Numerical Experiments on Pareto-optimal Task Assignment Representations by Tabu-based Evolutionary Algorithm

JERZY BALICKI
Naval University of Gdynia
ul. Smidowicza 69, 81-103 Gdynia,
POLAND
J.Balicki@amw.gdynia.pl

Abstract: - Meta-heuristics like evolutionary algorithms require extensive numerical experiments to adjust their capabilities of solving decision making problems. Evolutionary algorithm can be applied for finding solution in distributed computer systems. Reliability and the load balancing are crucial factors for a quality evaluation of distributed systems. Load balancing of the Web servers can be implemented by reduction of the workload of the bottleneck computer what 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 multi-criteria optimization of distributed systems A tabu mutation is applied for minimization the workload of the bottleneck computer. It can be obtained by task assignment as well as selection of suitable computer sorts. Moreover, a negative selection procedure is developed for improving non-admissible solutions. Extended numerical results are submitted.

Key-Words: - Evolutionary algorithm, Multi-criterion optimization, Distributed systems, Artificial intelligence, Pareto solutions

1 Introduction

Distributed systems are dynamically developed in different areas such as distance learning, e-commerce, telecommuting, Internet television, web radio, online advising, e-management and Internet health support. Among them, the Internet banking distinguishes according to the rapid development. It has been applied since 1994. Bank transactions are parallel performed subject to geographic spread of users. An average cost of a transaction via the web is lower about 100 times than the cost of a transaction carried out in the bank with the network of branches [16]. Besides, the Internet transaction cost is significantly lower than the transaction cost for the cash machine network or for the financial advice system through a phone network [5].

Computer technology is suitable for implementation of banking transactions [4]. 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 web.

Such complex systems require more advance optimization techniques for decision making by designers. So, the algorithms with artificial intelligence rules can be used [11]. Especially, evolutionary algorithms can inherit some abilities of tabu search techniques to improve a quality of obtained Pareto-suboptimal solutions [2]. A tabu search is the powerful meta-heuristic approach that has been applied for crucial applications in engineering, economics and science [20]. An advanced version of a multicriteria evolutionary algorithm with a tabu search as an advanced mutation operation has been suggested in [3].

In the first multi-criterion genetic algorithm called a *vector evaluated genetic algorithm* VEGA, a population of solutions is divided on N subpopulations, where N is the number of partial criteria [19]. For each nth subpopulation, the criterion F_n is a fitness function. However, a crossover and a mutation are carried out for the complete population. This method for a fitness evaluation has a weakness related to the discrimination of Pareto solutions located in an interior of the Pareto front.

Fourman has considered the selection with binary tournaments, where two randomly chosen solutions have been compared [9]. The hierarchical alternative is chosen and it is included to a mating pool of potential parents. A selection probability is calculated for the most significant aim. A random choice is carried out twice according to the roulette rule. Hierarchical tournaments push the population towards lexicographical solutions likewise the VEGA approach. Another selection is based on a random choice of a goal that is taken to comparison of selected solutions. Selection probabilities are constant or they can depend on the chosen purpose for the other tournament selection.

A ranking idea for non-dominated individuals has been introduced to avoid the prejudice of the interior Pareto alternatives by Goldberg [19]. Then, Srinivas and Deb [18] have built a non-dominated sorting genetic algorithm NSGA on the ideas mentioned by Goldberg. If some admissible solutions are in a population, then the Pareto-optimal individuals are determined, and after that they get the rank 1. Subsequently, they are temporary eliminated from the population. Next, the new Pareto-optimal alternatives are found from the reduced population and they get the rank 2. In this procedure, the level is increased and it is repeated until the set of admissible solutions is exhausted. All non-dominated individuals have the same reproduction fitness because of the equivalent rank.

To maintain the diversity of the population and to preclude premature convergence, fitness-sharing techniques have been developed [22]. A mating restriction assumes that individuals from a criteria space neighbourhood are similar, so that they can form stable niches. If a non-dominated evaluation for the current population has a long distance to the nearest non-dominated evaluation and there is a niche of non-dominated results, then the separated individual is supposed to be preferred by increasing its fitness before individuals from the niche.

