HybridAlgorithm Approach To Job Shop Scheduling Problem

Ye Li[1], [2], Yan Chen[2]

GJCST Classification D.4.1

[1]Electronic and Information College, Dalian University of Technology, Dalian,China 116026
[2.] Transportation Management College, Dalian Maritime University, Dalian,China 116026
liye_dlmu@sohu.com

Abstract- **In this paper, we analyze the characteristics of the dynamic job shop scheduling problem when machine breakdown and new job arrivals occur. A hybrid approach involving neural networks(NNs) and geneticalgorithm(GA) is presented to solve the dynamic job shop scheduling problem as a static scheduling problem. The objective of this kind of job shop scheduling problem is minimizing the completion time of all the jobs, called the makespan, subject to the constraints.The result shows that the hybrid methodology which has been successfully applied to the static shop scheduling problems can be also applied to solve the dynamic shop scheduling problem efficiency.**

Keywords- dynamic job shop, neural network, genetic algorithm, hybrid methodology, makespan

I. Introduction

Job shop scheduling(JSP) is usually a strongly NP complete problem of combinatorial optimization problems and is the most typical one of the production scheduling problems[1,2].Unfortunately, most publication in shop scheduling area focuses on the static shop scheduling. Very few of them suggest a comprehensive model and solution to the dynamically job shop problem[3,4].To deal with dynamic scheduling, most researches usually partition the scheduling process into two phases. In Phase 1, they consider the optimization of makespan under idealized conditions; then in Phase 2, they simply deal with reactive scheduling based on some scheduling rules, in case of accidental disturbance. Muhleman et al analyzed the periodic scheduling policy in a dynamic and stochastic job shop system. Their experiments indicated that more frequent revision was needed to obtain better scheduling performance[5]. Church and Uzsoy considered periodic and event-driven rescheduling approaches in a single machine production system with dynamic job arrivals. Their result indicated that the performance of periodic scheduling deteriorate as the length of rescheduling period increased and event-driven methods achieved a reasonably good performance[6]. Subramaniam et al demonstrated that significant improvements to the performance of dispatching in a dynamic job shop could be achieved easily through the use of simple machine selection rules[7].SQ. Liu et al presented a framework to model dynamic shop scheduling problem. Using the proposed framework, a metaheuristic was proposed to solve dynamic shop problem. The result showed that the metaheuristic methodology which had been applied to solve dynamic shop scheduling problem efficiently[8]. Borstjan andPeter proposed an alternativeway to avoid infeasibility by incorporating a repairing technique into the mechanism for applying moves to aschedule. Whenever an infeasible move was being applied, a repairing mechanism rearranged the underlying schedule insuch a way that the feasibility of the move was restored. The possibility of reaching infeasible solutions was, therefore,eliminated on the lowest possible conceptual level[9]. Hiroshi and Toshihiro considered the jobshop scheduling problem of minimizing the total holding cost of completed andin-process products subject to no tardy jobs. A heuristic algorithm based on the shifting bottleneck procedure wasproposed for solving the minimum total holding cost problem subject to no tardy jobs. Several benchmark problemswhich were commonly used for job-shop scheduling problems of minimizing the makespanwere solved by the proposedmethod and the results were reported[10].

