Multicriterion Evolutionary Algorithm for Workload Balancing of the Web Bank Servers

Jerzy Balicki,[†]*,*

Naval University of Gdynia, ul. Smidowicza 69, 81-103 Gdynia, Poland

Summary
Load balancing of the Internet bank can be implemented by reduction of the workload of the bottleneck Web server. Load balancing improves both a performance of the computer system and the reliability of a server network. An evolutionary algorithm improved by introducing a tabu search procedure is discussed for solving multi-criteria optimization problem of finding a set of Pareto-suboptimal task assignments. A tabu mutation is used for minimization the workload of the bottleneck computer.

Key words:
Evolutionary algorithm, multi-criterion optimization, Internet bank systems.

Introduction

Bank transactions can be parallel performed subject to geographic spread of users. An average cost of a transaction through the Internet is lower several times than the cost of a transaction carried out in the bank with the network of branches [2]. Furthermore, the Internet transaction cost is significantly lower than the transaction cost for the cash machine network or for the financial advise system through a phone network [3]. It is a decisive motivation of the tendency displayed itself in the Web banking expenditure has been doubled each year since 1997 [4]. Besides, computer technology is suitable for implementation of bank transactions [5]. Computer data can be processed, stored and transferred with using the relevant computer network technology.

On the other hand, evolutionary algorithms can inherit some abilities of tabu search techniques to improve a quality of obtained Pareto-suboptimal solutions [12]. A tabu search is the powerful meta-heuristic approach that has been applied for crucial applications in engineering, economics and science [10]. An advanced of a multicriteria evolutionary algorithm with a tabu search has been suggested in [1].

A relevant allocation of program modules in the Web bank network may significantly reduce the entirety time of a program run by taking a benefit of the particular properties of some workstations or an advantage of the balanced computer load. An adaptive evolutionary algorithm has been considered for finding solutions to multiobjective optimization problems related to task assignment that minimize a workload of a bottleneck computer and the cost of machines [1]. The total numerical performance of workstations is another criterion for assessment of task assignment. Additionally, a reliability of the system is a criterion that is significant to assess the quality of a task assignment in bank systems. Consequently, the problem with above criteria and some memory constraints is discussed.

1. Model of Web bank network

Banks use the Internet to support their tasks that are related to fundamental services. It may impact on the acceleration of the service and it permits users to save time. It improves, radically, the comfort for the bank clients because they may require an access to the database via the Internet to view the balance of the account or display a transaction history. On the other hand, some complicated secure procedures have to be introduced and still developed to protect this access to information.

A temporary deposit can be managed remotely what saves time of a client and gives an opportunity to an efficient managing of financial means. Furthermore, a money transfer between interior accounts is permitted for a user. In this way, the differences of interest rates can be respected and an advantage from a short-term save can be taken. Data of a bank transaction history can be exported to a program that supports the financial control of the home budget.

Monitoring of transactions, including those paid by a credit card, gives information about the current rate of payment. Thus, we can reasonably plan the other expenditures. A WWW home page of the bank often includes some procedures for an estimation of an interest of the deposit. It supports making decision about credits as well as deposits. Moreover, the process of required data preparing is subject to shorten. An information service gives the marketing advantage for a bank and it is relevant for users that active manage their financial resources. It is an essential proposal of conventional banks at the beginning of the Internet using.

Manuscript received October 5, 2006.
Manuscript revised October 25, 2006.

These fundamental tasks can be relatively effortless implemented by using the Apache server that is a powerful and flexible web server [1]. It implements the latest protocols, and it is extremely configurable and extensible with other modules. What is more, it runs on crucial operating systems. This server is dynamically being developed and also it encourages user feedback through new ideas, bug reports and patches.

The Apache server implements many frequently requested features, including databases for authentication. It allows setting up password-protected pages with massive numbers of authorized users, without bogging down the server. Customized responses to errors let setting up files, or even some scripts, which are returned by the server in response to errors. It is possible to setting up a script to intercept numerous server errors and perform on-the-fly diagnostics.

A transaction of payment from a bank account requires a more advanced approach related to the filling up a appropriate form in the secured web document. An order of payment is prepared, and then it is send to the destination account. Data from this order are translated by an additional program module to the format of data that are applied in the national inter-bank communication system. A persistent order of payment can be started or cancelled from the user terminal, too. A direct debit is another order of payment made by a client, but a purpose of this transaction is an account of user. Forms with invariable data are stored by the depository system and some most recent parts of prepared form may be altered, only.