Above multi-criterion techniques are based on a genetic algorithm. Another approach is an extension of evolution strategy. Binh and Korn have developed a multicriteria evolution strategy for combinatorial optimisation problems [15]. A Pareto archived evolution strategy called PAES was proposed by Knowles and Corne [15].

Zietzler has suggested an elitist selection in their strength Pareto evolutionary algorithm SPEA [22]. At each generation, a combined population with the external and the current population is constructed. In an external population, all non-dominated solutions discovered so far are archived. An elitist selection is still applied in the other elitist non-dominated sorting genetic algorithms.

Finding allocations of program modules in the Web bank system may reduce the entire workload 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 [3]. The total numerical performance of workstations is another criterion for assessment of task assignment and it has been involved to multicriteria task assignment problem in [3]. 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 a web bank system

Banks can use the Internet to support their conventional tasks that are related to basic services [6]. 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 [7].

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, web compliant web server [3]. It implements the latest protocols, 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 Server 2008/Vista, 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 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 allow 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 several 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 web 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 web 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 web 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.

Consequently, the web bank system consists of several task that handle to user requirements. Figure 1 shows Z_{max} - the workload of the bottleneck computer in the distributed system for generated task assignments by an enumerative algorithm. The function Z_{max} takes value from the period [40; 110] (TU - time unit) for 256 solutions. What is more, even a small change in task assignment related to the movement of a task to another computer or a substitution of computer sort can cause a relatively big alteration of its workload. For instance, the migration of one task from the assignment with Z_{max}=40 TU may increase the workload to the 64 or even 88 TU.

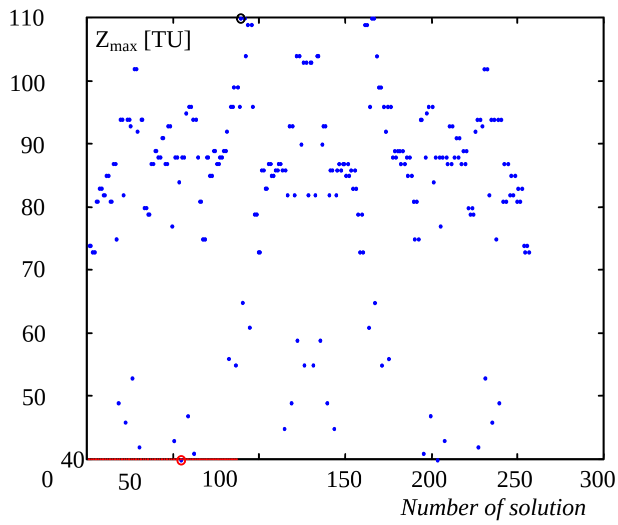

Fig. 1. Workload distribution of the bottleneck computer for generated solutions

3 Reliability and workload

A program module in the 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 $\{T_1,...,T_v,...,T_V\}$ [17].

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}.

Figure 2 shows an assignment of tasks to computers for the given set of tasks and the set of computer sorts. In the node w_i, a task T_v is assigned to the computer π_j and a task T_u is assigned to the computer π_1 in the node w_k. If there are some interactions between these tasks, then the communication time τ_{vuik} is included to the workload of the computers in the nodes w_i and w_k. It may change the load of a bottleneck computer, too. The computer π_j in the node w_i can decrease the total cost of computers, if it usually has a higher numerical performance ϑ_j according to the assumed performance benchmark.

