dr hab. inż. Tadeusz Witkowski
The Faculty of Production Engineering, Warsaw University of Technology, Poland
Arkadiusz Antczak
The Faculty of Electrical Engineering and Electronics, Łódź University of Technology, Poland
mgr inż. Tomasz Witek
The Faculty of Production Engineering, Warsaw University of Technology, Poland

THE APPLICATION OF GENETIC ALGORITHMS AND AN IMMUNE SYSTEMS APPROACH FOR JOB-SHOP SCHEDULING

The paper presents contemporary artificial intelligence tools – genetic algorithms (GA) and immune algorithms designed for the optymalisation of the production task scheduling problem for multi-asortment short-series production. The essential idea of SZEZA method has been presented. SZEZA method has the ability of part control of the sequence of tasks. The efficiency of SZEZA has been compared with the efficiency of GA and with efficiency of other metaheuristic methods. Problems related with the further development of mentioned algorithms for the purpose of a choice and defining their optimal decision factors and constraints have been described.

1. INTRODUCTION

Genetic algorithms represent a type of randomized local search heuristics inspired by natural selection in biology. Roughly speaking, a genetic algorithms behaves by maintaining a population of solution which evolves through a set of generations. At each step, the algorithm generates a new population of maximum size N from the current one by applying a certain set of randomly applied operators which make it possible to eliminate old solutions and generate new ones. Such a procedure presents an inherent parallelism, due to fact that many different solutions can be tested and modified and the same time. In particular, a new generation is usually derived through the following three phases: evaluation of fitness, selection and generation of new individuals. New generation are derived as long as prespecified stopping criterion is verified, such as when there is no improvement of the best solution between two consecutive generations or after a given number of generations. The construction and the method of an activity of the artificial immune system (AIS) imitates the natural immune system. AIS creates an analogy to immune system but only functions and compose elements can be imitate. The antigen represents requested conditions for the present moment. The set of possible antigens is large and a configuration of antigens which can appear is unknown in advance. The antibody is a list of instructions (it is an algorithm) requested for a creation a solution which meets the constraints described by antigen. The set of possible antibodies is usually not large but there are mechanisms of the agregation and the recombination of them which allow to create new antibodies.

Standard models of the antibody are collected in the library as the system memory. The adjustment (the adaptation) we define as the matching antigen and the antibody. The result of the matching is ideal if the antibody allows to generate a result which meets the requirements of the antigen. If it is not the result is some measure of the error between the achieved result and the requirement. If the matching is not correct the system has to search for new types of antibodies. It usually uses the evolution procedures. AIS and GA are often used for the solving of optimalisation problems. Series of chromosomes and antigens are generated in that case. Their requirements are more strict from the point of view of the goal function. AIS is more frequently used for the restoring of the optimalisation and the commition of results when the input data have been disturbed. For example it is especially important for production systems applications when its results need to be frequently corrected.

2. STATEMENT OF THE JOB ORDERING PROBLEM

Consider a set of jobs $Z = \{Z_i\}$, $i \in I$, where $I = \{1, 2, ..., n\}$ is an admissible set of details (nodes), $U = \{u_k\}$, $k \in \overline{1, m}$, is a set of executors (machines, worksites). Each job Z_i is a group of details Π_i of equal partial task p_i of a certain range of production. Operations of the technological processing of the i th detail are denoted by $\{O_{ij}\}_{j=\xi}^{H}$.

Then for Z_i, we can write $Z_i = (\Pi_i \{O_{ij}\}_{j=\xi}^{H_i})$,

where $O_{ij} = (G_{ij}, t_{ij(N)})$ is the j th operation of processing the i th group of details; ξ_i is the number of operation of the technological process at which one should start the processing the i th group of details; H_i is the number of the last operation for a given group; G_{ij} is a group of interchangeable devices that is assigned to the operation O_{ij}; $\overline{G_{ij}}$ is the i th technological route being considered as a sequence of the groups of devices; G is a set of all groups of devices arose in the matrix $\|\{Zi\}\|$; $t_{ij}(N)$ is an elementary working time (duration in minutes) of the operation O_{ij} with one detail d_i that depends on the number of machine N in the group (on the specified operations); t'_{ij} is the duration of tuning before the operation O_{ij}; $M = \max_i H_i$ is the number of "generalized" operations; N_{gr} is the number of all groups of machines.

