Study on integrated algorithm of complex multi-product flexible scheduling

Zhiqiang Xie[1,2], Shuzhen Hao[1], Lei Zhang[1] and Jing Yang[2]

[1.] College of Computer Science and Technology, Harbin University of Science and Technology, Harbin, China

[2.] College of Computer Science and Technology, Harbin Engineering University, Harbin, China

xzq0111@tom.com

Abstract—**In order to optimize the processing of complex multi-product flexible scheduling, a simplified method which converts complex multi-product into a virtual single-product is adopted. The complex multi-product flexible scheduling problem is transformed into the virtual single-product flexible scheduling problem. Aiming at the virtual single-product flexible scheduling problem, an integrated algorithm is proposed. This algorithm firstly adopts the allied critical method to confirm the scheduling sequence of operations. Then each operation is simulation processed by the sequencing algorithm on a set of capable machines. Calculate the virtual completion time of this operation and choose the machine which has the minimum virtual completion time. The routing sub-problem and sequencing sub-problem are solved at the same time by the proposed algorithm. Experiment shows that the proposed algorithm can solve complex multi-product flexible scheduling problem with constraint between jobs, it also can solve simple multi-product flexible job-shop scheduling problem with non-restraint between jobs and obtain satisfying result.**

Keywords- flexible scheduling, complex product flexible scheduling, integrated algorithm, simulation process

I. INTRODUCTION

Flexible scheduling problem [1] is an extension of the classical job-shop scheduling problem (JSSP). Each machine has only one processing capacity and there is no constraint between jobs in classical job-shop scheduling problem. However, modern flexible manufacturing system is based on CNC machine tools or machining centers. These machines have many processing capacities because they can change cutting tools and fixtures automatically. Therefore, the scheduling methods of classical job-shop scheduling problem can not solve the problem of flexible manufacturing with constraint between jobs. In flexible scheduling, each operation can be processed on many different machines. So, this kind of scheduling problem reduces machine constraints, and enlarges searching scope of practicable solutions, but it makes the problem intractable, and further increases the problem's complexity [2].

Being an extension of the classical job-shop scheduling problem which is known to be NP-hard [3], the flexible scheduling problem is NP-hard as well. Many different approaches have been proposed concerning the flexible scheduling problem, which are divided two types according to the steps of solving this problem, hierarchical approaches and integrated approaches. In hierarchical approaches, assignment of operations to machines and sequencing of operations on machines are treated separately, assignment and sequencing are considered independently [4-7]. In integrated approaches, assignment and sequencing are not differentiated [8-10]. These algorithms did not take into account the constraint between jobs and can only solve the simple product static flexible job-shop scheduling problem with non-restraint between jobs. In fact, complex product is widespread in the processing and assembly of discrete manufacturing enterprises, for example, there is constraint between jobs during the production process of special electric machinery. Xiong [11] et al. proposed a JSSP with process relativity; however, the flexible scheduling with process relativity is not considered. Aiming at the JSSP with many identical function machines, Xie [12] et al. considered the constraint between jobs, proposed a secondary optimization algorithm and an optimum scheduling algorithm. However, the scheduling problem with many identical function machines is that some operations of job can be processed by any machine in the identical function machine set and the processing time of the operation is the same, so it is the special example of the flexible scheduling problem.

In this paper, we consider the two problems of processing and assemblage at the same time. The major contribution of this paper is to present for the first time a new hierarchical scheduling algorithm to solve the complex multi-product flexible scheduling problem with constraint between jobs. It is shown that the proposed algorithm is feasible and effective.

II. PROBLEM DESCRIPTION

According to literature [13], the flexible scheduling problem may be described as follows:

Let $J=\{J_i\}_{1\leq i\leq n}$, a set of n jobs and a product consists of n jobs through constraints. Each job J_i consists of a predetermined sequence of operation $G_i=\{O_{ij}\}_{1\leq i\leq n}$ where O_{ij} denotes operation j of J_i and n_i is the total number of operations of job J_i. Let $M=\{M_k\}_{1\leq k\leq m}$, a set of m machines. M_{ij} denotes a set of machines, any of which can process operation O_{ij}, and $M_{ij}\subseteq M$. O_{ijk} denotes operation j of J_i, processed on machine M_k and T_{ijk} is its processing time on machine M_k. Let S_{ijk} be its starting time on machine M_k, and E_{ijk} be its completion time on M_k. Let C_k be the completion time of machine M_k.