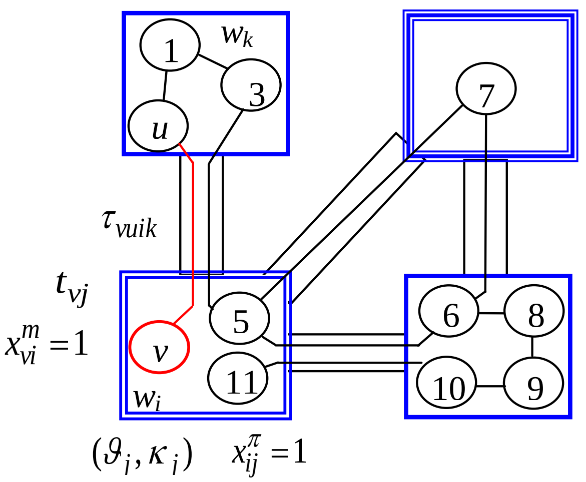

Fig. 2. Assignment of tasks to computers for the given set of tasks and the set of computer sorts

Let a computer π_j be failed independently due to an exponential distribution with rate λ_j. 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:

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

where

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

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

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

Figure 3 shows a relation between the reliability R_j of computer and parameter λ_j.=0.001 [TU^{-1}, TU – time unit].

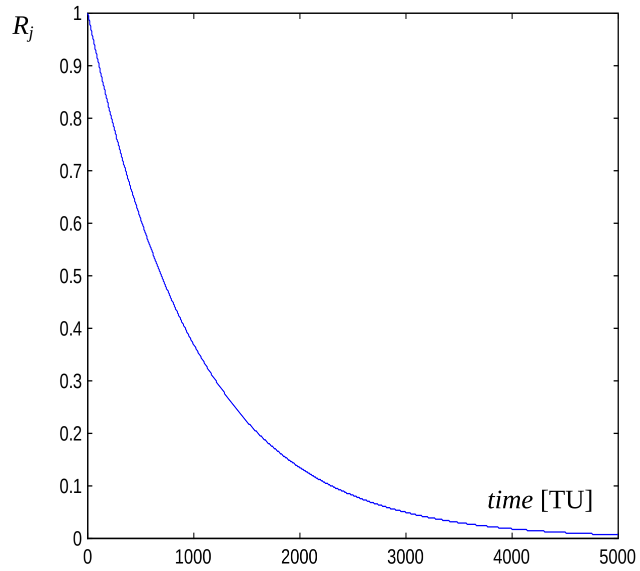

Fig. 3. A reliability of the selected computer

If there are two computers with λ_1.=0.001 [TU^{-1}] and λ_2.=0.002 [TU^{-1}], the reliability of the two-computer system decreases faster than the reliabilities for each of them. Figure 4 shows the relation between the measure of system reliability R and time of using this system for the chosen two-computer system.

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.

The cost of the parallel program performing is the most common used measure of an allowance evaluation [13]. 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 [8].

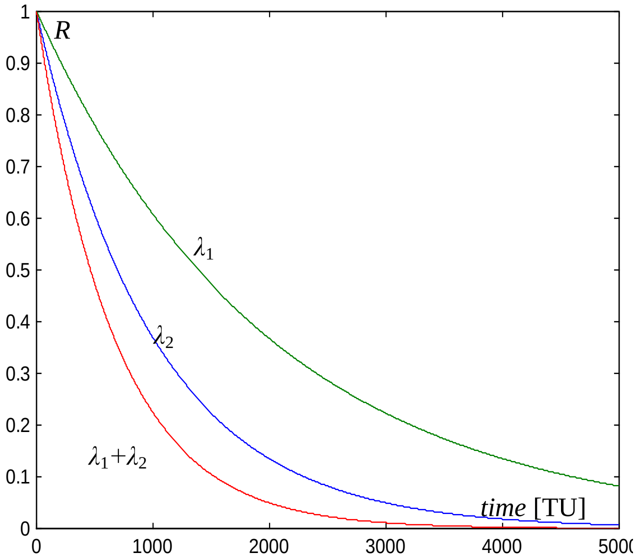

Fig. 4. The reliability of the two-computer system

A computer with the heaviest task load is the bottleneck machine, and its workload is a critical value that is supposed to be minimized. 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 the T_u.

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

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

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 [3].

Let consider the tabu algorithm version called TSZmax [2] that has been designed to find the task assignment with the minimum value of the workload of the bottleneck computer $Z_{\max}$. Figure 5 shows the process of the minimization $Z_{\max}$ from the initial value equal to 62 time units to 32.

