Send Orders of Reprints at reprints@benthamscience.net

The Open Electrical & Electronic Engineering Journal, **2013**, *7*, (Supple 1: M6) 55-61 55

Open Access

A Sequence List Algorithm For The Job Shop Scheduling Problem

Shuli Zhang*

School of Software, Harbin University of Science and Technology, China

Abstract: For the discrete manufacturing enterprises, the job shop scheduling problem is an important class of actual combinatorial optimization problem with resources and sequence constraints. According to the needs of the job shop scheduling problem, a sequence list algorithm for the job shop scheduling problem was designed in this paper. In order to make all jobs being finished as soon as possible, the goal of the sequence list algorithm is minimizing the maximal the finish time of all operations. In the sequence list algorithm, two types of sequence lists were built. They are the job sequence lists and the machine sequence lists. A job sequence list was used to store all operations of a job on the basis of its process constraints. A machine sequence list which is null initially was used to store all operations on a machine in accordance with the actual processing order. The important tasks of the sequence list algorithm are inserting all operations of the job sequence lists into the machine sequence lists and adjusting the processing order of the operations in the machine sequence lists. The sequence list algorithm could always achieve a good job shop schedule which ensures the select performance indicators. The feasibility and efficiency of the algorithm was verified through examples.

Keywords: Job shop scheduling, cost function, sequence list.

1. INTRODUCTION

In the discrete manufacturing enterprises, because of variety and small batch of the products, the production organization is very complicated, and the production plan and scheduling is especially difficult. The production plan and scheduling often depends on subjective experience. It causes that drawing up the production plan needs long time, and the production plan lacks timeliness and contingency. Thus, the production cycle of the products is long, the machine capacity factor is low, the amount of funds used is large, and the delivery of the products is delayed. The economic benefits and the social benefits of enterprise are severely affected.

During the production plan and scheduling, the job shop scheduling problem has many practical applications. The job shop scheduling problem is an important class of the actual combinatorial optimization problem with resources and sequence constraints.

The job shop scheduling problem should use limited resources to satisfy various process constraints, determine the actual processing sequence and the actual start time of jobs to ensure the select performance indicators [1-3]. The job shop scheduling problem has been proven to be NP-hard.

Many algorithms have been reported in literature to find solutions for the job shop scheduling problem, such as, tabu search algorithm [4], simulated annealing algorithm [5], neural networks [6], genetic algorithm [7], ant colony algorithm [8], pareto algorithm [9], and RFID technology [10]. But these algorithms can not adapt to the characteristics of the job shop problem, so it is extremely difficult for these job shop scheduling algorithms to generate the good schedules.

According to the unique characteristics of the job shop scheduling problem, a sequence list algorithm for the job shop scheduling problem was presented in this paper. In the sequence list algorithm, two classes of sequence lists were built. The job sequence lists were built on the basis of process constraints. For every job, a job sequence list was used to store all operations of the job. The machine sequence lists were built on the basis of processing order. For every machine, a machine sequence list was used to store all operations on the machine. All machine sequence lists are null initially. First, through inserting all operations of the job sequence lists into the machine sequence lists, a job shop schedule can be achieved. Then, through shortening the waiting time of the operations and improving the machine utilization, the job shop schedule can be adjusted. Using the sequence list algorithm to solve the job shop scheduling problems could always achieve the good job shop schedules which could ensure the select performance indicators.

2. DESCRIPTION OF THE JOB SHOP SCHEDULING PROBLEM

The job shop scheduling problem is described as:

There are m machines and n jobs. n jobs need to be processed on m machines. These machines are M_1, M_2,……,M_m, and for $1 \le k \le m$, each machine M_k has its processing ability. These jobs are J_1, J_2 ,……, J_n, and for $1 \le i \le n$, each job J_i has its processing requirements and planned finish time.

On the basis of processing requirements, each job must be processed according to the certain procedures. There are not operations which can be processed at the same time or in exchanging order. Thus, if job J_i has Q_i operations J_{i1},

*Address correspondence to this author at the School of Software, Harbin University of Science and Technology, China;
Tel: +86-0451-86397006; E-mail: zhangshuli0523@163.com

$J_{i2},\ldots,J_{iQi}$, there is a linear order among these operations, recorded as $J_{i1} \prec J_{i2} \prec \ldots\ldots \prec J_{iQ_i}$.