Prediction and optimization of payments are an extended bank tasks. A schedule of payment can be found subject to the history of transactions. Prediction of the payments can be made by using artificial intelligence techniques like neural networks or expert systems. The terms of transactions can be set up just before their deadlines as well as cheaper credits can be suggested.

Above tasks are supported by bank servers and client browsers. Advanced tasks make use of the interactivity, flexibility and multimedia property of the Internet, what gives an innovative approach to bank services. To attract clients, banks enrich their web pages by supplementary and contemporary information about exchange rates, tax regulations and stock exchange. Moreover, the high-quality web service is supposed to be included as the preferred page to the browser of users, what gives the opportunity to promote novel products or possibilities of cross selling.

The WWW bank service may recommends selling of the complementary financial products that are not proposed by the bank. Life insurance, pension funds, stockholder services and deposit certificates of the other companies are admissible on the web pages of the bank. In addition, airline tickets, tickets to cinemas, theatres, and sport matches can be sell as well as DVDs, CDs, books or even holidays. That is, a bank ought to create the Internet shop with the relevant catalog of financial and non-financial products. The WWW bank service is supposed to be profiled to adapt its content and an appearance to the personal preferences.

The most leading banks want to take advantage of the virtues of the worldwide distributed system. Because a potential client can apply a cell phone to receive and send the written, voice or multimedia messages, banks use these channels of communication to inform the clients about the state of their accounts. The mobile network and a micro browser can be used in effective way, especially if a mobile phone cooperates with a computer.

Online advising can be implemented by chat and conference software for written questions by a client, and then reading answers during the real-time dialog with a bank expert. A voice and vision communication can be carried out with using modern software.

Payment can be performed by sending an encoded number of a credit card with using the secure protocol. The safety of the transaction increases if the secure electronic transaction is used with the wallet software, the certificate and the accounting center.

2. Reliability of bank software system

Clients of a bank generate requires to a bank computer system and these events are handled by web servers that manage the program modules. A program module in the bank system can be activated several times during the interval of time when the heaviest load occurs. A set of program modules $\{M_1,...,M_m,...,M_M\}$ communicated to each others is considered among the coherent computer network with computers located at the processing nodes from the set $W=\{w_1,...,w_i,...,w_I\}$. In results, a set of program modules is mapped into the set of parallel performing tasks $\{T_1,...,T_v,...,T_V\}$ [18].

Let the task T_v be executed on computers taken from the set of available computer sorts $\Pi=\{\pi_1,...,\pi_j,...,\pi_J\}$. The overhead performing time of the task T_v by the computer π_j is represented by t_{vj}. Let a computer π_j be failed independently due to an exponential distribution with rate λ_j. The longer time of task execution, the higher probability of computer failure. We do not take into account of repair and recovery times for failed computer in assessing the logical correctness of an allocation. Instead, we shall allocate tasks to computers on which failures are least likely to occur during the execution of tasks. Computers can be allocated to nodes and also tasks can be assigned to them in purpose to maximize the reliability function R defined, as below:

$$R(x) = \prod_{v=1}^{V} \prod_{i=1}^{I} \prod_{j=1}^{J} \exp(-\lambda_j t_{vj} x_{vi}^m x_{ij}^\pi), \qquad (1)$$

where

$$x_{ij}^\pi = \begin{cases} 1 \text{ if } \pi_j \text{ is assigned to the } w_i, \\ 0 \text{ in the other case.} \end{cases}$$

$$x_{vi}^m = \begin{cases} 1 \text{ if task } T_v \text{ is assigned to } w_i, \\ 0 \text{ in the other case,} \end{cases}$$

$$x = [x_{11}^m, \dots, x_{1I}^m, \dots, x_{vi}^m, \dots, x_{VI}^m, x_{11}^\pi, \dots, x_{1J}^\pi, \dots, x_{ij}^\pi, \dots, x_{Ij}^\pi, \dots, x_{IJ}^\pi]^T.$$

A computer can be chosen several times from the set Π to be assigned to the node and one computer is allocated to each node. On the other hand, each task is allocated to any node.