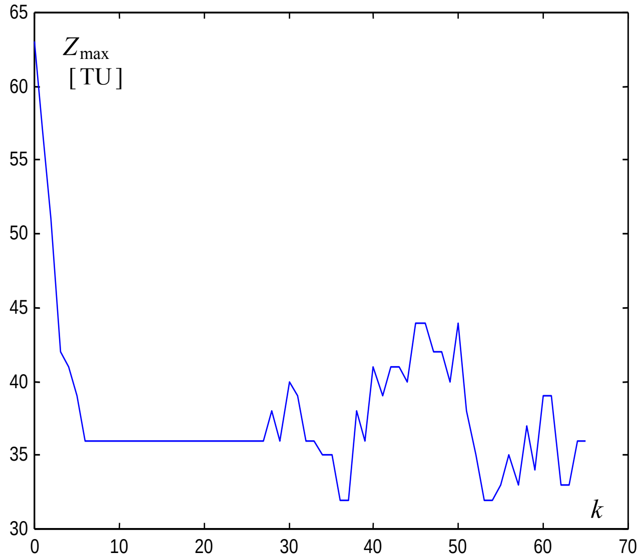

Fig. 5. Minimization of the bottleneck computer workload by the tabu search algorithm

Similarly, Figure 6 shows the process of the minimization $Z_{\max}$ from the initial value equal to 170 time units to 101. The solution with the value 170 was randomly taken from the current population of genetic algorithm with the probability p_{tm} .

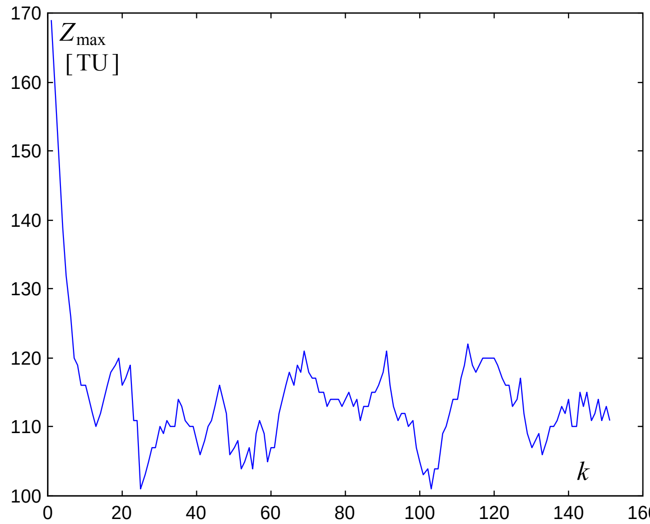

Fig. 6. Minimization of the bottleneck computer workload for the solution taken from the population

Each computer ought to be equipped with required capacities of resources for a program execution. Let the following memories $z_1,\ldots,z_r,\ldots,z_R$ 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:

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

The other measure of the task assignment is a cost of computers [2]:

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

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 [3]:

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

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

4 Multi-criterion evolutionary algorithm

Let $(\mathcal{X}, F, P)$ be the multi-criterion optimisation question for finding the representation of Pareto-optimal solutions [1]. It can be established, as follows:

1) $\mathcal{X}$ - an admissible solution set

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

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

2) F - a vector superiority criterion

$$F : \mathcal{X} \to \mathcal{R}^4 \qquad (7)$$

where

$\mathcal{R}$ – the set of real numbers,

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

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

3) P - the Pareto relation [1]

The relationship P is a subset of the product $Y \times Y$, where an evaluation set $Y = F(\mathcal{X})$. If $a \in Y$, $b \in Y$, and $a_n \le b_n, n = \overline{1,N}$, then the pair of evaluations $(a,b) \in P$. The meaning of the Pareto relationship respects the minimization of all criteria. That is why, criteria for maximization in (7) are written with minus. There is no task allocation $a \in \mathcal{X}$ such that $(F(a), F(x^*)) \in P$ for the Pareto-optimal assignment $x^* \in \mathcal{X}$ and $a \neq x^*$.

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 [3]. On the other hand, a reliability of the system and numerical performance are maximized.

An overview of evolutionary algorithms for multiobjective optimisation problems is submitted in [19]. 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 [12]. 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 [3].