If the operation O_{eg} and O_{ij} are processed on the same machine, and the operation O_{eg} is processed after the operation O_{ij} is finished, X_{ijeg}=1, otherwise, X_{ijeg}=0; if machine k is selected for the operation O_{ij}, Y_{ijk}=1, otherwise, Y_{ijk} =0.

The objective function is min(max(C_k)), k = 1, 2,…….m. Subject to:

$$C_k = \max(E_{ijk}),\ Y_{ijk}=1,\ i=1,2,\ldots,n,\ j=1,2,\ldots,n_i$$

$$S_{ijk}-E_{i(j-1)n}\geq 0,\ Y_{ijk}=Y_{i(j-1)n}=1$$

978-1-4244-5848-6/10/$26.00 ©2010 IEEE

$S_{egk}-E_{ijk}\geq 0$, $Y_{egk}=Y_{ijk}=1$, $X_{ijeg}=1$

In this paper, we also make the following hypotheses: All machines are available at time 0. At a given time, a machine can only execute one operation; it becomes available to the next operation once the assigned operation is completed. There are sequence constraints among operations of different jobs.

III. IMPROVED PROCESSING OPERATION TREE MODEL OF COMPLEX FLEXIBLE PRODUCT

Definition 1 improved processing operation tree The node of operation on the improved processing operation tree can be processed on many different machines and the corresponding processing times are different.

The values of node in processing operation tree contain three elements that are the sequence number of operation, the needed machine number of operation, and the processing time of operation. Modifying the content of the machine needed number of operation and the processing time of operation can make a node denote an operation which is processed on more than one machine. For example: operation 1 of product A can be processed on both machine 2 and machine 3, the corresponding processing time being 10 and 15 respectively, then the node of the operation can be denoted by A1/2, 3/10, 15.

Aiming at all products which arrive at the same time, the method of aligning the root nodes in literature [14] is adopted to construct the improved virtual processing operation tree. That is to say, the improved processing operation trees of all products are the sub-tree of the virtual processing operation tree. Thus the multi-products are transformed in to a virtual single-product.

IV. ALGORITHM DESIGN OF SINGLE- PRODUCT FLEXIBLE SCHEDULING

The flexible scheduling problem is divided into two sub-problems. The first one is to assign each operation to a machine. The second one is the computation of the starting time and the completion time of each operation. Aiming at these two sub-problems, an integrated algorithm is proposed to solve them at the same time.

A. *Allied critical path method*

Critical path of product is the longest path from leaf node to root node in the improved processing operation tree of product. If the number of the longest path is not unique, then the path which has more operations is the critical path. Allied critical path method fully considers the significant impact that operations in critical path have on the product scheduling, thus achieving the segment scheduling of critical paths by decomposing the improved processing operation tree [15].

Because the end branch of the processing operation tree has the characteristics of cyclic nesting and reproduction, operations on the end branch are divided into dependent operations and the independent ones in order to make a product fully processed on the machine [15]. Operations on critical path and dependent operation clusters are first scheduled, and then independent operations. Finally recursive algorithm is adopted to implement the scheduling optimization of the whole processing operation tree.

In flexible scheduling problem, the processing time of each operation is not unique. Because the object of this paper is to minimize the overall completion time, the possibility which each operation chooses the minimum processing time is the greatest. So we choose the minimum processing time of each operation to calculate the critical path. Then confirm the processing sequence of operations according to the allied critical path method.

B. *The sequencing algorithm*

Operations are divided into dependent operations and independent operations by analyzing their different characteristics. Adopt different scheduling algorithms to process dependent operations and independent operations, we can make the different scheduling methods complement each other and fully consider the parallelism of processing machines.