It is required to construct a quasioptimal plan-schedule H* that is given in the form of a matrix $\{t_{ij}^{p}, S_{ij}, t_{ij}^{k}\}$, $i \in \overline{1, n}$; $j \in \overline{1, M}$ (where t_{ij}^{p} and t_{ij}^{k} are instants of the beginning and the termination of the operation O_{ij}; $S_{ij} \in G_{ij}$ is the number of a specific machine assigned to the operation O_{ij}) and in the form of a vector of permutations (σ_1, σ_2, ..., σ_n), where $\sigma_i = (d_{1i}, d_{2i}, ..., d_{in})$, each of which assigns the order of starting of groups on the generalized operations. A certain function F(H) serves as a numerical test for estimating the plan H. A plan H* is called quasioptimal if $F(H^*) \le F(H)$ for all or almost all $H \in \Psi$, where Ψ is the set of admissible real plans. We use the Johnson criterion (minimal total time of performance of operations). The desired plan H* (ordering of production operations) should satisfy certain constraints [13].

3. VARYING ORDERING SCHEMES VIA THE CHOICE FUNCTION IN SZEZA

There exist various definitions of a membership function. Therefore, they should be interpreted using the real base of this concept, its sources, and real processes; moreover, the further use of the membership function should be taken into account. Below we examine the application of the membership function of the type "ordering the objects starting from a given sequence to its inverse" for the decision making in job ordering problems.

Now we present a method of randomization of preference rules. First, we choose a complex preference rule. The union of rules is understood in the sense that if a certain rule does not give an answer in the process of ordering details, then the following rule from the set of rules is applied instead of it, and so on. The resultant permutation σ_j^Q is created as an composition $\sigma_j^Q = \sigma_j \circ \sigma_j^R$, where " $\circ$ " is the symbol of composition of two permutations and σ_j^R is a random permutation.

The permutation σ_j^R is found using a statistical simulation method and is composed of the numbers {1, 2,..., n}. These trials are organized by tossing a point to a unit interval, which is divided into subintervals whose lengths form a geometrical progression with denominator Q_j; for each number of operation j $\in$ 1, M, we have a specific value Q. Subintervals that contains these points are removed at once, and their numbers are written to the permutation σ_j^R being constructed; the subintervals remained are shifted one to another. The length of the first subinterval is equal $\xi_1 = 1 - Q/(1 - Q^n)$. The dependence of the denominator on the number j of operation in the proposed randomized algorithm is given in the form of function $Q(j)$. Now we describe methods for control the length distribution of randomized intervals via the choice of the function $Q(j)$. Denote by J_T and J_{-T} the identical and the inverse permutations, respectively. We have analytically proved that

(a) the less is the denominator of the progression Q, the greater is the probability that $\sigma_j^R = J_T$ is chosen;

(b) for Q = 1, the permutation σ_j^R can be obtained with equal probability;

(c) the greater is Q, the greater is the probability to obtain σ_j^R.

The presented algorithm for generating reference rules (ordering schemes) are then used for estimation of schedules using the values of the test function; in our case, this function is the minimal total time of performance of all the jobs. Functions linear by sections have been used for the optimalisation of the choice of the order inside the space of possible results generated by SZEZA method. Values of $q(j)$ which have been the best for particular operations have been defined there. In a case of small tasks it was difficult to find a relation which defines the best schemes of the order. But such relation exists in a case of larger tasks (i.e. for 10 tasks and 27 machines). Decision factors of SZEZA method are variables such as: the method of a generating of single sequent ions (for operations), the form of a function for a choosing of sequentions of tasks (for a process). Parameters of SZEZA method are: the value of parameters $q(j)$ which controls the choosing of sequention for particular operations, the number of tests N_x for $q(j)$ on base operations (mostly first, central, last), the number of changes of sequentions of tasks, the start value of the generator of random numbers.

4. DESCRIPTION OF GENETIC ALGORITHM AGHAR

4.1 Steps of algorithm AGHAR

Algorithm of study of schedules AGHAR hugs following step:.