Recently, much attention has been paid to applying neural networks or genetic algorithms et al to production scheduling problems. Haibin Yu et al presented neural network and genetic algorithm to solve the expand job shop problem. The GA was used for optimization of sequence and NN was used for optimization of operation start times with a fixed sequence. New type of neurons were defined to construct neural network(CNN). The neurons can represent processing restrictions and resolve constraint conflicts. Combining gradient CNN with GA for sequence optimization, a hybrid approach was put forward. The approach had been tested by a large number of simulation cases and practical applications. It had been shown that the hybrid approach was powerful for complex JSP[11]. Shengxiang Yang et al presented a new adaptive neural network and heuristics hybrid approach for job shop scheduling. One heuristic was used to accelerate the solving process of neural network and guarantee its convergence; the other heuristic was used to obtain non-delay schedules from the feasible solutions gained by neural network. Computer simulations had shown that the proposed hybrid approach was of high speed and efficiency[12]. Hong Zhou and Yuncheng Feng proposed a hybrid heuristics method for $n/m/G/C_{\max}$, where the scheduling rules, such as shortest processing time(SPT) and MWKR, were integrated into the process of genetic evolution. In addition, the neighborhood search technique was adopted as an auxiliary procedure to improve the solution performance[13]. Byung developed an efficient method based on genetic algorithm to address JSP. The scheduling method based on single genetic algorithm and parallel genetic algorithm was designed. In the scheduling method, the initial population was generated through integrating representation and G&T algorithm, the new genetic operators and selection method were designed to better transmit the temporal relationships in the chromosome, and island model PGA were proposed[14]. Dirk and Christian considered a job shop scheduling problems with release and due-dates, as well as various tardiness objectives. The genetic algorithm can be applied to solve this kind of problem. The heuristic reduction of search space can help the algorithm to find better solution in a shorter computation time[15]. Jose presented a hybrid genetic algorithm for job shop scheduling problem. The chromosome representation of the problem was based on random keys. The schedules were constructed using a priority rule in which the priorities were defined by the genetic algorithm. Schedules were constructed using a procedure that generates parameterized active schedules. After a schedule was obtained a local search heuristic that was applied to improve the solution[16]. Guo proposed a universal mathematic model of the JSP problem for apparel assembly process. The objective of this model was to minimize the total penalties of earliness and tardiness by deciding when to start each order‘s production and how to assign the operations to machine. A genetic optimization process was then presented to solve this model. In which a new chromosome representation, a heuristic initialization process and modified crossover and mutation operators were proposed[17]. Masato and Kenichiproposed the modified geneticalgorithm with search area adaptation (mGSA) for solving the jobshop scheduling problem. To show the effectiveness of the proposed method that conducted numerical experiments by using two benchmark problems. It was shown that this method had betterperformance than existing GAs[18]. Young Su Yun proposed a new genetic algorithm (GA) with fuzzy logic controller (FLC) for dealing withpreemptive job-shop scheduling problems (p-JSP) and non-preemptive job-shop scheduling problems (np-JSP).The proposed algorithm considered the preemptive cases of activities among jobs under single machine schedulingproblems. For thesepreemptive cases, they first used constraint programming and secondly developed a new generepresentation method, a new crossover and mutation operators in the proposed algorithm[19].

In those papers, most publications in job shop scheduling area focus on the static shop scheduling problems and seldom takes into account the dynamic disturbance such as machine breakdown and new job arrivals. In this paper, a university mathematical model for dynamic job shop scheduling problem is constructed. The objective of this model is to minimize makespan. In order to solve this mixed- and multi-product scheduling problem, a combinationof a genetic algorithm and a neural network is used to find the optimal solution. The Back-Propagation Neural Network(BPNN) is designed to describe machine breakdown and new job arrivals etc, detecting whether constraints are satisfied and resolving the conflicts by their feedback adjustments. Then the BPNN can generate a feasible solution for the JSP. For sequence optimization and makespan, a GA is employed. The algorithm will then be used to solve the JSP problem of 10 working procedure and10 machines. Though the simulation, it is shown that the approach can be used to model real production scheduling problems and to efficiency find an optimal solution.

II. MODELING THE JOB-SHOP SCHEDULING PROBLEM

In a JSP we have a set N of jobs, $N = \{1,\ldots,n\}$, that have to be processed in a set M of stages, $M = \{1,\ldots,m\}$. At each stage i, $i \in M$ we have a set $M_i = \{1,\ldots,m_i\}$ of unrelated parallel machines that can process the jobs where $m_i \geq 1$. We consider the dynamic job shop case where stages might be skipped. Every job is a chain of operations and every operation has to be processed on a given machine for a given time. The task is to find the completion time of the very last operation is minimal. The chain order of each job has to be maintained and each machine can only process one job at the same time. Once an operation starts, it must be completed; two operations of a job can not be processed at the same time; no more than one job can be handled on a machine at the same time; the same priority level at each operation; there is no setup and idle time; the money value is not considered. The following additional definitions and notations will help in formulating the problem:

i. i : number ofmachines, $i \in \{1, 2, \ldots, m\}$;

ii. j_i : number of operations of machine i, $j \in \{1,2,\ldots,n\}$;

iii. p_{ij} : processing time of operation j on machine i; $i \in \{1, 2,\ldots, m\}$, $j \in \{1,2,\ldots,n\}$;

iv. o_j : sequence and technique restriction of job j ,such as job j passing though machine sequence = $(o_{j1}, o_{j2}, \cdots, o_{jn})$, $o_{ij} \in \{1,2,\cdots,m\}$, $i \in \{1, 2, \cdots, m\}$, $j \in \{1, 2, \cdots, n\}$;

v. t_{ij} : starting time of operation j on machine i;

vi. t_j : completion time of operation j.