Because dependent operations have more interaction, forward greedy rule is adopted to process them in order to make operations complete as soon as possible. Forward greedy rule is that, subject to the constraint, operations should be processed on the machine available as soon as possible to get the minimum starting time of operation, so the objective function is $\min(S_{ijk})$[12].

The processing of independent operations has good flexibility because the starting processing time is not restricted. According to the characteristics of independent operations, the scheduling algorithm of shortening idle time[16] is adopted for independent operations, so as to make remaining idle time as short as possible. The scheduling algorithm of shortening idle time is that independent operations are inserted in the position with minimal gap which the processing time of operation is subtracted from idle time of this operation's machine. So the optimal position of inserting independent operation is defined as $d_k=\min(d_{k,n,n+1}-T_{ijk})$, $(n=1, 2\ldots j-1)$, where d_k denotes the optimal inserting position of independent operations, and $d_{k,n,n+1}$ denotes the idle time n between operations processed n and $n+1$ on the machine k.

C. *Design of the integrated algorithm*

Definition 2 simulation process In order to confirm the processing machine of operation, operation is simulation processed on each machine of the set. It does not really arrange the operation on this machine.

Definition 3 virtual completion time The completion time of operation O_{ij} when it is simulation processed on machine M_k is defined as virtual completion time, which is denoted by E_{ijk}.

Operation O_{ij} is simulation processed by the sequencing algorithm on a set of capable machines M_{ij}. Calculating the virtual completion time E_{ijk} and choosing the machine which has the minimum E_{ijk} value. Thereby the processing machine and the processing time of the operation are confirmed. Then the operation is processed on the confirmed machine. The routing problem and sequencing problem are solving at the same time. Each operation chooses the processing machine with the minimum completion time, which can improve the utilization factor of machine and make the total processing time as short as possible.

V. ALGORITHM DESIGN OF COMPLEX MULTI-PRODUCT

Step1. Construct an improved virtual processing operation tree by improved processing operation trees of multi-products.

Step2. Construct a queue, Queue 1. Confirm the scheduling sequence of all operations by the allied critical path method, and save them in the Queue 1 in this order.

Step3. Get the head element from Queue 1. This operation is simulation processed on its set of capable processing machines by the sequencing algorithm. Calculate the virtual completion time E_{ijk} on each machine.

Step4. Comparing the E_{ijk} values of the operation, choose the processing machine which has the minimum E_{ijk} value. If the minimum E_{ijk} value is not unique, choose the processing machine with the shortest processing time. Thus the processing machine and the processing time of this operation are confirmed.

Step5. If this node is the dependent operation, then the operation is processed by forward greedy rule; otherwise, if this node is the independent operation, then the operation is processed by the scheduling algorithm of shortening idle time. When head node is processed, delete this node. Go to Step 3 until Queue 1 is null.

VI. COMPLEXITY ANALYSIS OF ALGORITHM

The algorithm above is implemented by VC++, supposing the total number of operations of product is n, the number of machines is m, each operation can be processed on different machines whose number is k (k<=m), and the complexity of algorithm is analyzed as follows:

(1) The independent operations are scheduled in descending order according to assembly time. Usually the total number of independent operations is $n/2$, and the number of the independent operations on each machine is $n/2m$. If we compare every two independent operations on each machine, the number of comparing is $C_{n/2m}^{2}$ = (n^2-$2mn$)/($8m^2$) in the worst condition.

(2) The idle times are scheduled in descending order according to idle time between operations assembled. Usually the number of dependent operations is $n/2$, the number of the dependent operations on each machine is $n/2m$, and the number of idle time between operations assembled is $n/2m$ at most. If we compare every two idle time on each machine, the number of comparing is $C_{n/2m}^{2}$ = (n^2-$2mn$)/($8m^2$) in the worst condition.

(3) Usually the total number of operations is n, the number of operations on each machine is n/m, the number of comparing is $C_{n/m}^{2}$ = (n^2-mn)/($2m^2$) for each operation to confirm the inserting position on the machine according to the beginning processing time of operation and idle time between operations scheduled. So the total number of comparing is $m C_{n-m}^{2}$ = (n^2-mn)/($2m$) in the worst condition.

