Pareto-optimal Load Solutions in the I-Banking by Evolutionary Algorithm

Jerzy Balicki, Honorata Balicka

Computer Science Department, Naval University of Gdynia
ul. Smidowicza 69, 81-103 Gdynia, Poland
J.Balicki@amw.gdynia.pl

Faculty of Economics, University of Gdansk
ul. Armii Krajowej 119/121, 81-824 Sopot, Poland
hbalicka@interia.pl

Abstract

Load balancing of the Web bank servers can be implemented by minimization of the workload of the bottleneck Web server. Load balancing improves both a performance of the system and the safety of the bottleneck computers. An evolutionary algorithm based on 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.

1. Introduction

Bank transactions are parallel performed subject to geographic spread of clients. An average cost of a transaction through the Internet is lower about 100 times than the cost of a transaction carried out in the bank with the network of branches [Burkhardt et al. 98]]. Moreover, 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 [Chissick et al. 00]. It is a decisive motivation of the tendency displayed itself in the Internet banking expenditure has been doubled each year since 1997 [Chorafas 99].

Computer technology is suitable for implementation of banking transactions [Cronin 97]. Computer data representing money are processed, stored and transferred with using the relevant computer technology. However, the lack of wide computer networks has impeded the progress of the bank systems till late ninetieth. When the Web site technology was developed to design an interactive service, then the technology conditions were convenient to make use of the bank systems via the Internet.

On the other hand, evolutionary algorithms can inherit some abilities of tabu search techniques to improve a quality of obtained Pareto-suboptimal solutions [Jaszkiewicz 01]. A tabu search is the powerful meta-heuristic approach that has been applied for crucial applications in engineering, economics and science [Glover et al. 97]. A first version of a multicriteria evolutionary algorithm with a tabu search as an advanced mutation operation has been suggested in [Balicki et al. 03].

Finding allocations of program modules in the Web bank system may reduce the entirety time of a program run by taking a benefit of the particular properties of some workstations or an advantage of the computer load. An adaptive evolutionary algorithm and an adaptive evolution strategy have been considered for solving multiobjective optimization problems related to task assignment that minimize a workload of a bottleneck computer and the cost of machines [Balicki et al. 03]. The total numerical performance of workstations is

ISSN: 1137-3601.

another criterion for assessment of task assignment. Furthermore, a reliability of the system is a criterion that is significant to assess the quality of a task assignment in bank systems. Subsequently, the problem with above four criteria and also memory constraint is discussed.

2. Model of bank software system

Banks can use the Internet to support their traditional tasks that are related to basic services [Pietrzak 97]. This substitution of some tasks may impact on the acceleration of the service and it permits users to save time. For example, a client of a bank may require an access to the database through the Internet to view the balance of the account or display a transaction history. It improves, radically, the comfort for the bank clients. On the other hand, some complicated secure procedures have to be introduced and still developed to protect the access to information against some offensive players.

Additionally, a temporary deposit can be opened or closed, remotely what saves the time of a client and gives an opportunity to an efficient management of money. Furthermore, a money transfer between interior accounts of an account is permitted for 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 management of the home budget. The financial management is extended to pay just before the deadlines of payment.

Monitoring of transactions, including transactions paid by a credit card, gives information about the current rate of payment. Thus, we can reasonably plan the other payments. A WWW home page of the bank includes some calculating procedures for estimating an interest of deposit in the given term. It supports making decision about credits as well as deposits. Moreover, the process of preparing necessary data is subject to shorten.

These basic tasks can be relatively easy implemented by using the Apache server that is a powerful, flexible, HTTP compliant web server [Balicki et al. 03]. It implements the latest protocols, including HTTP and it is highly configurable and extensible with third-party modules. It can be customized by writing modules using the Apache module API and it provides a source code. What is more, it runs on Windows 2000/XP, Netware 5.x and above, OS/2, and most versions of Unix/Linux, as well as several other operating systems. The Apache server is actively being developed and also it encourages user feedback through new ideas, bug reports and patches.