3. Workload of the bottleneck computer

The cost of the parallel program performing is the most common used measure of an allowance evaluation. If the number of computers is greater than 3 or the memory in a computer is limited, then a problem of the program completion cost minimization by task assignment is NP-hard [1]. The workload of the bottleneck computer is another fundamental criterion for the evaluation of an allocation quality. A computer with the heaviest task load is the bottleneck machine, and its workload is a critical value that is supposed to be minimized [6]. The workload $Z_i^+(x)$ of a computer allotted to the ith node for the allocation x is provided, as follows:

$$Z_i^+(x) = \sum_{j=1}^{J} \sum_{v=1}^{V} t_{vj} x_{vi}^m x_{ij}^\pi + \sum_{v=1}^{V} \sum_{\substack{u=1 \\ u \neq v}}^{V} \sum_{\substack{i_2=1 \\ i_2 \neq i}}^{I} \tau_{vu} x_{vi}^m x_{ui_2}^m, \qquad (2)$$

where τ_{vu} – the total communication time between the task T_v and T_u.

The workload $Z_i^+(x)$ of the bottleneck machine in the system is the critical value that should be minimized:

$$Z_{\max}(x) = \min_{i=\overline{1I}} \left\{ Z_i^+(x) \right\} \qquad (3)$$

An optimal task allocation for the cost of the parallel program performing does not guarantee the load stability on computers in some assignments, because the workstation with the heaviest load might possess a heavier workload than another bottleneck machine for the other task allocation in the distributed system. The workload of the bottleneck computer can be employed as an assessment measure of an allotment quality in systems, where the minimization of a response time is required, too.

Each computer ought to be equipped with required capacities of resources for a program run. Let the following memories $z_1, \dots, z_r, \dots, z_R$ be available in an entire system and let d_{jr} be denote the capacity of memory z_r in the workstation π_j. We assume the module m_v reserves c_{vr} units of memory z_r and holds it during a program run. Both values c_{vr} and d_{jr} are nonnegative and limited.

The memory limit in a machine cannot be exceeded in the ith node, what is written, as bellows:

$$\sum_{v=1}^{V} c_{vr} x_{vi}^m \le \sum_{j=1}^{J} d_{jr} x_{ij}^\pi, \; i = \overline{1,I}, \; r = \overline{1,R}. \qquad (4)$$

The other measure of the task assignment is a cost of computers:

$$F_2(x) = \sum_{i=1}^{I} \sum_{j=1}^{J} \kappa_j x_{ij}^\pi, \qquad (5)$$

where κ_j corresponds to the cost of the computer π_j.

The fourth measure of the task assignment is a total amount of computer performance that can be deliberated according to the following formula:

$$\widetilde{F}_2(x) = \sum_{i=1}^{I} \sum_{j=1}^{J} \vartheta_j x_{ij}^\pi, \qquad (6)$$

where ϑ_j is the numerical performance of the computer π_j for assumed bank task benchmark.

4. Optimization of task assignment

The total computer cost is in conflict with the numerical performance of a distributed system, because the cost of a computer usually depends on the quality of its components. Additionally, the workload of the bottleneck computer is in conflict with the cost of the system. If the inexpensive and non-high quality components are used, the load is moved to the high quality ones and workload of the bottleneck computer increases.

Let (x, F, P) be the multi-criterion optimization question for finding the representation of Pareto-optimal solutions [19]. It can be established, as follows:

---

1) x - an admissible solution set

$$\mathrm{X} = \{x \in \mathrm{B}^{I(V+J)} \mid \sum_{v=1}^{V} c_{vr} x_{vi}^{m} \leq \sum_{j=1}^{J} d_{jr} x_{ij}^{\pi},\ i=\overline{1,I},\ r=\overline{1,R};$$

$$\sum_{i=1}^{I} x_{vi}^{m} = 1,\ v=\overline{1,V};\ \sum_{j=1}^{J} x_{ij}^{\pi} = 1,\ i=\overline{1,I}\ \},\ \mathrm{B} = \{0, 1\}$$

2) F - a vector superiority criterion

$$F: \mathrm{X} \to \mathrm{R}^4 \tag{7}$$

where

R – the set of real numbers,

$F(x) = [-R(x), Z_{\max}(x), F_2(x), -\widetilde{F}_2(x)]^T$ for $x \in$ X,

$R(x)$, $Z_{\max}(x)$, $F_2(x)$ and $\widetilde{F}_2(x)$ are calculated by (1), (3), (5) and (6), respectively

3) P - the Pareto relation [7].