(4) Confirm the processing machine of each operation. In order to confirm the processing machine, each operation need be simulation processed on its processing machine set by the sequencing algorithm. Each operation can be processed on all machines in worst condition, the total number of comparing is $nm^2 C_{n/m}^{2}$ =n^3-n^2m.

So the complexity is O (n^3), namely this algorithm can run in quadratic polynomial time.

VII. SCHEDULING EXAPMLES

A. *Example of flexible job-shop scheduling problem with non-restrain t between jobs*

The effectiveness of the algorithm proposed is verified by the example in literature [4]. In the flexible job-shop scheduling problem with non-restraint between jobs, there are 10 jobs with 30 operations to be processed on 10 machines. The processing time of all operations is shown in Table 1. The node of operation O11 is denoted by O11/1,2,3,4,5,6,7,8,9,10/1,4,6,9,3,5,2,8,9,5. However, this node is denoted by O11 in order to express operations conveniently. Other operations are denoted in the same way. The improved processing operation trees of all jobs are shown in Figure. 1. The total processing time is 8 by the AL algorithm in literature [4]. However, the better solution can be obtained by the algorithm proposed and the total processing time is 7. Gantt chart is shown in Figure. 2.

B. *Example of flexible scheduling problem with constraint between jobs*

Product A and B are two complex products with constraint between jobs, which contain 12 jobs. The data of all operations which are processed on 10 machines is shown in Table 2. Product A consists of 7 jobs with 19 operations, the improved processing operation tree of Product A is shown in Figure. 3. Product B consists of 5 jobs with 13 operations, the processing operation tree of Product B is shown in Figure. 4. Constructing a virtual single-product by Product A and Product B, the virtual improved processing operation tree of this virtual single-product is shown in Figure. 5.

Adopt the allied critical path method to confirm the processing sequence of operations. The processing sequence is A4, A5, A6, A1, A2, A3, A13, A14, A15, A10, A11, A12, A7, A8, A9, A16, A17, A18, A19, B4, B5, B6, B1, B2, B3, B7, B8, B9, B10 ,B11, B12, B13.The Gantt chart of operations which have been processed is shown in Figure. 6 when operation B5 is processed. In order to confirm the processing time of operation B5, the sequencing algorithm is adapted to simulation process this operation. Calculate the virtual completion time of operation B5 which is simulation processed on each machine in the set of capable machines. The virtual completion time table of operation B5 is shown in Table 3.

TABLE I. THE PROCESSING TIME TABLE OF ALL OPERATIONS IN EXAMPLE 7.1

		M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
	O11	1	4	6	9	3	5	2	8	9	5
J1	O12	4	1	1	3	4	8	10	4	11	4
	O13	3	2	5	1	5	6	9	5	10	3
	O21	2	10	4	5	9	8	4	15	8	4
J2	O22	4	8	7	1	9	6	1	10	7	1
	O23	6	11	2	7	5	3	5	14	9	2
	O31	8	5	8	9	4	3	5	3	8	1
J3	O32	9	3	6	1	2	6	4	1	7	2
	O33	7	1	8	5	4	9	1	2	3	4
	O41	5	10	6	4	9	5	1	7	1	6
J4	O42	4	2	3	8	7	4	6	9	8	4
	O43	7	3	12	1	6	5	8	3	5	2
	O51	7	10	4	5	6	3	5	15	2	6
J5	O52	5	6	3	9	8	2	8	6	1	7
	O53	6	1	4	1	10	4	3	11	13	9
	O61	8	9	10	8	4	2	7	8	3	10
J6	O62	7	3	12	5	4	3	6	9	2	15
	O63	4	7	3	6	3	4	1	5	1	11
	O71	1	7	8	3	4	9	4	13	10	7
J7	O72	3	8	1	2	3	6	11	2	13	3
	O73	5	4	2	1	2	1	8	14	5	7
	O81	5	7	11	3	2	9	8	5	12	8
J8	O82	8	3	10	7	5	13	4	6	8	4
	O83	6	2	13	5	4	3	5	7	9	5
	O91	3	9	1	3	8	1	6	7	5	4
J9	O92	4	6	2	5	7	3	1	9	6	7
	O93	8	5	4	8	6	1	2	3	10	12
	O10,1	4	3	1	6	7	1	2	6	20	6
J10	O10,2	3	1	8	1	9	4	1	4	17	15
	O10,3	9	2	4	2	3	5	2	4	10	23