0. It hugs all preparatory actions, in this introduction of complete entrance data

1. Attributing of value hugs zero to changing N representing meter of population of (schedules)

2. Drawing of initial population P (n), which depends definite parameter of programme of quantity of random admissible schedules on draw out by lot

3. $P(n)$ estimates initial population. It depends on opinion of gathering of schedules in initial population (random). Opinion such depends every schedule of total being number on subordinating value of Johnson criterion of (total time of realization of productive passed order in minutes)

4. End? In this of step algorithm checks whether meter of population reached maximum value. It it was been possible here to realize different any condition of end of calculations (for example programme permits to finish simulation in any moment)

5. n:= n+1 enlargement of meter of population marks

6. Choose new population $P(n)$ from population $P(n\text{-}1)$. Operation this depends on choosing of new population from at present possessed population. Newly choose population replaces old (programme records old population in front of removal)

Selection of new population runes on following rules: probability of choosing of schedule to new population is proportional to value of counted of schedule opinion in step third (rule of roulette). It exists many different methods of selection of new population, e.g. : elitist method

7. Change population $P(n)$ (operation of mutation and crossings). It depends on use definite parameters of programme of quantity of genetic operation in population at present processed schedules. Applied operations genetic this: mutation of sequence of operation, mutation orders of machines as well as crossing of schedules

8. Estimate population $P(n)$. Operation this is the same how action from step of third's algorithm (programme executes this myself procedure)

9. End. It marks end of calculations of programme and lock of entrance files.

4.2 Building of chromosome and algorithm of reservation of machine

Chromosome folds letter from two related with itself. First list contains natural numbers. In front of beginning of calculations programme assigns every hugged part productive order one natural number. Given natural number steps out on this list so many times, what given part has technological operation. Size of this list is even total number of necessary operation to realization of order. This list we call sequence of operation. Thus chromosome introducing given schedule possesses even length of quantity of everybody of executed technological operation. At what of every attributed part is suitable machine from group technological of exchangeable machines. Chromosome it was been possible to introduce in following form scheme (fig. 1).

1	2	1	3	2	1	2	2	1	Number of part

7	2	6	4	1	3	8	4	3	Number of machine

Figure1. Folding chromosome oneself from two related with itself letter

Diagram this we read from left to right side. So e.g. figure 1 in upper government marks party of part of type d_1 . Because it is this first one from left side, this she marks simultaneously first on this part. For operation of first part d_1 i.e. O_{11} (O_{ij} – j - this operation of realization i- part) we execute temporary reservation on machine S7).
Second list is list of well been ordered threes (CID, NO, MID), from which first (CID) introduces number of technological operation represents natural number, representing part, second (NO) for this part, meanwhile third (MID) is number natural representing used machine to realization of this operation. Chromosome (genotype) represents infinitely many schedules. Function of reservation serves (fenotyp) to change of chromosomes into single schedule. Reservations of machine it were been possible to introduce t_p and final time t_k of occupation of machine to j behind help of initial time - this operation of realization and - this part of (party). Function of reservation possesses essential feature: she be able to into effective way to exchange that is internal representations chromosome into external representation that is schedule.
Schedule this possesses sure proprieties:

1. this is schedule realizing set productive order
2. this is admissible schedule
3. this is schedule, which it does not possess superfluous pauses, which they can be eliminate through earlier beginning of operation.

4.3 Genetic operations, repair chromosome and tuning algorithm

Genetic operations of crossing or mutation type have been precisely presented in [3,17]. Because of that only the procedure of the repairing of forbidden chromosomes created as the result of crossing or mutation type is presented. For example lets presume chromosome which represents three parts and each parted is proceeded as follows: three operations (C1), three operations (C2) four (C3) operations. After crossing and mutation the chromosome will have form as fig. 2 presents.

Position on letter	1	2	3	4	5	6	7	8	9	10
Part (CID)	**C1**	**C1**	C2	C1	C1	C1	C3	C3	C3	C3
Operation (OID)	**O1**	**O2**	O2	O1	O2	O1	O1	O2	O3	O4

Figure 2. Form of descendant's chromosome

In case when inadmissible of sagoes chromosomes function of repair repairs it so, to to get admissible chromosome as closest original. For example (fig. 3) chromosome first after repair looks as follows:

Position on letter	**1**	**2**	3	4	5	6	7	8	9	10
Part (CID)	**C1**	**C1**	C2	C1	C2	C2	C3	C3	C3	C3
Operation (OID)	**O1**	**O2**	O1	O3	O2	O3	O1	O2	O3	O4

Figure 3. Chromosome after repair

Optimal parameters of the algorithm such as the scale of the population, the probability of the mutation and the probability of crossing have been defined during the harmonisation of algorithm AGHAR. This definition needs the predefinition of the method of the mutation, the crossing and the selection and the predefinition of the structure of the data for the problem.