If $J_{ij} \prec J_{i,j+1}$, then calling that J_{ij} is the previous operation of $J_{i,j+1}$, and $J_{i,j+1}$ is the successive operation of J_{ij}. Then, in the job shop scheduling problem, the number of previous operations of any operation is at most 1, and the number of successive operations of any operation is at most 1 too.

Supposed that the planned finish time of job J_i is t_i; for operation J_{ij}, the processing time is T_{ij}, the processing machine is P_{ij}, the actual start time is S_{ij}, and the actual finish time is F_{ij}, then the constraints of the job shop scheduling problem are:

(1) The number of the machines, the number of the jobs, the processing time and the processing machine of each operation are all known.

(2) A job moves sequentially during processing:

(3) At the same time, a job is only processed on a machine:

For operation J_{ij} and operation J_{il} of job J_i ($1 \le i \le n$, $1 \le j, l \le Q_i$, $J \neq l$), $S_{ij} \le S_{il} \le F_{ij}$, $S_{ij} \le F_{il} \le F_{ij}$, $S_{ij} \le S_{ij} \le F_{il}$ and $S_{il} \le F_{ij} \le F_{il}$ are all false.

(4) At the same time, a machine is only used for processing a job:

For operation J_{ij} and operation J_{gh} on machine M_k ($1 \le F$ij $\le n$, $1 \le j \le Q_i$, $1 \le h \le Q_g$, $i \neq g$, $1 \le k \le m$), $S_{ij} \le S_{gh} \le F_{ij}$, $S_{ij} \le S_{gh} \le F_{ij}$, $S_{gh} \le S_{ij} \le F_{gh}$ and $S_{gh} \le F$ij $\le F_{gh}$ are all false.

(5) An operation is only processed on one machine of many machines:

For any operation J_{ij}, $P_{ij} = M_k$ ($1 \le i \le n$, $1 \le j \le Q_i$, $1 \le k \le n$), then $K \neq l$ ($1 \le l \le n$).

(6) A machine only processes one operation of a job:

For operation J_{ij} on machine M_k ($1 \le i \le n$, $1 \le j \le Q_i$, $1 \le k \le m$), $M_k \neq P_{il}$ ($1 \le l \le Q_i$, $l \neq j$).

(7) Not allowed to interrupt during an operation is being processed:

For any operation J_{ij} ($1 \le i \le n$, $1 \le j \le Q_i$), $F_{ij} = S_{ij} + T_{ij}$.

(8) The operations can be permitted to wait for being processed, the machines can be permitted to wait for a job:

For operation J_{ij} of job J_i on machine M_k ($1 \le i \le n, 1 \le j \le Q_i$, $1 \le k \le m$), if operation J_{gh} ($1 \le g \le n$, $1 \le h \le Q_g$) is processed before operation J_{ij} on machine M_k, $S_{ij} = \max\{F_{i,j-1}, F_{gh}$.

(9) The actual start time of the first operation of any job is bigger than 0 or equal to 0:

For operation J_{i1} of job J_i ($1 \le i \le n$), $S_{il} \ge 0$.

(10) Any operation can only start to be processed after its previous operation has been processed:

For operation $J_{i,j-1}$ and operation J_{ij} of job J_i ($1 \le i \le n$, $1 \le j, j\text{-}1 \le Q_i$), $S_{ij} \le F_{i,j-1}$;

(11) On the same machine, another operation can start to be processed after an operation has been processed:

For operation J_{ij} and operation J_{gh} on machine M_k ($1 \le$ i, $1 \le j \le Qi$, $1 \le h \le Qg$, $i \neq g$, $1 \le k \le m$), $S_{ij} \le F_{gh}$ or $S_{gh} \le F_{ij}$.

The question is determining the actual start time, the actual finish time, and the processing order of all operations on all machines under the processing requirements and the planned finish time of all jobs, in order to optimize the select processing performance indicators. Job shop scheduling problem should use limited resources to meet various constraints of the job, determine the processing sequence and the actual start time of all operations on corresponding machines to ensure the selected performance indicators.