In above multiobjective optimization problem related to bank task assignment, a workload of a bottleneck computer and the cost of machines are minimized. On the other hand, a reliability of the system and numerical performance are maximized.

5. Multi-criterion evolutionary algorithm

Evolutionary algorithms can be used for solving some combinatorial optimization problems, for instance the Consensus Tree Problem [8]. An overview of evolutionary algorithms for multiobjective optimization problems is submitted in [17]. The name "adaptive evolutionary algorithm" for evolutionary ones is related to the changing of some parameters as a crossover probability, a mutation rate, a population size, and the others during the searching [14, 15]. For considered algorithm, the crossover probability is decreased due to the number of new generations [9].

Figure 1 shows a scheme of the adaptive multi-criterion evolutionary algorithm called AMEA. This algorithm permits on achieving better results for task assignment than the other multiobjective evolutionary algorithms [1].

The preliminary population is created in a specific manner (Fig. 1, line 3). Generated individuals satisfy constraints $\sum_{i=1}^{I} x_{vi}^{m} = 1, v=\overline{1,V};\ \sum_{j=1}^{J} x_{ij}^{\pi} = 1,\ i=\overline{1,I}$ by introducing integer representation of chromosomes, as follows:

$$X = (X_1^m, \ldots, X_v^m, \ldots, X_V^m, X_1^\pi, \ldots, X_i^\pi, \ldots, X_J^\pi), \tag{8}$$

where $X_v^m = i$ for $x_{vi}^m = 1$ and $X_i^\pi = j$ for $x_{ij}^\pi = 1$.

Furthermore, we assume that $1 \leq X_v^m \leq I$ and $1 \leq X_i^\pi \leq J$. An integer representation of chromosomes lessen the quantity of allotments x from $2^{I(V+J)}$ to $I^V J^I$. If x is admissible, then the fitness function value (Fig. 1, line 4) is estimated, as below:

1. BEGIN
2. t:=0, set the size of population L, p_m:=1/M, M – the length of x
3. generate initial population $\boldsymbol{P}(\boldsymbol{t})$, t – the number of population
4. calculate ranks $r(x)$ and fitness $f(x), x \in \boldsymbol{P}(t)$
5. *finish*:=FALSE
6. WHILE NOT *finish* DO
7. BEGIN /* new population */
8. t:= t+1, $\boldsymbol{P}(t) := \varnothing$
9. calculate selection probabilities $p_s(x)$ $x \in \boldsymbol{P}(t-1)$
10. FOR L/2 DO
11. BEGIN /* reproduction cycle */
12. 2WT-selection of a potential parent pair (**a,b**) from the $\boldsymbol{P}(t\text{-}1)$
13. S-crossover of a parent pair (**a,b**) with the adaptive crossover rate p_c, $p_c := e^{-t/T_{\max}}$
14. S-mutation of an offspring pair (**a',b'**) with the rate p_m
15. $\boldsymbol{P}(t)$:=$\boldsymbol{P}(t)\cup$(**a',b'**)
16. END
17. calculate ranks $r(x)$ and fitness $f(x), x \in \boldsymbol{P}(t)$
18. IF ($\boldsymbol{P}(t)$ converges OR $t \geq T_{max}$) THEN *finish*:=TRUE
19. END
20. END

Fig. 1. Adaptive multicriteria evolutionary algorithm

$$f(x) = r_{\max} - r(x) + P_{\max} + 1, \tag{9}$$

where $r(x)$ denotes the rank of an admissible solution, $1 \leq r(x) \leq r_{\max}$.

In the two-weight tournament selection (Fig. 1, line 12), the roulette rule is carried out twice. If two potential parents (a, b) are admissible, then a dominated individual is eliminated. If two admissible solutions non-dominate each other, then they are accepted. If potential parents (a, b) are non-admissible, then an alternative with the smaller penalty is selected.

The fitness sharing technique can be substituted by the adaptive changing of main parameters. The quality of attained solutions increases in optimization problems with one criterion, if the crossover probability and the mutation rate are changed in an adaptive way proposed by Sheble and Britting [16]. The crossover point is randomly chosen for the chromosome X in the S-crossover operator (Fig. 1, line 13). The crossover probability is equal to 1 at the initial population and each pair of potential parents is obligatory taken for the crossover procedure.