5. APPLICATION AN IMMUNE SYSTEMS APPROACH

5.1 The characteristic of immune diversifical algorithm

In works [8] was proposed system PRAIS which uses the immune algorithm diversifical. The reason of the choice of this mechanism of modifying antibody was be faster convergence of the diversifical algorithm in cooperation with other ones for example with algorithm of expression of genes [13]. The diversifical algorithm was introduced by Forrest and her collaborators [5, 12]. In this algorithm dependent on evolution population folds contains binary chains about length l and single chain represent antibody. Antigens (patterns) are represented through binary chains about lengths l. As degree of adjustment of antibody p to antigen a was chosen Hamming's distance, between representing chains. This algorithm tests population of antibodies what differs it from other algorithms which tests population of antigens. Authors have put two questions: in what way it is possible to produce universal (and so recognizing any antigen) antibodies, and which conditions have to be fulfilled that they turn into being specialized antibody (recognizing small number "of similar" antigens).

This diversifical algorithm of the calculation of the adaptation of an antibody - is based on the best adaptation strategy, which is used in classifying systems:

1. Repeat n /A/ times (usually =3)
2. Choose antigen at random a $\in$ A
3. Take random sample S which will include G antibodies (A)
4. For every antibody p $\in$ S count adjustment h (a, p)
5. Choose antibody p* $\in$ S about the highest adjustment
6. Modify his adjustment f(p*) = f(p*) + h (a, p*).

To modify population of antibodies with the counted adaptation usually have been used the genetic algorithms. Authors also because of that fact have applied genetic algorithm AGHAR with following different parameters: the size of population of antibodies $|P|$, the size of population of antigens $|A|$, the size of sample σ, the length of chain l, probability of crossing p_c, the probability of mutation p_m.

The diversifical algorithm possesses the row of interesting property. Firstly it permits to uncover different patterns creating population of antigens. Secondly, in dependence from a size of sample σ, algorithm produces or universal, or specialized antibody. At small values of parameter σ are produced universal antibodies, and at large values of parameter σ - specialized antibodies. At indirect values of this parameter both types of antibodies come into being. In practical uses can be important the number of recognized antigens through particular antibody (it should be enough universal). Requested to [13] name number of different recognized antigens through single antibody - degree of covering. On the other hand it can be important supporting of population of different antibodies too, provided that antibodies from different population should recognize the same antigen (this phenomenon–redundancy). For example in the job-scheduling it has to be worked out schedule in such way, that it will be possible to adapt it to current situation comparatively easily. Then degree of covering measures the number of common patterns for large number of situation. Simultaneously in single schedule one should to find such a sequences of activities which are typical for different situations; the number of this sequences is the degree of redundancy. In order to control degrees of coverings and redundancy Hart and Ross [8], modified algorithm (A) having accepted, that instead of a random single choice of an antigen $a \in A$ it is chosen at random without repetitions α antigens from the set A.

5.2 Immunological algorithms to generate flexible schedules

The additional difficulty to project scheduling is necessity of applying of standard genetic operators in case of dissolving such assignments with genetic algorithms.

The most of works concerning ranking of assignments look for such a schedule that minimize particular criterion, e.g. time of standstill of machines. However such optimum solutions can be sensitive even, to small changes of parameters of assignment. For example, caused by independent factors extension of time of duration of one operation can cause total break down of a schedule. To avoid such a situation, in work [9] was proposed utilization of immune algorithm for flexible schedules. Identified set of disturbances influencing onto realization of unit stages is here seen as a set of antigens, and antibodies form set of adapted plans for disturbances. Immune system is responsible for efficient generating plans in case of potential disturbance in their realization. It was accepted, that workshop produces parts, which demand different programs of processing. To create schedules quickly adapting to its current situation, was accepted that is accessible the set of schedules, worked out earlier and from them - how from blocks – is creating the updated plan of realization of tasks. Genes are in this case short sequences of activities (taken from accessible and checked schedules), from which it is been possible to generate desirable plan. Obviously, to estimate adaptation of single genotypes it should be work out appropriate representation of genes previously. In system PRAIS which is for evolution of antibodies (schedules), antigens are chains about lengths j that represent assignments, which are realized on a machine m. We remind, that j means the number of assignments to be done on a machine m. Chain positions are filled in integers identifying specific assignment. The set of antigens defining assignment creates universum of antigens.