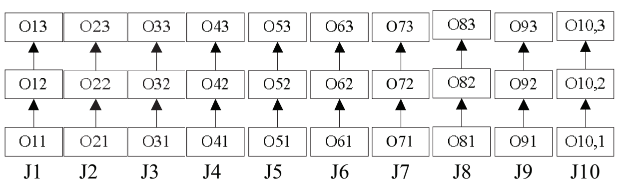

Figure 1. The processing operation trees of jobs

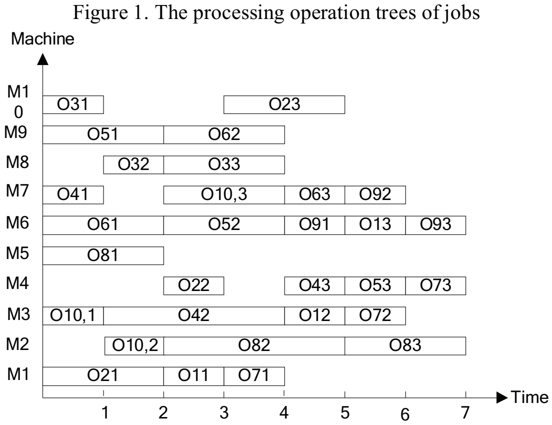

Figure 2. Gantt chart of this example

The machines with minimum virtual completion time E_{ijk} are machine M2 and M7, which is not unique. The processing time is 3 when operation B5 is processed on machine M2, and the processing time is 4 when operation B5 is processed on machine M7. So operation B5 is processed on machine M2. The processing machines and processing times of other operations are confirmed by this method. Gantt chart is shown in Figure.7 after all operations are processed by the proposed algorithm, and the total processing time is 14.

The example and Gantt chart show that, the proposed algorithm is a new algorithm for solving complex multi-product flexible scheduling problem with constraint between jobs. This algorithm has few constraint conditions and satisfying complexity.

TABLE II. THE PROCESSING TIME TABLE OF ALL OPERATIONS IN EXAMPLE 7.2

		M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
	A1	1	4	6	9	3	5	2	8	9	5
	A2	4	3	1	3	4	8	10	4	11	4
	A3	3	2	5	1	5	6	9	5	10	3
	A4	2	10	4	5	9	8	4	15	8	4
	A5	4	8	7	1	9	6	2	10	7	3
	A6	6	11	3	7	5	3	5	14	9	2
	A7	8	5	8	9	4	3	5	3	8	1
	A8	9	3	6	1	2	6	4	1	7	2
	A9	7	1	8	5	4	9	1	2	3	4
A	A10	5	10	6	4	9	5	1	7	1	6
	A11	4	2	3	8	7	4	6	9	8	4
	A12	7	3	12	1	6	5	8	3	5	2
	A13	7	10	4	5	6	3	5	15	2	6
	A14	5	6	3	9	8	2	8	6	3	7
	A15	6	2	4	3	10	4	3	11	13	9
	A16	8	9	10	8	4	2	7	8	3	10
	A17	7	3	12	5	4	3	6	9	2	15
	A18	4	7	3	6	3	4	1	5	2	11
	A19	7	3	4	5	8	14	6	5	10	9
	B1	1	7	8	3	4	9	4	13	10	7
	B2	3	8	1	2	3	6	11	2	13	3
	B3	5	4	2	1	2	1	8	14	5	7
	B4	5	7	11	3	2	9	8	5	12	8
	B5	8	3	10	7	5	13	4	6	8	4
B	B6	6	2	13	5	4	3	5	7	9	5
	B7	3	9	1	3	8	1	6	7	5	4
	B8	4	6	2	5	7	3	1	9	6	7
	B9	8	5	4	8	6	1	2	3	10	12
	B10	4	3	3	6	7	3	2	1	20	6
	B11	3	1	8	1	9	4	1	4	17	15
	B12	9	2	4	2	3	5	2	4	10	23
	B13	4	8	7	9	11	5	4	6	2	3