The preliminary population is created in a specific manner (Fig. 7, 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:

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

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

Furthermore, we assume that $1 \le X_v^m \le I$ 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 $I^V J^I$. The binary coding of chromosomes causes that the search space includes 2^M elements and it is much greater potential chromosomes than in the search space generated by the integer coding. The integer coding of chromosomes causes that the search space contains $I^V J^I$ alternatives. Because $M=I(V+J)$, then the number of potential chromosomes can be calculated. Table 1 presents numbers of potential chromosomes for given number of tasks and computers for J=0 what means that the possibility of computer sort exchange is not considered.

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 ($\boldsymbol{P}(\boldsymbol{t})$ converges OR $t \geq T_{max}$) THEN *finish*:=TRUE
19. END
20. END

Fig. 7. Adaptive multicriteria evolutionary algorithm

Table 1. Numbers of potential chromosomes for given number of tasks and computers
A – binary coding
B – integer coding

V	Number of computers I			
	5		100	
	A	B	A	B
2	1024	25	$1{,}607\times10^{60}$	10^4
10	$1{,}126\times10^{15}$	$9{,}766\times10^{6}$	$1{,}072\times10^{301}$	10^{20}
10^2	$3{,}273\times10^{150}$	$7{,}889\times10^{69}$	$1{,}436\times10^{3004}$	10^{200}
10^3	$1{,}486\times10^{1502}$	$4{,}223\times10^{684}$	$1{,}325\times10^{30005}$	10^{2000}

Let the time of comparing two chromosomes be 1 nanosecond for a reference computer. According to the table 1, we can estimate the calculation time is 4.02×10^{11} centuries for the case with 10 tasks and 10 computers if the binary search space is viewed by a systematic way. However, it takes merely 10 seconds for the integer coding of chromosomes.

If x is admissible, then the fitness function value (Fig. 7, line 4) is estimated, as below:

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

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

A procedure of finding ranks $r(x)$ is the key procedure in a multi-criterion evolutionary algorithms. There are two main approaches for the rank assignment. The first procedure is based on the number of non-dominated solution levels [18] and the second one is based on the superior individuals [19].

Let twelve individuals in the current population have evaluations shown in Figure 10. The ranking procedure based on the number of non-dominated solution levels [18] (Fig. 10a) assigns the other rank distribution than the ranking procedure based on the number of superior individuals [6] (Fig. 10b). Points E and D are dominated by the larger number of evaluations and have larger ranks than the ranks calculated by of non-dominated level procedure. They have fewer chances to be selected for the second ranking scheme than for the Goldberg's one. It might produce premature convergence. Moreover, for the second procedure points A and C have dissimilar ranks, but they are non-inferior individuals, if solutions with rank 1 are temporary eliminated.

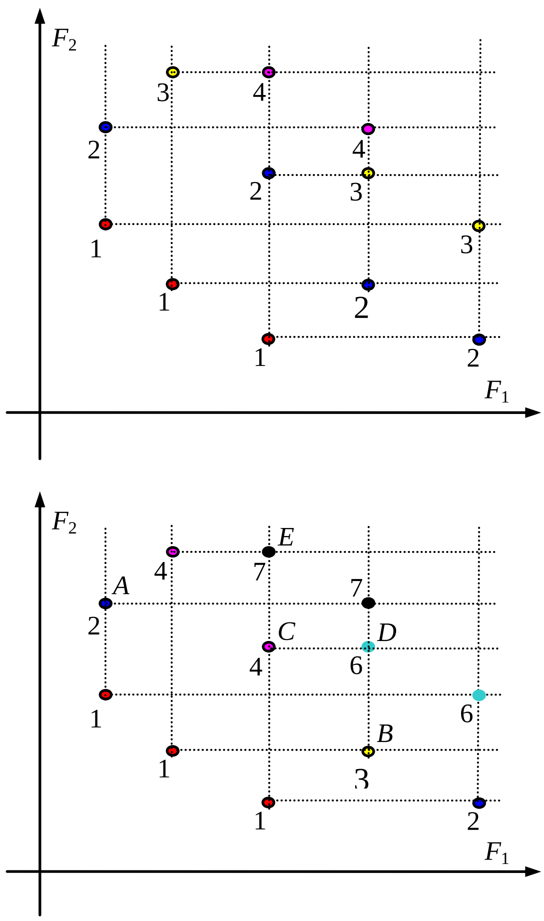

Fig. 10. Bi-criteria ranking procedures applying the number of:
a) non-dominated levels b) superior individuals