The Apache server implements many frequently requested features, including DBM databases for authentication. It allows setting up password-protected pages with enormous numbers of authorized users, without bogging down the server. Customized responses to errors allows setting up files, or even CGI scripts, which are returned by the server in response to errors and problems. It is possible to setting up a script to intercept 500 server errors and perform on-the-fly diagnostics for users.

Information service gives the marketing advantages for a bank and is relevant for users that active manage their money. It is an essential proposal of conventional banks at the beginning of the Internet using.

An execution of payment from a bank by a client requires more advanced approach related to the filling up a relevant form in the secured WWW page. An order of payment is prepared by a client, and then it is send to the destination account. Data from the payment 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 the constant data are stored by the depository system and the user may alter some latest parts of prepared form, only.

Prediction and optimization of payments in the next months are out of the ordinary tasks performed by the bank computer system. 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 are set up just before the deadlines as well as cheaper credits are suggested.

Above tasks from the conventional bank are implemented by the Internet bank that based on distributed software consisted of bank servers and client browsers. Advanced tasks go beyond this service and they make use the interactivity, multimedia property and flexibility of the Internet, what gives an innovative approach to bank services.

To attract clients, banks enrich their WWW services by additional current information about exchange rates, tax regulations and stock exchange that are not related to the bank services. Moreover, the high-quality home page of the WWW bank service is supposed to be included as the favorite page to the browser of clients, what gives the opportunity to promote novel products or higher possibilities of cross-selling.

The WWW bank service may recommend 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, air-lines tickets, tickets to cinemas, theatres, matches can be sell as well as 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 appearance to the personal preferences of a client.

The most leading banks go beyond above set of tasks performed via the Internet because they want to take advantage of the virtues of the worldwide distributed system. For the reason that a potential client can use a mobile phone to receive and send written messages SMS and to receive voice messages and info messages, banks use these channel of information communication to inform the clients about the state of their accounts. A development of the WAP service based the GSM network on gives a micro browser that is convenient to use the Internet resources, especially if a mobile phone cooperates with a computer.

Advising online can be implemented by chat software for written questions by a client, and then reading answers during a real time dialog with an expert from the bank. A voice dialog is possible by the Internet and also a voice and vision communication can be carried out with using modern software.

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

3. 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 are implemented as 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 $\{T_1,...,T_v,...,T_V\}$ [Weglarz 98].

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:

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

where

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

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

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

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.

4 Workload of the bottleneck computer

The cost of the parallel program performing is the most common used measure of an allowance evaluation [Balicki et al. 03]. 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. The workload of the bottleneck computer is another fundamental criterion for the evaluation of an allocation quality [Kafil et al. 98]]. A computer with the heaviest task load is the bottleneck machine, and its workload is a critical value that is supposed to be minimized [Chu et al. 87]. The workload $Z_i^+(x)$ of a computer allotted to the ith node for the allocation x is provided, as follows:

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

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:

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

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 consignment than another bottleneck machine for the other task allocation in a 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 execution. Let the following memories $z_1,...,z_r,...,z_R$ 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 any machine cannot be exceeded in the ith node, what is written, as bellows:

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

The other measure of the task assignment is a cost of computers [Balicki et al. 03]:

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

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 [Balicki et al. 03]:

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

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

5 Optimization of task assignment

Let (X, F, P) be the multi-criterion optimisation question for finding the representation of Pareto-optimal solutions [Zitzler et al. 00]. It can be established, as follows:

---

1) X - an admissible solution set

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

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

2) F - a vector superiority criterion

$$F: \mathsf{X} \rightarrow \mathsf{R}^4 \quad (7)$$

where

R – the set of real numbers,

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

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

3) P - the Pareto relation [Coello 97].

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.

In above multiobjective optimisation problem related to bank task assignment, a workload of a bottleneck computer and the cost of machines are minimized [Balicki et al. 03]. On the other hand, a reliability of the system and numerical performance are maximized.