vii. $X_{ijk} = \begin{cases} 1 & \text{if operation } j \text{ precedes operation } k \\ 0 & \text{otherwise} \end{cases}$

viii. $z_{ij} = \begin{cases} 1 & \text{if operation } j \text{ is allocated on machine } i \\ 0 & \text{otherwise} \end{cases}$

ix. C_{ij} : the completion time of operation j on machine i

x. $C_{\max}$: makespan, at the end of the production step, is thus of its final operation o_j.

According to above suggestion, parameter and decision variable of problem, the mathematical model is identified as followed:

$$\min C_{\max}$$

$$s.t$$

$$\sum_{j=1}^{n} z_{ij} = 1$$

$$\sum_{j=1, j\neq k}^{n} \sum_{i=1}^{m} X_{ijk} = 1$$

$$t_{ij} + p_{ij} \leq t_{i+1,j}$$

$$t_{ij} + p_{ij} \leq t_{ik} + (1 - X_{ijk})M$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n}(X_{ijk} + X_{ikj}) \leq 1$$

The objective function is to minimize the maximum completion time (makespan). In a job shop environment, how should the jobs be scheduled and how should they be rescheduled when dynamic events occur, so that the makespan is dynamically minimize? In this study, we restrict our attention to two dynamic factors, the machine breakdown and new job arrivals only.

III. BPNN MODEL

Artificial neural networks are parallel computationaldevices consisting of groups of highlyinterconnected processing elements called neurons. Neurons are basic elements of BPNN. Acommon neural cell or neuron is defined by linearly weighted summation of its input signals, and serially connected non-linear activity function $F(T_i)$.

$$T_i = \sum_{j=1}^{n}(W_{ij} X_j) \quad Y_i = F(T_i) \quad T_i(k+1) = T_i(k) + Y_i(k)$$

where W_{ij} is the connection weight of the j th input signal X_j and the i th neuron. T_i is the weighted summation of the i th neuron. $F(T_i)$ is the activity function and Y_i is the output of the i th neuron.Links among neurons are through their weights. They represent the scheduling restriction. They also reflect the adaptation or adjustment to resolve constraint conflicts through proper feedback links, when restrictions are not met. The working orderand start time etc are used as input nodes, and the feedback represents iterative adjustment, and the breakdown and new job arrivals etc are used as output nodes.

In the event of machine breakdown, two scenarios should be resume or the entire job to be taken out from the schedule. For the first case, the unfinished operation usually has priority to be processed first when the machine has been repaired, considering the set up time or other realistic. For the second case, the affected job should be taken out either to be discarded or processed offline. We consider the first case.

To solve the job shop scheduling problem, the BPNN is adopted that can generate a feasible solution. x_1 represents input node, and y_1 represents output node. For example, $x_1 = 0$ represents that all machines are working order, otherwise $x_1 = 1$. $x_2 = 0$ represents that the start time of each operation is above or equal to 0, otherwise $x_2 = 1$. $x_3 = 0$ represents that all the job is processed, otherwise $x_3 = 1$. $y_1 = 0$ represents that new job arrivals don‘t occur, otherwise $y_1 = 1$. $y_2 = 0$ represents that the machine don‘t break down, otherwise $y_2 = 1$. $y_3 = 0$ represents that due dates isn‘t tardiness, otherwise $y_3 = 1$. $y_4 = 0$ represents X_{ijk}, otherwise $y_4 = 1$. $y_5 = 0$ represents z_{ij}, otherwise $y_4 = 1$.

According to built BPNN, it has three input neurons and five output neurons and six hidden neurons. The training sample is as table 1.