A rank assignment respects the Pareto condition and also reduces the potentially dominated consequences of relatively high-superior individuals by establishing an unsurprising, restricted quantity of selection pressure in favour of such solutions. On the other hand, a rank assignment over stresses the difference between closely clustered evaluations of solutions so that the better ones can be chosen more.

Ranks are assigned to the admissible solutions, only. If x is admissible, then the fitness function value can be calculated, as below:

$$f(x) = \mu L(-r(x) + r_{\max} + 1) + \theta_{\max}, \qquad (10)$$

where
$\theta_{\max}$ – the maximal value of the penalty function θ in the current population,
μ – the preference coefficient for feasible solutions (usually μ=2).

In the two-weight tournament selection (Fig. 7, 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 [17]. The crossover point is randomly chosen for the chromosome X in the S-crossover operator (Fig. 7, 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. 7, 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.

To improve the quality of task assignment, we propse the development of the negative selection algorithm (NSA) from an immune systems. The immune system can be seen as a distributed adaptive system that is capable for learning, using memory, and associative retrieval of information in recognition. The negative selection algorithm is based on the discrimination principle that is used to know what is a part of the immune system is.

Detectors are randomly generated to reduce those detectors that are not capable of recognizing themselves. Subsequently, detectors proficient to distinguish trespassers are kept. An adjusted detection is performed probabilistically by the NSA. An antigen is a molecule that stimulates a response against trespassers. The term originated from the notion that they can stimulate antibody generation. Moreover, the immune system consists of some viruses as well as bacteria.

An antibody (an immunoglobulin) is a large Y-shaped protein used to identify and neutralize foreign objects like bacteria and viruses. The antibody recognizes a specific target - an antigen. The negative selection can be used to manage constraints in an evolutionary algorithm by isolating the contemporary population in two groups. Feasible solutions called “antigens” create the first cluster, and the second cluster of individuals consists of “antibodies” – infeasible solutions. For that reason, the NSA is applied to generate a set of detectors that verify the state of constraints.

We assume the initial fitness for antibodies is equal to zero. Then, a randomly chosen antigen G^- is compared to the selected antibodies. After that, the distance S between G^- and the antibody B^- is calculated due to the amount of similarity at the genotype level. The measure of genotype similarity between the antigen and the antibody depends on their representation. This assessment of similarity for the integer version is, as follows [21]:

$$S\,(G^-, B^-) = \sum_{m=1}^{M} \left| G_m^- - B_m^- \right|, \qquad (11)$$

where
M − the length of the solution,
G_m^- – value of the antigen at position m, $m = \overline{1, M}$,
B_m^- – value of the antibody at position m, $m = \overline{1, M}$;

The negative selection can be implemented by a modified genetic algorithm. In that approach, infeasible solutions that are similar to feasible ones are preferred in the current population. Although, almost all the random choices are based on the uniform distribution, the pressure is directed to improve the fitness of appropriate infeasible solutions.

Figure 11 shows the cut of the evaluation space that is explored by the most effective meta-heuristic AMEA* [3].