A crossover operation supports the finding of a high-quality area in the search space. Especially,it is important in the early search stage. If the number of generation t increases, the crossover probability decreases according to the formula $p_c = e^{-t/T_{\max}}$. Some search areas are identified after several crossover operations on parent pairs. That is why, value p_c is smaller and it is equal to 0.6065, if t=100 for maximum number of population T_{max}=200. The final smallest value p_c is 0.3679. A crossover probability decreases from 1 to exp(-1), exponentially.

In S-mutation (Fig. 1, line 14), the random swap of the integer value by another one from a feasible discrete set is applied. If the gene X_v^m is randomly taken for mutation, the value is taken from the set $\{1,...,I\}$. If the gene X_i^{π} is randomly chosen, the value is selected from the set $\{1,...,J\}$. A mutation rate is constant in the AMEA and it is equal to 1/M, where M represents the number of decision variables.

6. Level of convergence to Pareto front

The AMEA is able to find task assignment representation for several numerical instances of multiobjective optimization problem [7] that was confirmed by extended simulations. Quality of obtained solutions can be assessed by a level of convergence to the Pareto front [1].

Let the Pareto points $\{P_1, P_2,..., P_U\}$ be given for a instance of the task assignment problem (7). If the AMEA finds the efficient point $(A_{u1}, A_{u2}, P_{u3}, A_{u4})$ for the cost P_{u3}, this point is associated to the uth Pareto result $(P_{u1}, P_{u2}, P_{u3}, P_{u4})$ with the same cost of computers.

The distance between points $(A_{u1}, A_{u2}, P_{u3}, A_{u4})$ and $(P_{u1}, P_{u2}, P_{u3}, P_{u4})$ is calculated according to an expression $\sqrt{(P_{u1}-A_{u1})^2+(P_{u2}-A_{u2})^2+(P_{u4}-A_{u4})^2}$. If the point $(A_{u1}, A_{u2}, P_{u3}, A_{u4})$ is not discovered by the algorithm, we assume the distance is $\sqrt{\left(P_{u1}-A_{u1}^{\min}\right)^2+\left(P_{u2}-A_{u2}^{\max}\right)^2+\left(P_{u4}-A_{u4}^{\min}\right)^2}$, where $A_{u1}^{\min}$ is the minimal reliability of the system, $A_{u2}^{\max}$ is the maximum load of the bottleneck computer, and $A_{u4}^{\min}$ is the minimum performance of the system for the instance of problem (7).

The level of convergence to the Pareto front is calculated, as follows:

$$S = \sum_{u=1}^{U} \sqrt{(P_{u1}-A_{u1})^2+(P_{u2}-A_{u2})^2+(P_{u4}-A_{u4})^2}. \quad (10)$$

An average level $\overline{S}$ is calculated for several runs of the evolutionary algorithm.

7. Tabu search as mutation operation

Initial numerical examples indicated that some obtained task assignments had higher value of the workload of the bottleneck computer than an optimal one for instances with the number of tasks larger than 15. We suggest reducing this disadvantage by an introduction a tabu algorithm [11] as an advanced mutation operator. According to this concept, a new procedure should be added to the line 14 (Fig. 1), as follows:

14 b) Tabu-mutation of an offspring pair **(a',b')** with the constant tabu-mutation probability p_{tabu}

A tabu-mutation is implemented as the tabu algorithm TSZmax [1] that has been designed to find the task assignment with the minimum value of the function $Z_{\max}$. Better outcomes from the tabu mutation are transformed into improvement of solution quality obtained by the adaptive multicriteria evolutionary algorithm with tabu mutation AMEA+. This algorithm gives better results than the AMEA (Fig. 2). After 200 generations, an average level of Pareto set obtaining is 1.8% for the AMEA+, 3.4% for the AMEA. 30 test preliminary populations were prepared, and each algorithm starts 30 times from these populations. For integer constrained coding of chromosomes, there are 12 decision variables. The binary search space consists of 1.0737×10^9 chromosomes and includes 25 600 admissible solutions.

