ISSN 1392 – 124X INFORMATION TECHNOLOGY AND CONTROL, 2008, Vol.37, No.1

DECISION MAKING TO SOLVE MULTIPLE CRITERIA OPTIMIZATION PROBLEMS IN COMPUTER NETWORKS

Tomas Petkus[1], Ernestas Filatovas[2]

[1]*Vilnius Pedagogical University, Studentu str. 39, LT-08106 Vilnius, Lithuania*

[2]*Institute of Mathematics and Informatics, Akademijos str. 4, LT-08663 Vilnius, Lithuania*

Abstract. The paper presents two new parallel solution strategies to solve complex multiple criteria problems by applying a computer network. The multiple criteria problem is iterated by selecting interactively different weight coefficients of the criteria. The process is organized by designating the computers as *the master* and *the slaves*. The experimental trials have been carried out to compare the effect of the designed strategies depending on the number of computers and experience of the researcher solving this optimization problem. The essence of the investigation is distribution of jobs between the researcher and computer.

Keywords: multiple criteria optimization, parallel computing, MPI.

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.

Computer networks are widespread and permit us to solve complex optimization problems by using simple computers. Furthermore, the networks enable us to solve considerably more complex problems by using the aggregate power of many computers.

The methods for interactive solving a complex multiple criteria optimization problem by using computer network are analyzed. The investigation in this paper presents two interactive strategies and the experimental analysis on their basis, which lets us detect the effect of computer aid to the user on the interactive solution of multiple criteria optimization problems, applying a computer network.

2. Statement of the Optimization Problem

Let us analyze a multiple criteria optimization problem:

$$\min_{X=(x_1,\ldots,x_n)\in\bar{A}} f_j(X), j=\overline{1,\mu}, \quad (1)$$

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

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

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

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

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

$$\min_{X=(x_1,\ldots,x_n)\in\bar{A}} \sum_{j=1}^{\mu}\lambda_j f_j(X) \quad (2)$$

Then the solving process of problem (2) is reiterated by selecting different combinations of coefficient values $\lambda_j, j=\overline{1,\mu}$. Many solutions are obtained from the computer network and they are points of Pareto. The most acceptable of them are selected. The problem of this type was solved by Dzemyda and Petkus 0. This was a problem of selection of the optimal nutritive value.

3. Interactive Use of the Computer Network

3.1. The idea of interactive multiple criteria optimization

The scheme (generalized) of algorithm to solve the optimization problem is presented in Figure 1.

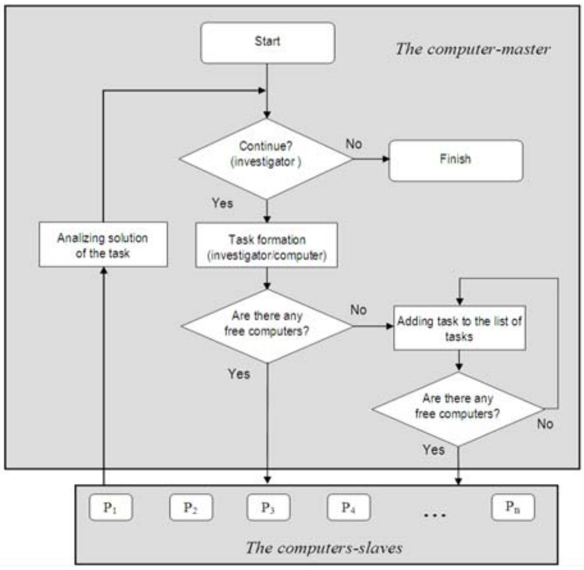

Figure 1. The scheme to solve the optimization problem

The great number of computers-slaves enables us to design new strategies that allow us to form many tasks and send them to the computer network. A special memory (list of tasks) is realized to memorize new tasks, if all the computers-slaves are busy.

Solution time of the multiple criteria optimization problem depends on the investigator's attitude. The investigator decides when to stop the solving process according to his opinion.

3.2. Analyzed strategies

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

Basic strategy. Tasks for the computer network are formed only by the investigator (this strategy was presented in previous works) 0.