Table 1: some BPNN training sample

Sample number	input			output				
	x1	x2	x3	y1	y2	y3	y4	y5
1	0	1	0	0	0	1	0	0
2	0	0	0	1	0	0	0	0
3	1	0	0	0	0	0	0	0
4	0	0	1	0	0	0	0	1

IV. DESCRIPTION OF GA AND BPNN MODEL

This section first gives out the description of two models, which are used to improve the performance of job shop scheduling problem. One is BPNN that is used to accelerate the solving process of JSP and guarantee feasible solution, the other is GA that is used to obtain the global optimal solution from feasible solution with determined order of operations. The BPNN model is set three levels, which I_i is

The $i's$ input ofinput layer and H_i is output of hidden layer and O_i is output of output layer. So WIH_{ij} is weight between input layer and hidden layer and WHO_{ji} is weight between hidden layer and output layer. Secondly the algorithm of hybrid approach of BPNN and GA for job shop scheduling problem is presented as follows:

Step 1-Initialization population P is generated, which is include probability of crossover P_c andprobability of mutation P_m and initializing WIH_{ij} and WHO_{ji}. Real coding is adopted, and initial population is 30.

```
for i=1:10
L=M(i,:);
for j=1:10
L(j)=L(j)+1;
end
M(i,:)=L;
end
NIND=40;
MAXGEN=     200;
GGAP=0.9;
XOVR=0.8;
MUTR=0.6;
[R,Q]=size(P);
[S2,Q]=size(O);
S1=6;
S=R*S1+S1*S2+S1+S2;
```

Step 2-The fitness is defined and sort order, and network individual is selected as the following probability

$$p = f_i / \sum_{i=1}^{N} f_i$$

Then f_i is adaptive value of individual i, and evaluated by error sum of squares.

$$f(i) = 1/E(i) \quad E(i) = \sum_p \sum_k (V_k - T_k)^2$$

```
FitnV=ranking(ObjV);
SelCh=select('sus', Chrom, FitnV, GGAP);
SelCh=across(SelCh,NIND*GGAP,XOVR,WNumber);
        SelCh=aberrance(SelCh,NIND*GGAP,MUTR,WNumber); disp_fqre=100;
        max_epoch=3000;err_goal=0.002;lr=0.01;
        TP=[disp_fqre  max_epoch err_goal lr];[W1,B1,W2,B2,te,tr]=trainbp(W1,B1,'tansig',W2,B2,'purelin',P,O,TP);
```

Step 3-The crossover is operated in the population G_i *and* G_{i+1} according to probability of crossover Pc, so the offspring G_i' *and* G_{i+1}' are generated.

Step 4-The individual G_j is selected randomly according to probability of mutation Pm, so the offspring G_j' is generated.

```
[PVal ObjVSel N]=cal(SelCh,NIND*GGAP,T,M,PNumber,MNumber,WPNumber);    [Chrom ObjV] =reins(Chrom, SelCh,1, 1, ObjV, ObjVSel);
[PVal ObjV1 N]=cal(Chrom,NIND,T,M,PNumber,MNumber,WPNumber);
```

Step 5-

```
if gen==1
    Val1=PVal;
    Val2=N;
    MinVal=min(ObjV);
    end
```

Step 6-If the optimal solution is obtained, stopping the program and the best solution is output, otherwisegoing back to step3.

V. Simulation Study

We take the benchmark $10/10/J/C_{\max}$ problem. The simulation is finished under Matlab environment. Through 200 epochs searching, the fitness goes stabilization. The sum squared error and fitness curve are showed in figure 1.