For the other instance with 15 tasks, 4 nodes, and 5 computer sorts there are 80 binary decision variables. An average level of convergence to the Pareto set is 16.7% for the AMEA+ and 18.4% for the AMEA. A maximal level is 28.5% for the AMEA+ and 29.6% for the AMEA. For this instance the average number of optimal solutions is 19.5% for AMEA+ and 21.1% for AMEA.

An average level of convergence to the Pareto set, an maximal level, and the average number of optimal solutions become worse, when the number of task, number of nodes, and number of computer types increase. An average level is 34.6% for the AMEA+ versus 35,7% for the AMEA, if the instance includes 50 tasks, 4 nodes, 5 computer types and also 220 binary decision variables.

Using tabu search as a mutation operator causes that this hybrid algorithm is a memetic algorithm [13]. Memetic algorithms is a population-based approach for heuristic search in optimization problems. They are orders of magnitude faster than traditional genetic algorithm for some problem domains. They combine local search heuristics with crossover operators. Since they are most suitable for parallel computers and distributed computing systems, they are called parallel genetic algorithms, genetic local search or hybrid genetic algorithm.

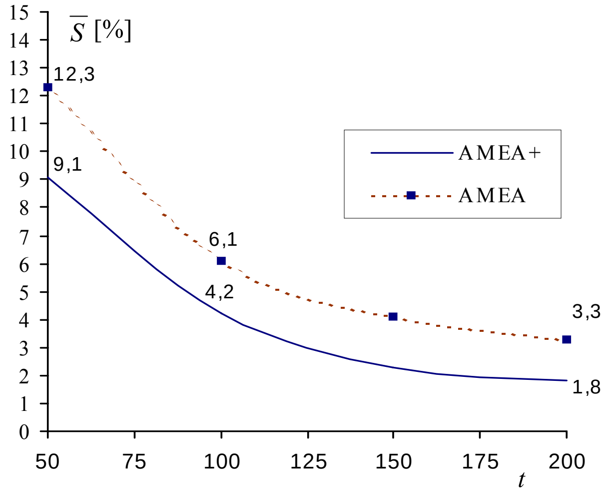

Fig. 2. Outcome convergence for the AMEA+ and AMEA

Figure 3 shows the Pareto front for the test problem with 30 decision variables. We take into account two criteria F_2 and Z_{max}. Pareto points are denoted as P_1, P_2,..., P_5. Points determined by AMEA+ are marked by circles. All points have been found by AMEA+ for this instance.

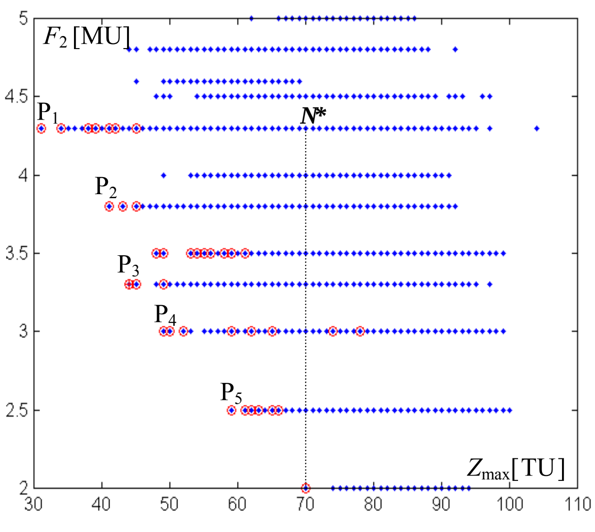

Fig. 3. Pareto front and results of AMEA+

8. Concluding remarks

The competitiveness between banks and between banks and the other financial institutions directly caused the development of the Internet banking. It forces finding methods for improving possibilities of distributed computer systems. The load balancing may improve both performance of the system and the safety of the bottleneck hosts in the bank system using the Internet. It can be obtained by task assignment as well as a selection of suitable computer sorts.

To find optimal solutions, the adaptive evolutionary algorithm with a tabu mutation AMEA+ is proposed. It is an advanced technique for finding Pareto-optimal task allocations in four-objective optimisation problem with the maximisation of the system reliability and distributed system performance. Moreover, the workload of the bottleneck computer and the cost of computers are minimized.

Tabu search algorithm can be used to improve a quality of an offspring that is randomly chosen from the current population maintained by an evolutionary algorithm. The workload of the bottleneck computer is selected to be improved by the tabu algorithm for the four-criterion task assignment problem.