First strategy. Tasks for the computer network are formed only by the computer-master. The computer-master generates all tasks for the computer network: starting tasks, further tasks that depend on the obtained solutions. The investigator does not form any tasks for the computer network. He decides when a solution is acceptable.

Second strategy. Tasks for the computer network are formed by combining the investigator and the computer-master. The computer-master generates initial tasks. The investigator forms new tasks, using the obtained solutions and his own experience. The computer-master generates new tasks if the investigator was late to do that for the computer network. The weight coefficients are generated with regard to the last investigator's decisions on selecting the starting point of a task.

3.3. Peculiarity of the strategies

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

Let $t_\varepsilon << t_s$. Hence, t_ε 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 solved by one computer. The task computing time t_s is constant, i. e., we assume that it is not affected by the problem parameters.

The task formation time t_f is variable. The creation of the first task takes up a lot of time for the investigator at the initial point of the work. The following tasks are rapidly formed until the first result from the network has been received. Subsequently t_f settles upon a certain value depending on the number of computers in the network. 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 denote a sufficient number of computers. This number may be defined as follows:

$$K = \frac{t_s}{t_f}. \quad (3)$$

The sufficient number of computers for the basic strategy is three (Figure 2) – the computers will not be idle and the investigator will not stand idle, too. In the general 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 Figure 2);
- if $Kt'_f - t_s < 0$, the investigator will stand idle;
- if $Kt'_f - t_s > 0$, the computers will be idle;
- where t'_f is the task formation time in the first strategy.

Hence, if $t'_f \geq t_s$, then $K = 1$, i. e. one computer is sufficient for the calculations.

If the multiple criteria optimization problem is solved by the first or the second strategies, the situation is different. The computer-master forms tasks much faster than the investigator ($t''_f << t'_f$) and many computers can be used in the computer network: they will not be idle (this situation is illustrated in Figure 3), where t''_f is the task formation time in the second strategy.

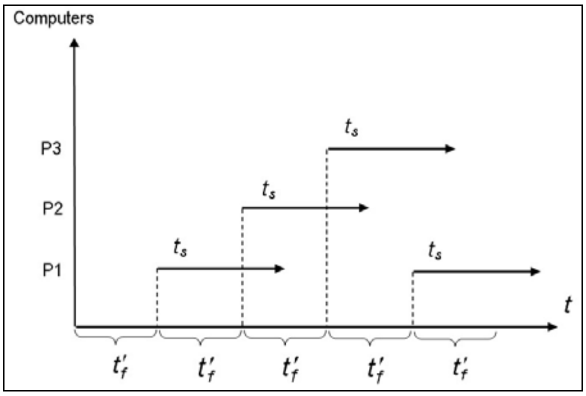

Figure 2. Efficiency conditions of the basic strategy

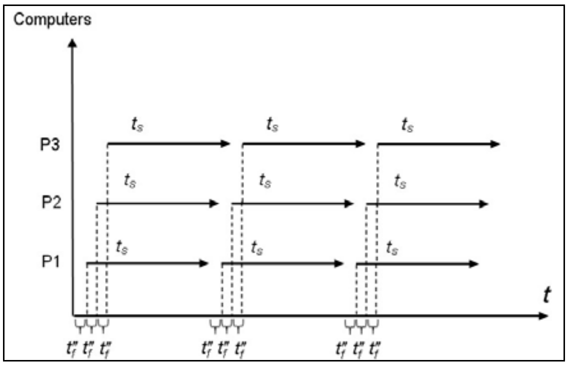

Figure 3. Efficiency conditions of the new strategies

3.4. Computer software and hardware

The multiple criteria optimization problem (1) has been solved by using the computer network with the software package MPI (Message Passing Interface) [4]. The package permits separate computers to compose a single parallel computer. The MPI communication and subprogram library runs the application programs written in FORTRAN, C and C++. The software package supplies with the functions to automatically start up tasks on a virtual machine and allows the tasks to communicate and synchronize with one another. In our case, the virtual machine is composed of 26 computers (Pentium 4, 3.2 GHz) connected to the local Windows XP network (1 Gbps). The optimization problem (2) has been solved with different values of the weights of criteria, using a variable metrics algorithm from the optimization software package MINIMUM 0.