6 Multi-criterion evolutionary algorithm

An overview of evolutionary algorithms for multiobjective optimisation problems is submitted in [Van Veldhuizen et al. 00]. The name "adaptive evolutionary algorithm" for evolutionary algorithms is related to the changing of some parameters as a crossover probability, a mutation rate, a population size, and the others during the searching [Michalewicz 96]. For considered algorithm, the crossover probability is decreased due to the number of new generations.

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 [Balicki et al. 03].

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:

$$X=(X_1^m,...,X_v^m,...,X_V^m,X_1^\pi,...,X_i^\pi,...,X_J^\pi), \quad (8)$$

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

1. BEGIN
2. t:=0, set the size of population L, p_m:=1/M, M – the length of x
3. generate initial population ***P*(*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 ***P***(*t*-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. ***P***(*t*):=***P***(*t*)∪(**a'**,**b'**}
16. END
17. calculate ranks $r(x)$ and fitness $f(x), x\in \boldsymbol{P}(t)$
18. IF (***P***(*t*) converges OR $t\geq T_{max}$) THEN *finish*:=TRUE
19. END
20. END

Fig. 1. Adaptive multicriteria evolutionary algorithm

Furthermore, we assume that $1\le X_v^m \le I$ and $1\le X_i^\pi \le J$. 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:

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

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

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 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 optimisation problems with one criterion, if the crossover probability and the mutation rate are changed in an adaptive way proposed by Sheble and Britting [Sheble 95]. 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 solution area in the search space. It is important in the early search stage. If the number of generation t increases, the crossover probability decreases according to the formula $p_c=e^{-t/T_{\max}}$. The search space or 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.

7 Level of convergence to Pareto front

The AMGA is able to find task assignment representation for several numerical instances of multiobjective optimisation problem (7) that was confirmed by extended simulations. Quality of obtained solutions can be assessed by a level of convergence to the Pareto front [Balicki et al. 03].

Let the Pareto points $\{P_1, P_2, ..., P_U\}$ be given for any instance of the task assignment problem (7). If the AMGA finds the efficient point $(A_{u1}, A_{u2}, P_{u3}, A_{u4})$ for the cost of computers P_{u3}, this point is associated to the *u*th 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.

8 Tabu search as mutation operation

Initial numerical examples indicated that 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 [Hansen 97] 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 [Balicki et al. 03] 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 and the search space consists of 25 600 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.

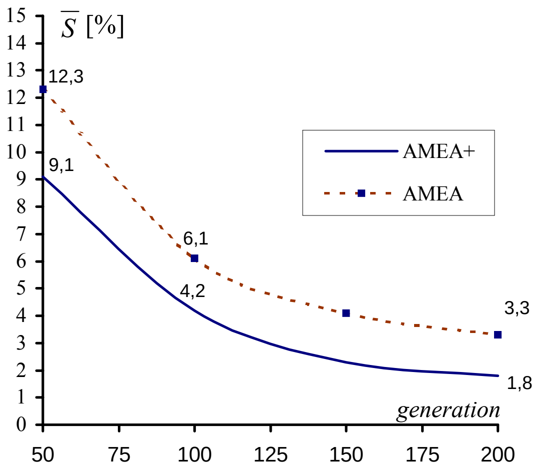

Fig. 2. Outcome convergence for the AMEA+ and the AMEA

Using tabu search as a mutation operator causes that this hybrid algorithm is a memetic algorithm [Merz et al. 98]. Memetic Algorithms is a population-based approach for heuristic search in optimization problems. They are orders of magnitude faster than traditional genetic algorithm

k

for some problem domains. Basically, 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.

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 $P_1, P_2, \ldots, P_5$. Points determined by AMEA+ are marked by circles. All points have been found by AMEA+ for this instance.