Antibodies are represented by sequences of integers about lengths l, it was accepted that length l is significantly smaller j than the number of assignments. This assumption was based on presumption, that short antibodies are able possible to fit to antigens more effectively. Moreover, it is been possible to identify common characteristics easily. Way of calculation the adaptation of antibody to antigen is based general on applied rules of diversifical algorithm (A). To find adjustment of antibody to antigen was applied rule approximate to one described by equation:

$$m_{ij} = \sum_{k} G\left(\sum_{n} \left| e_i(n+k) - p_j(n) \right| - s + 1 \right) \quad (1)$$

where $e_i(n)$ – value of n th bit of i th epitop, $p_j(n)$ – value n th bit of j th paratop, s- level of adjustment , and $s \leq \min(l_e, l_p)$. If both strings have not less than s complementary bits it means that the paratop has recognized an alien epitop.

In GA, that modifies population of antibodies, were applied three types of crossing dependent on relations between antibodies creating couple of parents. Let S means set of antibodies modified by system PRAIS. "Blocks" are his elements, from which will be created proper schedules. For that purpose was designed system CLARISA [8] using one from three methods: simple recombination, somatic recombination and adding of assignment. System CLARISA adds antibodies until the moment, when or was created schedule of all assignment, or no antibody from the set S can not be already add.

6. THE COMPUTER'S EXPERIMENTS

The comparison of effectiveness of algorithm SZEZA was one of aims of work out of the list of algorithms such as AGHAR. Compared algorithms were constructed for the production system presented below. The system produce corps of speed reductors. Corps differs by some construction's solutions, for example the one of important difference is size of corps. Each task contains a consignment of elements of d_i type, which has to be performed before the defined term. The input data are: the matrix of the groups of technologically equivalently machines (processors), the matrix of technological routes, the matrix of technological operations exact to a group of technologically equivalently machines, the matrix of a labour consumption t_{ij} i-th operation, the matrix of a labour consumption of a rearming of machines before a proceeding j-th operation and i-th element. Articles [14-20] present results of studies of analysed problem. Own and implemented methods (partly modified because of its specific character) have been used for analysis. The best results (because of Johnson criterion) have been achieved for algorithms listed below (list presents an estimate order and by increasing of effectiveness):

1. Random algorithm (Monte Carlo method)
2. Complex rules of priority
3. SZEZA (method which controls a task sequence)
4. A Greedy Randomized Adaptive Search Procedure (GRASP) [2]
5. Genetic algorithm AGHAR
6. Hybrid algorithm (GRASP + CSANN) [2, 21].

Methods such as: Theory of Constraints (TOC), Tabu Search (TS), Hopfielda model and AIS (presented in this article) are implemented contemporary for the purpose of the comparison with method mentioned above. It is necessary to mention that valuations of effectiveness of particular methods presented by literature refer to a type of a production system. For example it use to be pointed that TSAB (Tabu Search Algorithm with Backtracking) [7] is very good algorithm for the flow production problem. It is possible thanks to the reduction of the size of neighbourhood and thanks to the effective construction of a searching and function of aspiration. Other papers show effectiveness of GA for an analysis of a large problem in compare with TS. The studies presented in [10] show benefits of algorithms based on Artificial Neural Networks (ANN) for the task's order on a single machine. Then immune algorithms, relatively fresh, seem to be not less effective than GA [13]. Also very good results have been achieved in [19] thanks to algorithm based on Constraint Satisfication Adaptive Neural Network (CSANN) [2] for the serial flow of the production and serial – parallel flow of the production by hybrid algorithm (GRASP + CSANN). The application of algorithm CSANN for the solving of the problem defined in article has been presented in [19] and the precise explanation of it has been delivered in [20].

7. CONCLUSIONS

The short presentation of algorithm AGHAR and its comparison with other random algorithms such as SZEZA which partly controls the sequence of tasks shows positive results. Possibilities of both algorithms – AGHAR and SZEZA can be extended perhaps. The first of them can be improved thanks to better analysis of decision factors. The second of them can be improved thanks to better analysis of the space of possible solutions using other functions of selection of parameters which controls the sequence of tasks. The possibility of further extension of effectiveness of described algorithms is fixed with an application of a genetic and an immune algorithms for an optimalisation of functions used for a searching inside the space of possible results. The role of AIS is tied with the fact that they belong to so called "new engineering" and they are very elastic, flexible and resistant for small disturbances with compare to traditional method.