A special graphic interface has been designed for solving the multiple criteria optimization problem in accordance with the selected calculation strategies 0. Now it has been developed for the new strategies. The special graphic interface 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 MPICH v.1.2.5 package that allows us to compute programs realized by Microsoft Visual C++ version 6.0 **Error! Reference source not found.**.

4. Experimental investigation

The experimental investigation has been carried out on the basis of the designed basic, first and second strategies for multiple criteria optimization problems to be solved interactively by applying a computer network. Selection of the optimal nutritive value problem has been investigated and random permissible minimal and maximal norm violation levels have been generated. Different multiple criteria optimization problems applying 1, 6, 12, 24 computers-slaves have been solved where time expenditure for a single problem (2) was one minute. The objectives of investigation were to detect:

- the effect of the number of computers applied in the process;
- the comparative efficiency of different strategies depending on the number of computers applied in the calculations.

Each trial of the investigation has been iterated for ten times (10 realizations). Each problem of the trial has been solved for at least 30 minutes. Each next iterated experiment has been recorded: the values of a 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 up the average of the criterion values.

The combined criterion values were obtained by the formula $V_i = \sqrt{K_i^2 + S_i^2}$, where i is the time moment, K_i is the normalized cost price, S_i is the normalized sum of violations of the requirements. The values of S_i were arranged in the interval [0; 1] by the formula $\frac{S_i - a}{b - a}$, where a is violation of the least obtained sum, b is violation of the greatest obtained sum, S_i is the sum of violations of the requirements. The values of K_i are calculated in an analogous way.

4.1. The dependence of the strategies on the number of computers

Figure 4 represents the results obtained solving the problem (1) by the basic strategy, when the computing time for solving a single problem (2) was 1 minute. One and six computers-slaves were applied. As expected, when solving the problem by six computers-slaves, better results are reached in less time.

Figure 5 illustrates the results obtained solving the problem (1) by the first strategy with one, six, and 24 computers-slaves. An increase in the number of computers provides better results. This tendency remains during all the experiments. An increase in the number of computers-slaves enables us to form more tasks and provides more solutions.

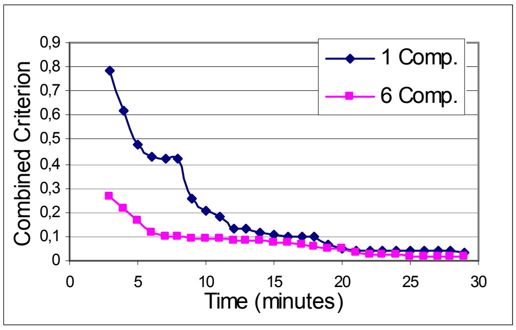

Figure 4. Dependence of the basic strategy on the number of computers

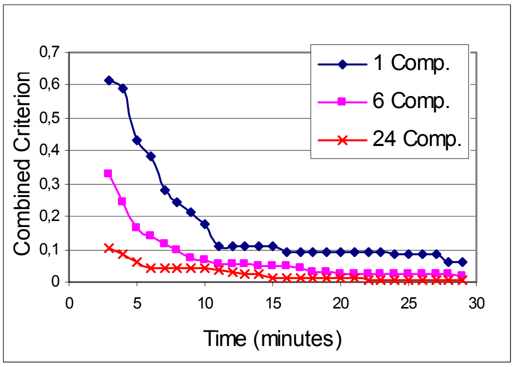

Figure 5. Dependence of the first strategy on the number of computers

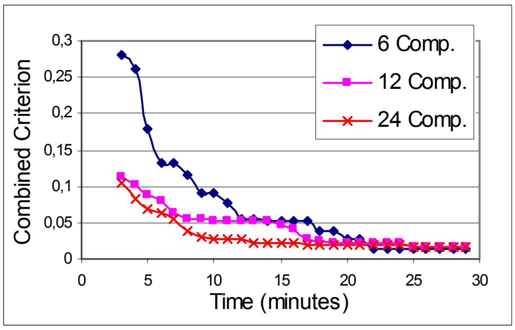

Figure 6. Dependence of the second strategy on the number of computers

Figure 6 illustrates the results solving the problem (1) by the second strategy with 6, 12, and 24 computers-slaves. An increase in the number of computers provides better results faster. But starting the 12-th minute the value of the combined criterion does not change, because the global minimum was reached (or a very close value to it). The combination of the computer and investigator proved the best productivity in case where the optimization problem was solved by the second strategy. The time for obtaining a proper solution depends on the number of computers-slaves (Figure 6).