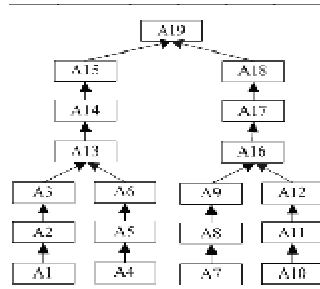

Figure 3. The processing operation tree of product A

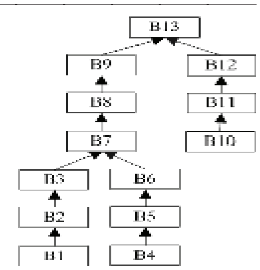

Figure 4. The processing operation tree of product B

VIII. CONCLUTION

In this paper, we proposed the problem model and an integrated scheduling algorithm by analyzing the complex multi-product flexible scheduling problem with constraint between jobs. The proposed algorithm provides a reference method for this kind of problem.

Multi-products are converted into a virtual single-product by adopting the method of constructing virtual improved processing operation tree. This method simplifies the problem; operations are simulation processed by the sequencing algorithm. Choose the processing machine with the minimum virtual completion time for each operation, which enhance the machine's efficiency and minimize the total processing time of all operations; the improved processing operation tree model is adopted to describe the flexible scheduling problem, and the constraint between jobs is considered. Theoretical analysis and experiment show that the algorithm presented can solve the complex multi-product flexible scheduling problem with constraint between jobs in polynomial time. And it also can obtain satisfying result for solving the simple multi-product flexible job-shop scheduling problem with non-restraint between jobs.

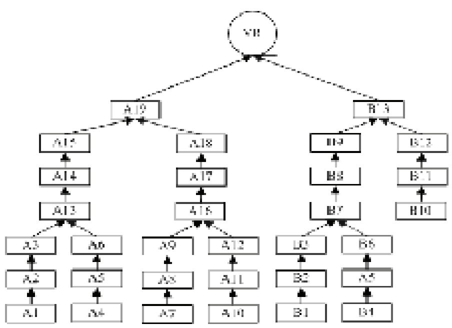

Figure 5 . The virtual processing operation tree

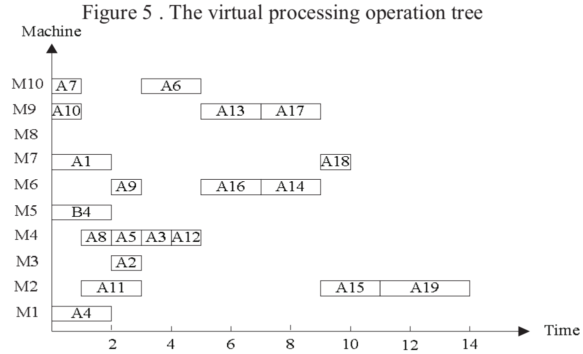

Figure 6. Gantt chart before operation B5 is processed

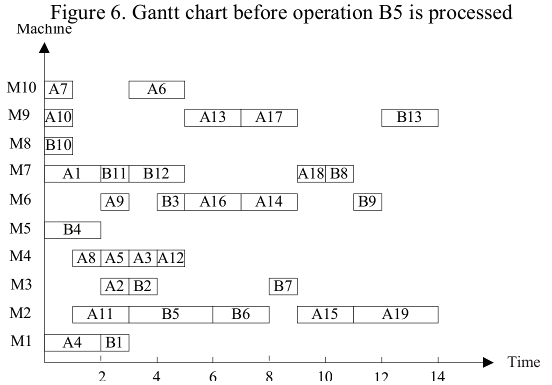

Figure 7. The final Gantt char after all operations are processed

TABLE III. THE VIRTUAL COMPLETION TIME OF OPERATION B5

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
E_{ijk}	10	6	13	12	7	21	6	8	17	9

ACKNOWLEDGMENT

This project is supported by the National Natural Science Foundation of China (No. 60873019), China Postdoctoral Science Foundation and Heilongjiang Province Postdoctoral Science Foundation.