3. THE SEQUENCE LIST ALGORITHM FOR THE JOB SHOP SCHEDULING PROBLEM

3.1. Cost Function

Considering various performance requirements of the job shop scheduling problem, the sequence list algorithm for the job shop scheduling problem requires all jobs could be finished as soon as possible.

Thus, the goal of the sequence list algorithm for the job shop scheduling problem is minimizing the maximal finish time of all operations. The cost function could be defined as:

$$\min \max_{\substack{1 \le i \le n \\ 1 \le j \le Q_i}} \{F_{ij}\}$$

3.2. Thinking in the Sequence List Algorithm

From the description of the job shop scheduling problem, there are two classes of queues. First, according to the certain procedures of a job, the operations of each job can make up a queue. In this queue, the actual start time of any operation is not smaller than the actual finish time of its previous operation. Secondly, according to the actual processing sequence, the jobs on each machine can make up a queue. In this queue, the actual processing time of any operation is not overlapping each other. These queues can be implemented by using the sequence lists.

First, the job sequence lists are built on the basis of process constraints. For every job, according to the linear order among the operations, a job sequence list is built to store all operations of the job. Each node of the job sequence list should contain the given information of the corresponding operation, such as the processing time, the processing machine, the pointer to the successive operation. Because the procedures of all jobs are known, all job sequence lists could be built immediately, and they should not be changed forever.

Secondly, the machine sequence lists are built on the basis of the processing order. For every machine, according to the actual processing sequence, a machine sequence list is built to store all operations on the machine. Each node of a machine sequence list should contain the determined information of the corresponding operation, such as the actual start time, the actual finish time, the pointer to the next operation. Because the actual processing sequence on any machine is not determined, all machine sequence lists are null initially.

Then the job shop scheduling problem is changed to how to insert all operations of n job sequence lists into m machine sequence lists.

In the sequence list algorithm, the earliest start time of operation J_{ij} is E_{ij}, the latest start time of operation J_{ij} is L_{ij}:

$E_{i1} = 0$;

$E_{ij} = \sum_{k=1}^{j-1} T_{ik}$;

$L_{iQ_i} = t_i - T_{iQ_i}$;

$L_{ij} = L_{i,j+1} - T_{ij}$.

Thus:

$S_{ij} = \max\{E_{ij}, F_{i,j-1}, F_{gh}\}$;

$F_{ij} = S_{ij} + T_{ij}$;

In order to achieve the good job shop schedules which could ensure the cost function, the job shop schedule needs to be adjusted through shortening the waiting time of the operations. The adjustment method is as follows:

Set a pointer p points to the machine sequence list of M_k ($k = 1,2,\dots,m$) in proper order.

For pointer p:

First, set a pointer p' points to the first operation J_{ij} in p.

Secondly, determine whether operation J_{ij} can be early processed? If operation J_{ij} can be early processed, modify the earliest start time, the actual start time, the actual finish time of operation J_{ij} and its successive operations of job J_i. If operation J_{ij} can not be early processed, determine whether operation J_{pq} which is after operation J_{ij} in p can be early processed? p' points to the next operation J_{pq} in P in proper order and repeat above processes.

Finally, if p' is null, it indicates that all operations in p have finished adjustment.

3.3. Description of the Sequence List Algorithm

The program of the sequence list algorithm for the job shop scheduling problem is:

Step1. FOR $i = 1$ TO n DO

BEGIN

(1)For job i, build a job sequence list JL_i;

(2)FOR $j = 1$ TO Q_i DO

Compute the earliest start time E_{ij} of operation J_{ij}:

$E_{i1} = 0$;

$E_{ij} = \sum_{k=1}^{j-1} T_{ik}$;

(3) FOR $j = Q_i$ TO 1 DO

Compute the latest start time L_{ij} of operation J_{ij}:

$L_{iQ_i} = t_i - T_{iQ_i}$;

$L_{ij} = L_{i,j+1} - T_{ij}$.

END

Step 2. Sort all job sequence lists by ascending t_i to get SJL_i.

Step 3. FOR $k = 1$ TO m DO

For machine k, build a machine sequence list ML_k which is null.