The comparison of the basic and first strategies is shown in Figures 7 and 8. When 6 computers are employed in both strategies the results are similar. In the basic strategy, the investigator cannot analyze more solutions obtained and select a new proper combination of the weight coefficients λ_j. In this case, tasks formed by the investigator are similar to that as the computer-master. A further increase in the number of computers-slaves with the basic strategy is senseless – the investigator cannot afford working with more computers-slaves. The computer-master generates tasks rather fast; therefore it is able to form tasks for many computers-slaves.

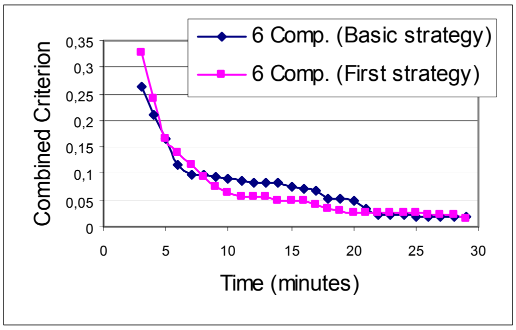

Figure 7. Comparison of the basic and first strategies with six computers

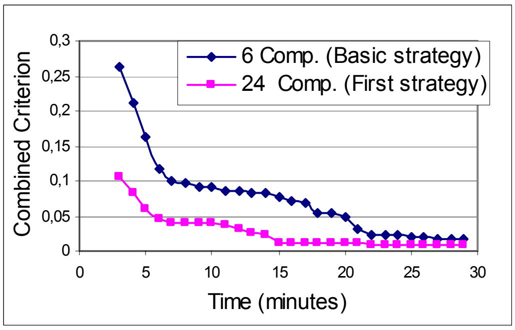

Figure 8. Comparison of the basic and first strategies with 24 computers

The experimental investigation with 24 computers-slaves is illustrated in Figure 8. A great advantage of the first strategy is evident. A proper solution was achieved in 13 minutes using 24 computers-slaves. The same results were obtained by the basic strategy using 6 computers-slaves, only in 18 minutes. Hence, the first strategy is superior when a great number of computers is available.

Figure 9 shows the results of the second strategy, when 12 and 24 computers were employed comparing with the basic strategy when six computers were used. A proper solution was achieved in 18 minutes by the second strategy, using 12 computers-slaves, and in 11 minutes, using 24 computers-slaves. The same results were obtained by the basic strategy using six computers-slaves only in 22 minutes. This experimental investigation has proved an obvious advantage of the second strategy in comparison with the basic strategy.

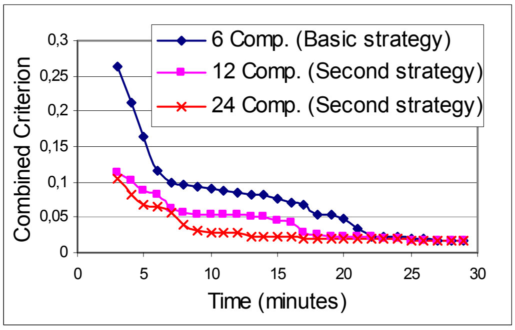

Figure 9. Comparison of the basic and second strategies

The comparison of the first and second strategies is shown in Figures 10 and 11. The number of computers-slaves and investigator's experience in the second strategy influence the results.