REFERENCES

[1] BRUCKER P, SCHLIE R., "Job shop scheduling with multi-purpose machines," Computing, vol.45, pp: 369-375, 1990.

[2] JANSEN K., MASTROLILLI M. and SOLIS-OBA R., "Approximation Algorithms for Flexible Job Shop Problems," International Journal of Foundations of Computer Science, vol.16,no. 2, pp:361-379,2005.

[3] GAREY M.R, JOHNSON D.S, "Computers and Intractability: A Guide to the theory of NP-Completeness," Freeman, San Francisco, CO, USA, 1979.

[4] KACEM I, HAMMAD I S, BORNE P, "Approach by localization and multi-objective evolutionary optimization for flexible job shop scheduling problems," IEEE Transactions on Systems, Man and Cybernetics, Part C,vol. vol.32, pp:408-419,2002.

[5] ZRIBI N, KACEM I, EL KAMEL A, "Optimization by phases for the flexible job shop scheduling problem," Proceedings of the 5th Asian Control Conference, vol. 3, pp: 1889-1895, 2004.

[6] ZHANG C Y, RAO Y Q, LI P G. "Bilevel genetic algorithm for the flexible job shop scheduling problem". Chinese Journal of Mechanical Engineering, vol.43, no. 4, pp: 119-124, 2007.

[7] XIA W J, WU ZH M. "Hybrid particle swarm optimization approach for multi-objective flexible job-shop scheduling problems". Control and Decision, vol.20, pp: 137-141, 2005.

[8] SUN ZH J, ZHU J Y. "An Intelligent Scheduling Optimization of Job-shop with Flexible Process Routines". Mechanical science and technology, vol.20, pp: 348-350, 2001.

[9] MATI Y., REZQ N, XIE X L. "An Integrated Greedy Heuristic for a Flexible Job Shop Scheduling Problem," The Proceedings of IEEE International Conference on Systems, Man, and Cybernetics, vol.4, pp: 534-2539, 2001.

[10] ZHANG W C, ZHENG P E, WU X D. "Solution to flexible Job Shop scheduling problems with capacitated constraints based on ant colony & genetic algorithms". Computer Integrated Manufacturing Systems, vol.13, no 2, pp: 333-337, 2007.

[11] XIONG H G, LI J J. "Heuristic Method for Dynamic Job Shop Scheduling Problem with Operation Relativity". Chinese Journal of Mechanical Engineering, vol.42, no 8, pp: 50-55, 2006.

[12] Zhiqiang Xie , Gangjie Ye ,Dali Zhang . "New Nonstandard Job Shop Scheduling Algorithm." Chinese Journal of Mechanical Engineering, vol.21, no 4, pp :97-100, 2008.

[13] Zhiqiang Xie a, Shuzhen Hao , Guangjie Ye ,et al. "A new algorithm for complex product flexible scheduling with constraint between jobs." Computers & Industrial Engineering, vol 57, no 3, pp:766-772, 2009.

[14] Zhiqiang Xie , Jing Yang , Guang Yang . "Dynamic Job-Shop Scheduling Algorithm with Dynamic Set of Operation Having Priority". Chinese Journal of Computer, vol.31, no 3, pp: 502-508, 2008.

[15] Zhiqiang. Xie, Qiushan. Liu, Peili.Qiao, "Dynamic Job Shop Scheduling Algorithm Based on ACPM and BFSM," Journal of Computer Research and Development, vol. 40, no 17, pp:977-983,2003.

[16] Zhiqiang Xie , Shuzhen Hao , Jing Cong . "Optimization controlling algorithm of shortening idle time for the dynamic assembly scheduling." Electric Machines and Control, vol.12, no 1, pp : 69-73, 2008.