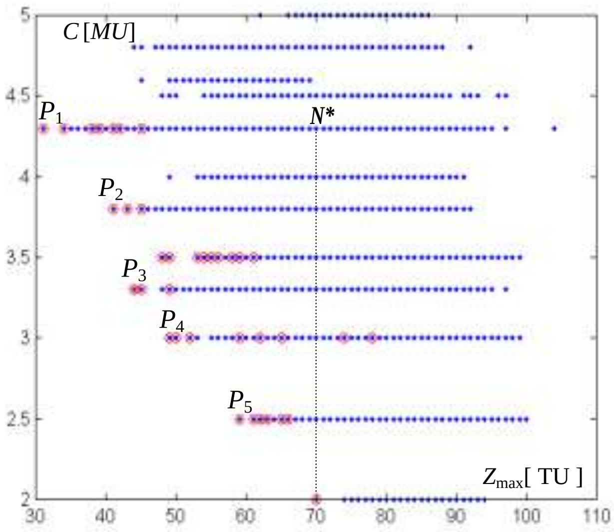

Fig. 11. Pareto front and results of AMEA*

Let the instance be considered, where there are 10 tasks, 2 nodes, and 5 computer types. It induces 30 binary decision variables and 1 073 741 824 binary task assignments. Z_{max} is a value from [26;75] [time unit], F_2 is from [2, 10] [money unit], and $\tilde{F}_2$ from [200, 600] [Mflops].

The number of an admissible set has an upper bound equal to $I^{V}J^{I}$. Figure 12 shows the set of assignment evaluation for two criteria: the load of the bottleneck computer and the system performance. It has been found by the exhausted searching of the search space. The point y^o is an ideal point for the presented space with the best possible coordinates for optimisation preferences. It is not included to the criterion space. An anti-ideal point y^- with the worst possible coordinates for optimisation preferences is not in the criterion space, too. Small triangles represent efficient points.

5 Convergence and tabu-mutation

Let the Pareto points $\{P_1, P_2,..., P_U\}$ be given. If the AMGA finds the efficient point $(A_{u1}, A_{u2}, P_{u3}, A_{u4})$ for the cost of computers P_{u3}, that point is associated to the uth Pareto result $(P_{u1}, P_{u2}, P_{u3}, P_{u4})$ with the same cost of computers.

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

$$S = \sum_{u=1}^{U} \sqrt{(P_{u1} - A_{u1})^2 + (P_{u2} - A_{u2})^2 + (P_{u4} - A_{u4})^2}. \quad (12)$$

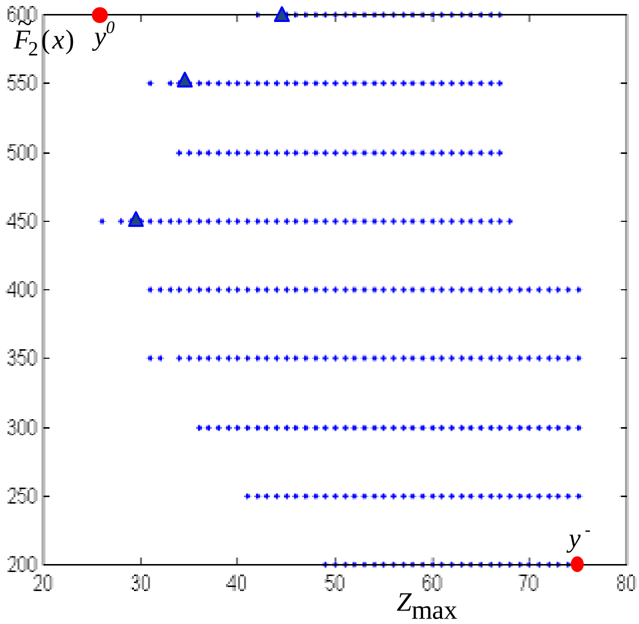

Fig. 12. Set of assignment evaluation for two criteria: the load of the bottleneck computer and the system performance

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

We suggest an introduction a tabu algorithm [10] as an advanced mutation operator, and procedure should be added to the line 14 (Fig. 7), 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 [3] 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. 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.