Our future works will concern on a development the combination between tabu search and evolutionary algorithms for finding Pareto-optimal solutions. Tabu search algorithms can be used for the local improving of non-dominated solution in population.

References

[1] Balicki J., Immune Systems in Multi-criterion Evolutionary Algorithm for Task Assignments in Distributed Computer System. Lectures Notes in Computer Science, Vol. 3528, 2005, pp. 51-56.
[2] Burkhardt T., Lohmann K. (Eds.). Banking und electronic commerce in Internet. Springer Verlag, Berlin 1998.
[3] Chissick M., Kelman A., Electronic commerce: law and practice. Sweet - Maxwell, London 2000.
[4] Chorafas D., The Commercial banking handbook. McMillan Business, London 1999.
[5] Cronin M. J. (Ed), Banking and finance on the Internet, John Wiley &Sons, New York 1997.
[6] Chu, W. W., Lan, L. M. T., Task Allocation and Precedence Relations for Distributed Real-Time Systems. IEEE Transactions on Computers, Vol. C-36, No. 6, 1987, pp. 667-679.
[7] Coello Coello C. A., Van Veldhuizen D. A., Lamont G.B., Evolutionary Algorithms for Solving Multi-Objective Problems. Kluwer Academic Publishers, New York 2002.
[8] Cotta C., On the Application of Evolutionary Algorithms to the Consensus Tree Problem. In G. R. Raidl, J. Gottlieb (Eds.): Evolutionary Computation in Combinatorial Optimization, Proceedings on the 5th European Conference, 2005, Lausanne, Switzerland, pp. 58-67.
[9] Deb, K., Multi-Objective Optimization using Evolutionary Algorithms, John Wiley & Sons, Chichester.
[10] Glover F., Laguna M., 1997. Tabu Search. Kluver Academic Publishers, Boston 2001.

[11] Hansen M. P., Tabu Search for Multicriteria Optimisation: MOTS. Proceedings of the Multi Criteria Decision Making, 1997, Cape Town, South Africa. pp. 123-134.
[12] Jaszkiewicz, A. Hapke, M. Kominek, P., Performance of Multiple Objective Evolutionary Algorithms on a Distributed System Design Problem – Computational Experiment. Lectures Notes in Computer Science. Special Issue: Evolutionary Multi-Criterion Optimization by E. Zitzler, K. Deb, L. Thiele (Eds), Vol. 1993, 2001, Springer-Verlag. pp. 241-255.
[13] Merz, P., Freisleben, B., Memetic Algorithms and the Fitness Landscape of the Graph Bi-Partitioning Problem. Lecture Notes in Computer Science, Vol. 1498. 1998, pp. 765-774.
[14] Michalewicz, Z., Genetic Algorithms + Data Structures = Evolution Programs. Springer Verlag, Berlin Heidelberg New York 1996.
[15] Schaefer, R., Kołodziej, J., Genetic Search Reinforced by the Population Hierarchy. In De Jong K. A., Poli R., Rowe J. E. (eds): Foundation of Genetic Algorithms, Morgan Kaufman Publisher. 2003, p. 383-399.
[16] Sheble, G. B., Britting, K., Refined Genetic Algorithm – Economic Dispatch Example. IEEE Transactions on Power Systems, Vol. 10, No. 2. 1995, pp. 117-124.
[17] Van Veldhuizen, D. V., Lamont, G. B., Multiobjective Evolutionary Algorithms: Analyzing The State-Of-The-Art. Evolutionary Computation. Vol. 8, No. 2, 2000, pp. 125-147.
[18] Weglarz, J., Nabrzyski, J., Schopf, J., Grid Resource Management: State of the Art and Future Trends. Kluwer Academic Publishers, Boston 2003.
[19] Zitzler, E., Deb, K., and Thiele, L., Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. Evolutionary Computation, Vol. 8, No. 2. 2000, pp. 173-195.

Jerzy Balicki received the M.S. and Ph.D. degrees in Computer Science from Warsaw University of Technology in 1982 and 1987, respectively. During 1982-1997, he stayed in Computer Center High School of Gdynia to study management systems, mobile systems, and decision support systems. He is now with Distributed Computer Laboratory of Naval University of Gdynia.