematics and Informatics as a Principal Researcher. The main field of scientific interests is the interaction of optimization and data analysis. The interests include optimization theory and applications, multiple criteria decisions, neural networks, and data analysis.

**T. Petkus** graduated from Vilnius Pedagogical University, Lithuania, in 1995, MA of informatics in 1996. Recently he is a doctoral student and a lecturer in the Department of Informatics of Vilnius Pedagogical University. Scientific interests cover parallel computing and multiple criteria decisions.

# Kompiuterių tinklo taikymas sudėtingiems optimizavimo uždaviniams spręsti

Gintautas DZEMYDA, Tomas PETKUS

Straipsnyje nagrinėjamas metodas interaktyviai spręsti daugiakriterinius optimizavimo uždavinius panaudojant kompiuterių tinklą. Pasiūlyto metodo esmė glūdi ne kokio nors optimizavimo metodo išlygiagretinime, bet konkrečių optimizavimo užduočių paskirstyme tinklo kompiuteriams. Optimizuojama tikslo funkcija yra kriterijų svorinė suma. Daugiakriterinis optimizavimo uždavinys sprendžiamas daug kartų, interaktyviai parenkant skirtingus kriterijų svorių koeficientus. Tuo būdu tinkle darbas organizuojamas "šeimininko" (kompiuterio kuris koordinuoja likusiųjų kompiuterių darbą) ir "darbininkų" (kompiuterių, kurie atlieka jiems pavestą darbo dalį) principu. Pradžioje tyrinėtojas siunčia į tinklą tam tikrą kiekį optimizavimo uždavinių. Tai tokie patys tikslo funkcijos optimizavimo uždaviniai, tik su skirtingais kriterijų svorio koeficientais. Kai "darbininkas" baigia savo darbo dalį, siunčia rezultatus "šeimininkui". Taip elgiasi kiekvienas tinko kompiuteris. Tyrinėtojas gavęs eilinius kažkurio tai kompiuterio rezultatus, atsižvelgdamas į juos ir į prieš tai buvusius, priima naują sprendimą, t.y. nustato naujus svorio koeficientus kriterijams ir siunčia naują užduotį į tinklą. Taip daugiakriterinis uždavinys sprendžiamas tol, kol tyrinėtoją tenkina gauti rezultatai. Autorių tikslas buvo sukurti ir ištirti interaktyvias strategijas didelės apimties (vienkartinio sprendimo laiko atžvilgiu) daugiakriteriniams uždaviniams spręsti panaudojant kompiuterių tinklus. Buvo panaudotas lygiagrečių skaičiavimų paketas MPI (Message Passing Interface). Bandymai atlikti su operacinės sistemos Windows NT valdomu kompiuterių tinklu. Tikslo funkcijai optimizuoti esant fiksuotiems kriterijų svoriams buvo panaudotas programų paketas "Minimum" bei autorių sukurta programinė įranga. Pasiūlyto metodo veikimas iliustruojamas optimalios pašarų sudėties parinkimo uždavinio pagrindu.