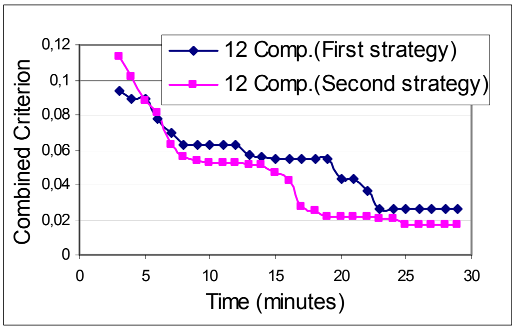

Figure 10. Comparison of the first and second strategies with 12 computers

Figure 10 illustrates the results obtained by the first strategy, and the second strategy with 12 computers-slaves employed. A great advantage of the second strategy in comparison with the basic or first strategies is obvious. This is the strategy when the investigator selects the way of solution of the multiple criteria optimization problem.

Figure 11 shows the results obtained by the first and second strategies when 24 computers-slaves were employed.

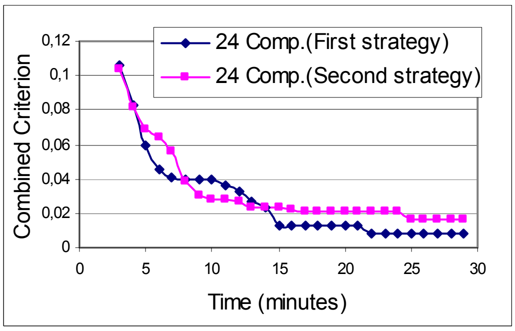

Figure 11. Comparison of the first and second strategies with 24 computers

While analyzing the results, it can be noticed that the investigator's influence on the problem solution is limited. The investigator selects a worse way of optimization problem solution when there are many computers-slaves in the computer network. He spends a lot of time for analyzing the solutions obtained, thus the investigator lacks time to form new tasks.

4.2. Efficiency of the computer network

The computer efficiency in the parallel solution is defined: $E_p = \frac{S_p}{p}$, where $S_p = \frac{T_0}{T_p}$ (speedup), T_0 is the time to compute the problem applying one computer (serial algorithm), T_p is the time to obtain an adequate result applying p computers (parallel algorithm).

The efficiency of the calculation results is presented in an analogical definition: $E = \frac{J_n}{J_m}$, where J_n is the control combined criterion; J_m is the combined criterion investigated.

In this work the efficiency of the calculation was estimated every minute of the process.

Figure 12 shows the efficiency of the computer network, when 12 computers were employed in the process of solving the problem by the first and second strategies. In the second strategy the investigator selects the way of solving of the problem, while the computer-master is very helpful for generating more new tasks. The investigator's influence on the optimization problem solution is greatest when 12 computers-slaves were employed.

Figure 12. Efficiency comparison of new strategies

5. Conclusions

New strategies of the interactive optimization method by using the computer network have been proposed. The investigation presents two developed strategies and the experimental analysis based on them, which enabled us to detect the effect of computer resource power on the interactive solution of multiple criteria optimization problems, applying the computer network. Numerous data of the investigation allow us to draw the next conclusions:

- more than one computer was used to solve the multiple criteria optimization problem; the developed strategies reduce the solution time of the problem, improve the efficiency of the problem solution;

- the new strategies enable one to apply a great number of computers more efficiently as compared with the basic strategy;
- human-computer interaction makes it possible to apply a larger number of computers;
- the number of available computers in solving the problem is important for selecting the strategy.

Acknowledgment

The research is partially 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-33/07)".

References

[1] **G. Dzemyda.** (Ed.) 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*, 1985, 41 (*in Russian*).

[2] **G. Dzemyda, T. Petkus.** Selection of the optimal nutritive value. *IFORS Conference "Organizational Structures, Management, Simulation of Business Sectors and Systems", Technologija, Kaunas*, 1998, 73-77.

[3] **G. Dzemyda, T. Petkus.** Application of Computer Network to Solve the Complex Applied Multiple Criteria Optimization Problems. Informatica, Vilnius, 2001, 45-60.

[4] The Message Passing Interface (MPI) standard. *http://www-unix.mcs.anl.gov/mpi/.*

[5] MPICH-A Portable Implementation of MPI. *http://www-unix.mcs.anl.gov/mpi/mpich*.

Received August 2007.