LITERATURE

1. Askin R. G., Goldberg J.B.; Design and analysis of Lean Production Systems, John Wiley, New York 2002.
2. Binato S., Hery W.J., Loewenstern D.M., Resende M.G.C., A greedy randomized adaptive search procedure for job shop scheduling, AT&T Labs Technical Report 2000, pp.1-19.
3. Elzway S., Witkowski T., Methods construction of genetic algorithms to scheduling production, in: Modeling and optimization. Methods and application, Akademicka Oficyna Wydawnicza EXIT, Warsaw 2002, V33-V42.
4. Fang H.L., Genetic algorithms in timetabling and scheduling, A dissertation, Departament of Artificial Intelligence University of Edinburgh 1994.
5. Forrest S., Javornik B., Smith R.E., Perelson A.S., Using genetic algorithms to explore pattern recognition in the immune system. Evolutionary Computation, 1, 1993, pp. 191-211.
6. Goldberg D.E., Genetic algorithms and them of use, WNT, Warsaw 1995.

7. Grabowski J., Nowicki E., Smutnicki C. A block approach w zagadnieniach szeregowania zadań, Akademicka Oficyna Wydawnicz ELIT, Warsaw 2003.
8. Hart E., Ross P., The immune system approach to scheduling in changing environments, GECCO ' 99, pp. 1559-1566.
9. Hart E., Ross P., Nelson J., Producing robust schedules via an artificial immune system, IEEE World Congress on Computational Intelligence, International Conference on Evolutionary Computing, 1998.
10. Janiak A., Choose problems and algorithms of ranking of assignments and alocation of supplies, Akademicka Oficyna Wydawnicza PLJ, Warsaw 1999.
11. Michalewicz Z., Genetic algorithms + Data structures = Evoluton Programs, Springer-Verlag, Berlin-Heidelberg-New York, 1999.
12. Smith R.E., Forrest S., Perelson A.S., Searching for diverse, cooperative populations with genetic algorithms, Evolutionary Computation, 1, 1993, pp. 127-149.
13. Wierzchoń S.T., Artificial immune systems, Akademicka Oficyna Wydawnicza EXIT, Warsaw 2001.
14. Witkowski T., Scheduling tasks and scheduling production by of Szeza method optimization, Oficyna Wydawnicza Politechniki Warszawskiej, Warsaw 1998.
15. Witkowski T., Decisions in management enterprise, WNT, Warsaw 2000.
16. Witkowski T., Antczak A., Antczak P., Modele sztucznej inteligencji w systemach wspomagania decyzji przedsiębiorczych, w: Nowe możliwości i metody działań innowacyjnych małych i średnich przedsiębiorstw w układach wirtualnych, Wyd. WSE-I, Warsaw 2003, pp. 274-288.
17. Witkowski T., Elzway S., Antczak A., Projektowanie podstawowych operacji algorytmów genetycznych dla planowania produkcji, Probliemy uprawljenja i informatiki 6, 2003, pp. 129-147.
18. Witkowski T., Antczak A., Operation Research and Investigation of Variables and Parameters of the Genetic Algorithm for Optimization of Production Planning, Proceedings XXVI International Workshop Analysis on MDS: Analysis and Management of Ecological, Technogenic and Telecommunication Risk, Kiev 2003, pp. 69-77.
19. Witkowski T., Antczak P., Use of Artificial Neuron Networks for Task of Classification and Optimization of Production Planning, Proceedings XXVI International Workshop Analysis on MDS: Analysis and Management of Ecological, Technogenic and Telecommunication Risk, Kiev 2003, pp. 77-86.
20. Witkowski T., Antczak P., Strojny G., Use Constraint Satisfaction Adaptive Neural Network for Job- Shop Scheduling, Proceedings Conference Automation 2004, Warsaw 2004
21. Yang S., Wang D., Constraint Satisfaction Adaptive Neural Network and Heuristics Combined Approaches for Generalized Job-Shop Scheduling, IEEE Transactions Neural Networks, vol. 11, 2, 2000, pp. 474-486.