Step4. FOR $i = 1$ TO n DO

BEGIN

(1) Set a pointer $p1$ which points to the first operation J_{ij} in SJL_i;

(2) If $p1$ is null, it indicates all operations of job J_i have been inserted into machine sequence lists; else:

If $p1$ points to an operation J_{ij} which needs to be processed on machine $P_{ij} = M_k$, search the first operation J_{g1h1} which satisfies $L_{g1h1} > L_{ij}$. There are two cases:

① If there is not any operation J_{g1h1}, insert operation J_{ij} into the end of ML_k, compute the actual start time S_{ij} and the actual finish time F_{ij} of operation J_{ij}:

Supposed that operation J_{g2h2} is before operation J_{ij} on machine P_{ij}, then:

$S_{ij} = \max\{E_{ij}, F_{i,j-1}, F_{g_2h_2}\}$;

$F_{ij} = S_{ij} + T_{ij}$;

If $L_{ij} < S_{ij}$, modify the latest start time of operation J_{ij} and its successive operations of job J_i:

$L_{ij} = S_{ij}$;

$L_{i,j+1} = L_{ij} + T_{ij}$.

② If there is operation J_{g1h1}, insert operation J_{ij} into the position in front of J_{g1h1} in ML_k, compute the actual start time S_{ij} and the actual finish time F_{ij} of operation J_{ij}, compute the actual start time and the actual finish time of operation J_{g1h1} and its successive operations in ML_k, modify the latest start time of operation J_{ij} and its successive operations in ML_k.

(3) $p1$ points to the next operation J_{ij} in SJL_i, go to (2).

END

Step5. FOR $k = 1$ TO m DO

BEGIN

For machine k, set a pointer $p2$ which points to the machine sequence list ML_k.

(1) Set a pointer $p3$ which points to the first operation J_{ij} in $p2$.

(2) $f = 0$.

(3) If $p3$ is null, it indicates that $p3$ points to the end of $p2$, all operations in $p2$ have finished adjustment; else:

(4) $d = S_{ij} - f$.

Table 1. The Procedures of Each Job of Exp.1

Job *i*	P_{i1}	P_{i2}	P_{i3}
J_1	M_1	M_4	M_3
J_2	M_4	M_2	M_3
J_3	M_1	M_2	
J_4	M_4	M_3	

Table 2. The Processing Time and the Planed Finish Time of Exp.1

Job *i*	T_{i1}	T_{i2}	T_{i3}	t_i
J_1	3	3	3	12
J_2	1	2	3	9
J_3	2	4		9
J_4	2	2		6

Table 3. The Earliest Start Time of the Operations of Exp.1

Job *i*	E_{i1}	E_{i2}	E_{i3}
J_1	0	3	6
J_2	0	1	3
J_3	0	2	
J_4	0	2	

(5) If $d \leq 0, f = F_{ij}$, $p3$ points to the next operation in $p2$, go to (3); else:

If $d > 0$, determine whether operation J_{ij} can be early processed?

① If operation J_{ij} can be early processed, $S_{ij} = \max\{E_{ij}, f\}$, modify the earliest start time and the actual start time of operation J_{ij} and its successive operations of job J_i, $p3$ points to the next operation in $p2$, go to (3).

② If operation J_{ij} can not be early processed, determine whether operations after operation J_{ij} can be early processed?

If there is not an operation, $f = F_{ij}$, $p3$ points to the next operation in $p2$, go to (3);

If there is an operation J_{gh}, set a pointer $p4$ which points to J_{gh}, move $p4$ before $p3$, modify S_{ij} of operation J_{ij}, modify the earliest start time and the actual start time of its successive operations of job J_i, modify the earliest start time and the actual start time of successive operations of J_{gh} of job J_g, $p3 = p4$, go to (3).

END.

3.4. Analysis of Time Complexity

From the constraints of the job shop scheduling problem, $\max\limits_{1 \leq i \leq n}\{Q_i\} \leq m$.