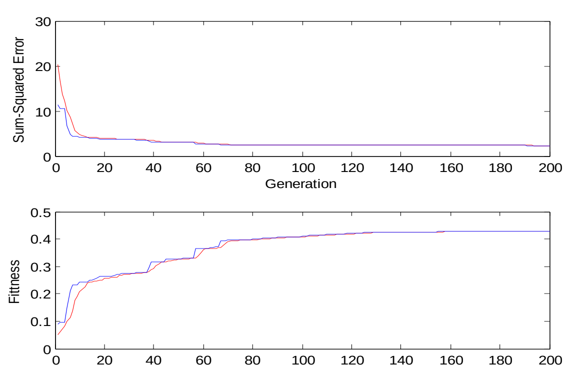


Figure 1:sum-squared error and fitness curve

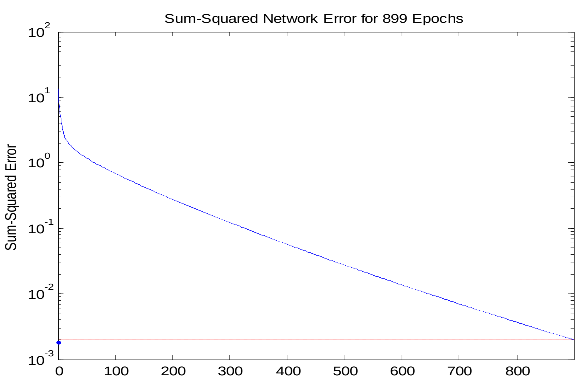


Figure 2: sum-squared network error of BP algorithm

The result of simulation is as follows:[0.0004,0.9930,0.0054,0.0015;0.0013, -0.0036,0.0008,0.0011;0.9998,0.0110,-0.0074,-0.0032;0.0006,-0.0302,0.0230,0.0069;-0.0015, 0.0123, -0.0080,0.9973]. So the idea output is [0 1 0 0;0 0 0 0;1 0 0 0;0 0 0 0;0 0 0 1], and the runtime is 10.6750 seconds.

The sum-squared network error of BP algorithm is showed in figure 2.The error objective is 0.002, and learn rate is 0.01.From the figure, the error objective is convergence to 0.02 when the BPNN algorithm run 899 epochs and the runtime is 13.690000 seconds.

Although the idea result is gotten by weight ofNN that is trained by GA from the above comparing, it takes longtime comparing with BPNN algorithm. Because GA is convergence by heuristic searching such as method of exhaustion, in addition, the complexity of network structure anda large amount of calculated data. For example, the weight of BPNN and threshold number is 58, and the thirty populations are 1740.Such number will be coding, decoding, crossover and mutation, and the dealt data is much greatness. So the searching time is longer. Considering the BPNN is accuracy to seek optimal solution, but it traps into local optimizationeasily. The GA has global searching capacity, and we could combine the GA with BPNN, which show each advantage.

A. Ga-Bp Algorithm

The principle of GA-BP algorithm is the optimal initial value is inherited by GA that focuses at the random position firstly, which is as the initial weight of BPNN. Secondly, it is trained by BPNN.

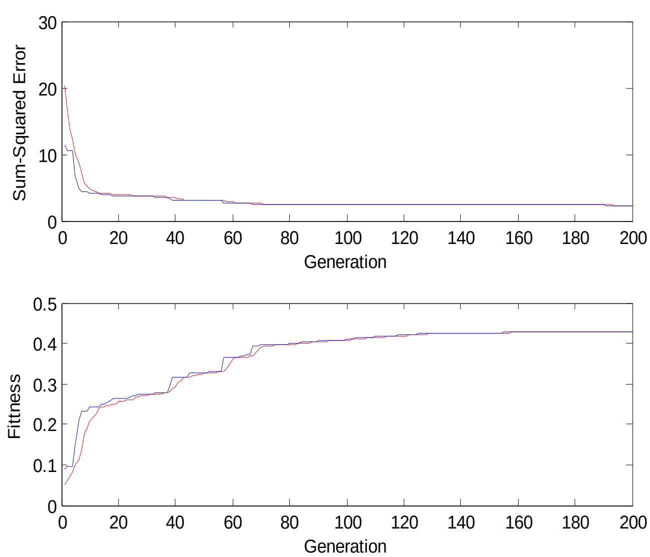
i. The algorithm of hybrid approach for job shop scheduling problem is presented as follows:

Step 1- 5: The same as above, which is NN that is trained by GA

Step 6: The sum squared error is calculated. If the predetermined value(εGA) is obtained, going to step 7, otherwise going back to step 3

Step 7: The optimal initial value is inherited as the initial weight of BPNNby GA. It is trained by BPNN till the predetermined precision ε_{BP}(ε_{BP}<ε_{GA}) is gotten.