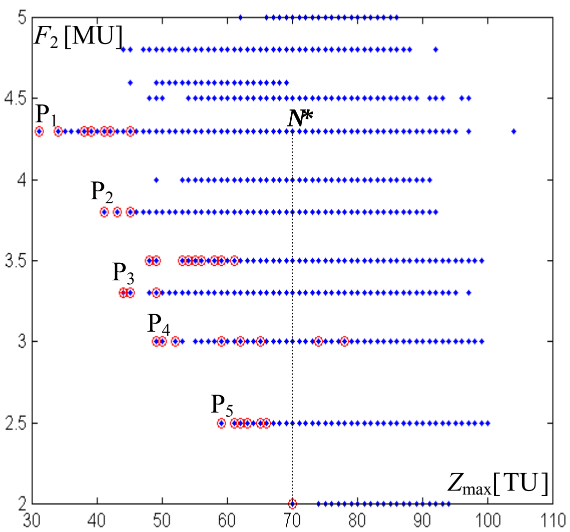

Fig. 3. Pareto front and results of AMEA+

9. Concluding remarks

The competitiveness between banks and also 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-criteria 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

J. Balicki, H. Balicka: *Load balancing of the Internet bank servers by tabu-based algorithm*, chapter in M. Paprzycki (ed.): *Electronic Commerce. Research and Development*. Systems, Man and Cybernetics Society, IEEE, Technical University of Gdańsk, Gdańsk 2003, pp. 11-18.

Burkhardt T., Lohmann K. (Eds.), *Banking und electronic commerce in Internet*, Berlin Verlag 1998.

Chissick M., Kelman A., *Electronic commerce: law and practice.* Sweet - Maxwell, London 2000.

Chorafas D., *The Commercial banking handbook*, McMillan Business, London 1999.

Cronin M.J. (Ed), *Banking and finance on the Internet*, John Wiley &Sons, New York 1997.

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) 667-679

Coello Coello C. A.: *A Comprehensive Survey of Evolutionary-Based Multiobjective Optimisation Techniques*. Knowledge and Information Systems. An International Journal, Vol. 1 (1999) 269-308

Glover F., Laguna M.: *Tabu Search*. Kluver Academic Publishers, Boston (1997)

Hansen M. P.: *Tabu Search for Multicriteria Optimisation: MOTS*. Proceedings of the Multi Criteria Decision Making, Cape Town, South Africa (1997)

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 Springer-Verlag (2001) 241-255.

Kafil, M. Ahmad, I.: *Optimal Task Assignment in Heterogeneous Distributed Computing Systems*. IEEE Concurrency, Vol. 6, No. 3 (1998) 42 - 51

Merz, P., Freisleben, B.: *Memetic Algorithms and the Fitness Landscape of the Graph Bi-Partitioning Problem,* Lecture Notes in Computer Science, Vol 1498, (1998) 765-774

Michalewicz, Z.: *Genetic Algorithms + Data Structures = Evolution Programs*. Springer Verlag, Berlin Heidelberg New York (1996)

Pietrzak E.: *Exchange rate policy, foreign exchange market, and derivatives, state-of-the-art and perspectives*. Transformacja Gospodarki, 88, Gdansk 1997 (in Polish).

Sheble, G. B., Britting, K.: *Refined Genetic Algorithm – Economic Dispatch Example*. IEEE Transactions on Power Systems, Vol. 10, No. 2 (1995) 117-124

Van Veldhuizen, D. V., Lamont, G. B.: *Multiobjective Evolutionary Algorithms: Analyzing The State-Of-The-Art. Evolutionary Computation,* Vol. 8, No. 2 (2000) 125-147

Weglarz J. (ed.): *Recent Advances in Project Scheduling*. Kluwer Academic Publishers, Dordrecht (1998)

Zitzler, E., Deb, K., and Thiele, L.: *Comparison of Multiobjective Evolutionary Algorithms: Empirical Results*. Evolutionary Computation, Vol. 8, No. 2 (2000) 173-195