Similar approach is based on genetic programming. Several practical problems has been solved by the genetic algorithm that operates on the population of computer programs. Koza has solved the problem of finding the global minimum for the following function [14]:

$$f(x_1,\ldots,x_5) = (x_1-1)^2 + \left(x_2-\sqrt{2}\right)^2 + \left(x_3-\sqrt{3}\right)^2 + (x_4-2)^2 + \left(x_5-\sqrt{5}\right)^2$$

The desired global optimum value is 0 and it occurs at the point $\left(1, \sqrt{2}, \sqrt{3}, 2, \sqrt{5}\right)$. The terminal set for this problem consists of the ephemeral random floating-point constant atom $\mathcal{R}$. Whenever the ephemeral random constant is chosen for any endpoint of the tree during the creation of the initial random population, a random number of a specified data type in a specified range is generated and attached to the tree at that point [14]. So, the initial random population contains a variety of different random constants. Once generated and inserted into an initial random program, these constants remains fixed. When we create floating-point random constants, we use a granularity of Δr=0.001 in selecting real numbers within the specified range. Then, random constants are moved around from tree to tree by the crossover operation.

6. Concluding remarks

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 and a negative selection mechanism 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:

1. Ameljanczyk A., *Multicriteria Optimization*, WAT Press, Warsaw 1986
2. Balicki, J.: *Tabu-based Evolutionary Algorithm with Negative Selection for Pareto-optimization in Distributed Systems***.** Proc. on the 7th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering And Data Bases, February 2008, Cambridge, pp. 327-332
3. Balicki J., *Immune Systems in Multi-criterion Evolutionary Algorithm for Task Assignments in Distributed Computer System*. LNCS, 3528, Springer, Heidelberg, 2005, pp. 51—56
4. Burkhardt T., Lohmann K. (Eds.), *Banking und electronic commerce in Internet*, Berlin Verlag 1998.
5. Chissick M., Kelman A., *Electronic commerce: law and practice.* Sweet - Maxwell, London 2000.
6. Chorafas D., *The Commercial banking handbook*, McMillan Business, London 1999.
7. Cronin M.J. (Ed), *Banking and finance on the Internet*, John Wiley &Sons, New York 1997.
8. 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
9. Coello Coello C. A.: *A Comprehensive Survey of Evolutionary-Based Multiobjective Optimisation Techniques*. Knowledge and Information Systems. An International Journal, Vol. 1, 1999, pp. 269-308
10. Glover F., Laguna M.: *Tabu Search*. Kluver Academic Publishers, Boston 1997
11. Hansen M. P.: *Tabu Search for Multicriteria Optimisation: MOTS*. Proceedings of the Multi Criteria Decision Making, Cape Town, South Africa 1997
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 Springer-Verlag, 2001, pp. 241-255

13. Kafil, M. Ahmad, I.: *Optimal Task Assignment in Heterogeneous Distributed Computing Systems*. IEEE Concurrency, Vol. 6, No. 3, 1998, pp. 42 - 51
14. Koza, J.R.: *Genetic programming*. The MIT Press, Cambridge 1992
15. Michalewicz, Z.: *Genetic Algorithms + Data Structures = Evolution Programs*. Springer Verlag, Berlin Heidelberg New York 1996
16. Pietrzak E.: *Exchange rate policy, foreign exchange market, and derivatives, state-of-the-art and perspectives*. Transformacja Gospodarki, Vol. 88, Gdansk 1997
17. Sheble, G. B., Britting, K.: *Refined Genetic Algorithm – Economic Dispatch Example*. IEEE Transactions on Power Systems, Vol. 10, No. 2, 1995, pp. 117-124
18. Srinivas N., Deb K.: *Multiobjective Optimisation Using Nondominated Sorting in Genetic Algorithms*. Evolutionary Computation, Vol. 2, No. 3, 1994, pp. 221-248
19. 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
20. Weglarz, J., Nabrzyski, J., Schopf, J.: *Grid Resource Management: State of the Art and Future Trends*. Kluwer Academic Publishers, Boston 2003
21. Wierzchon, S. T.: *Generating optimal repertoire of antibody strings in an artificial immune system*. In M. Klopotek, M. Michalewicz and S. T. Wierzchon (eds.) Intelligent Information Systems. Springer Verlag, Heidelberg/New York, 2000, pp. 119-133
22. Zitzler, E., Deb, K., and Thiele, L.: *Comparison of Multiobjective Evolutionary Algorithms: Empirical Results*. Evolutionary Computation, Vol. 8, No. 2, 2000, pp. 173-195