B. Experiment result



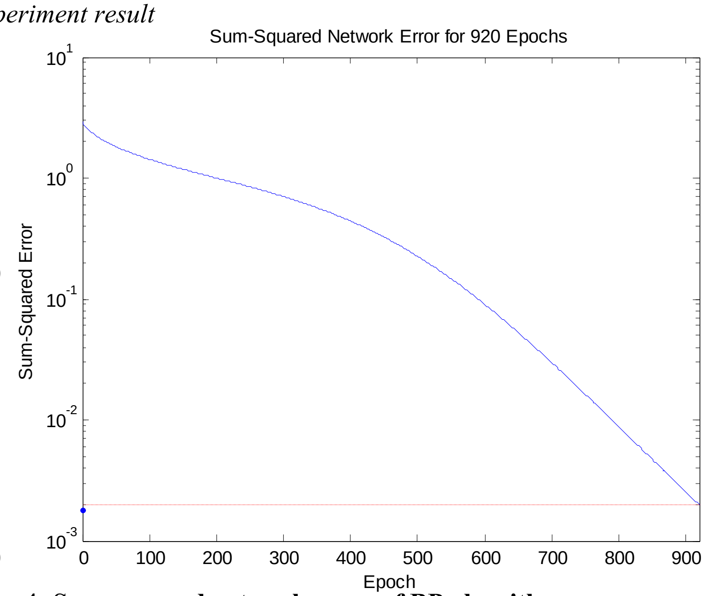


Figure 3:Sum-squared error and fitness curve of GAFigure 4: Sum-squared network error of BP algorithm

The sum squared error and fitness curve of GA are showedfrom figure 3, and the training objective of BPNN is showed from figure 4. We set initial population of GA is 30, and predetermined value is 5. The result of simulation is as follows:[0.0115,0.9647,0.0097,0.0114;0.0039,-0.0018,0.0007,0.0034;1.0025,-0.0076,-0.0034,0.0042;-0.0007,-0.008,0.0041,0.0037;-0.0073, 0.0081, -0.0059,1.0039]. So the idea output is [0 1 0 0;0 0 0 0;1 0 0 0;0 0 0 0;0 0 0 1], and the runtime is 5.739000 seconds.

The objective value is obtained through 80 epochs by GA, and the predetermined precision is convergence by 920 epochs. The run time is 18.326. It is obviously that the GA-BP algorithm is better than BP algorithm that is in convergence rate andRuntime.

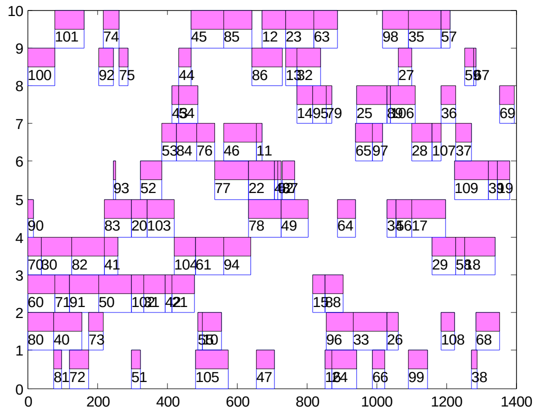


Figure 5: Gantts chart of JSP

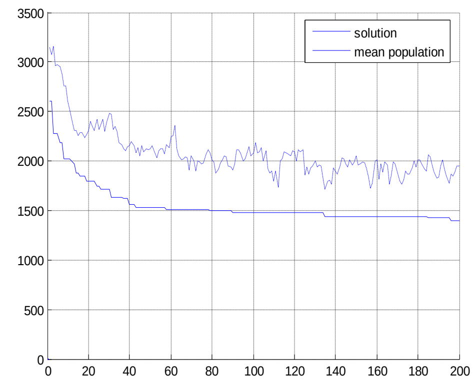


Figure 6:The solution and population curve

The result shows the sequence of each job, and makespan is 1395 from figure 5. The makespan is attained 1395, when the iterative number is 80, and the mean population is randomfrom the figure

VI. . Conclusion

In this paper, we analyze the characteristics of the dynamic job shop scheduling problem, and present a new hybrid approach, combining the BPNN with GA for solving when machine breakdown and new job arrivals occur.The BPNN is used to obtain feasible solution during the iterations. In order to overcome the shortcomings that BP algorithm is usually trapped to a local optimum and it has a low speed of convergence weights. The GA is adapted to the globe optimal searching. This algorithm can effectively and reliably be used in JSP problem. Simulation has shown that the proposed hybrid approach for JSP has excellent performance with respect to the quality of solution and speed of calculation.

VII. Reference

1) HeinzGroflin,Andreas Klinkert. Feasible insertions in job shop scheduling,Short-cycles and stable sets. European Journal of Operational Research, 177(2),2007: 763–785
2) Gerhard J. Woeginger. Inapproximability results for no-wait job shop scheduling. Operations Research Letters, 32(4), 2004: 320–325
3) Liu SQ, Ong HL. A comparative study of algorithms for the flowshop scheduling problems. Asia-Pacific J Operation Research, 19(2),2002: 205-222
4) Wein LM, Chevalier PB. A broader view of the job shop scheduling problem. The Institute of ManagementSciences, 38(7), 1992: 1018-1033
5) Muhleman AP, Lockett AG, Farn CK. Job shop scheduling heuristics and frequency of scheduling. *Int J Production Research*, 20(2), 1982: 227-241
6) Church LK, Uzsoy R. Analysis of periodic and event-driven rescheduling policies in dynamic shops.*Int J Computer Integrated Manufacturing*, 5(3), 1992:153-163
7) Subramaniam V, Lee GK et al. Machine selection rules in a dynamic shop. Int J Advanced Manufacturing Technology, 16(1), 2000:902-908
8) SQ Liu, HL Ong, KM Ng. Metaheuristics for minimizing the makespan of the dynamic shop scheduling problem. Advances in Engineering Software, 36(3), 2005: 199-205
9) Bortjan Murovec, Peter Suhel. A repairing technique for the local searchof the job-shop problem.European Journal of Operational Research, 153(1), 2004:220-238
10) Hiroshi Ohta, Toshihiro Nakatani. A heuristic job-shop scheduling algorithm to minimize thetotal holding cost of completed and in-process productssubject to no tardy jobs. International Journal Production Economics,101(1), 2006:19–29
11) Habin Yu, Wei Liang. Neural network and genetic algorithm-based hybrid approach to expanded job-shop scheduling.Computers & Industrial Engineering, 39(3-4), 2001: 337-356
12) Shengxiang Yang, Dingwei Wang. A new adaptive neural network and heuristics hybrid approach for job-shopscheduling. Computer & Operations Research, 28(10), 2001: 955-971
13) Hong Zhou, Yuncheng Feng, Limin Han. The hybrid heuristic genetic algorithm for job shop scheduling.Computers & Industrial Engineering, 40(3), 2001: 191-200
14) Byung Joo Park, Hyung Rim Choi, Hyun Soo Kim. A hybrid genetic algorithm for the job shop scheduling problem.*Computers & Industrial Engineering*, 45(4), 2003:597-613
15) Dirk C. Mattfeld, Christian Bierwirth. An Efficient Genetic Algorithm for Job Shop Scheduling with Tardiness Objectives. European Journal of Operational Research, 155(3), 2004: 616-630
16) Jose Fernando Goncalves, Jorge Mendes, Mauricio Resende. A Hybrid Genetic Algorithm for the Job Shop Scheduling Problem. *European Journal of Operational Research*, 167(1), 2005: 77-95

17) Z.X Guo, W.K Wong, S.Y Leung et al. Mathematical Model and Genetic Optimization for the Job Shop Scheduling Problem in a Mixed- and Multi-Product Assembly Environment: A Case Study Based on the Apparel Industry. Computers & Industrial Engineering, 50(3), 2006:202-219
18) Masato Watanabea, Kenichi Ida, Mitsuo Gen. A genetic algorithm with modified crossover operator and searcharea adaptation for the job-shop scheduling problem. *Computers & Industrial Engineering*, 48(4), 2005: 743-752
19) Young Su Yun. Genetic algorithm with fuzzy logic controller for preemptive andnon-preemptive job-shop scheduling problems.Computers & Industrial Engineering, 43(3), 2002: 623-644