In the sequence lists algorithm for the job shop scheduling problem, Step1 takes $O(n \text{ x } m)$ $(m + m + m)$), that is $O(n \text{ x } m)$; Step2 takes $O(n\log_2 n)$; Step3 takes $O(m)$, Step4 takes $O(n \text{ x } (1+m+1))$, that is $O(n\text{x}m)$; Step5 takes $O(m \text{ x } (1+1+1+1+n))$, that is $O(n \text{ x } m)$.

So the sequence lists algorithm for the job shop scheduling problem total takes $O(n \text{ x } m)$, the time complexity of the sequence lists algorithm is $O(n \text{ x } m)$.

4. EXAMPLES

The feasibility and efficiency of the algorithm is verified through two examples as follow:

Exp.1: There are 4 jobs and 3 machines in this job shop problem. The job set is $J = \{J_1, J_2, J_3, J_4\}$, and the machine set is $M = \{M_1, M_2, M_3\}$.

The procedures of each job are given in Table **1**.

The processing time of each operation and the planed finish time of each job are given in Table **2**.

According to the sequence lists algorithm for the job shop scheduling problem, first, gets the earliest start time of the operations; the result is shown in Table **3**.

Secondly, gets the planned latest start time of the operations; the result is shown in Table **4**.

Finally, implement scheduling, the scheduling Gantt result is shown in Fig. (**1**).

From Fig. (**1**), a good schedule was generated by the sequence list algorithm for the job shop scheduling problem. The schedule ensures that each job can be finished before its planned finish time.

In this schedule, the machine capacity factor is more than 80%.

Table 4. The Planned Latest Start Time of the Operations of Exp.1

Job *i*	L_{i1}	L_{i2}	L_{i3}
J_1	3	6	9
J_2	3	4	6
J_3	3	5	
J_4	2	4	

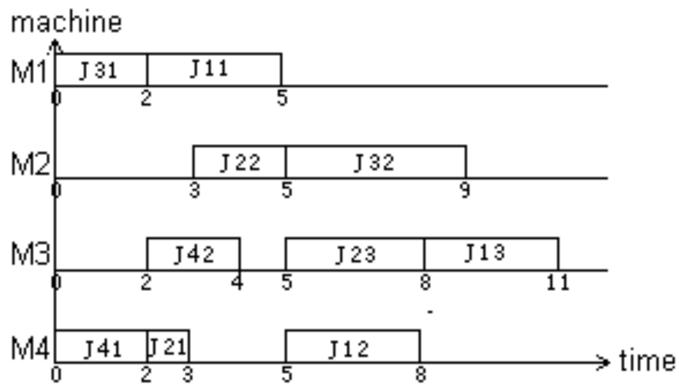

Fig. (1). The scheduling Gantt of Exp.1.

Table 5. The Procedures of Each Job of Exp.2

Job *i*	P_{i1}	P_{i2}	P_{i3}	P_{i4}
J_1	M_1	M_3	M_2	M_4
J_2	M_3	M_1	M_2	M_4
J_3	M_1	M_4	M_3	M_2
J_4	M_3	M_2	M_1	M_4
J_5	M_4	M_2	M_3	M_1
J_6	M_2	M_4	M_1	M_3
J_7	M_2	M_3	M_4	M_1
J_8	M_4	M_1	M_2	M_3

Table 6. The Processing Time and the Planed Finish Time of Exp.2

Job *i*	T_{i1}	T_{i2}	T_{i3}	T_{i4}	t_i
J_1	4	4	7	3	38
J_2	3	4	4	2	36
J_3	3	4	6	3	38
J_4	2	6	4	3	36
J_5	4	5	5	3	35
J_6	4	6	5	4	37
J_7	2	4	5	5	39
J_8	3	/3	2	5	40

Exp.2: There are 8 jobs and 4 machines in this job shop problem. The job set is $J = \{J_1, J_2, J_3, J_4, , J_5, , J_6, , J_7, , J_8\}$, and the machine set is $M = \{M_1, M_2, M_3, M_4\}$.

The procedures of each job in this example are given in Table **5**.

The processing time of each operation and the planed finish time of each job in this example are given in Table **6**.

According to the sequence lists algorithm for the job shop scheduling problem, first, gets the earliest start time of the operations; the result is shown in Table **7**.

Table 7. The Earliest Start Time of the Operations of Exp.2

Job *i*	E_{i1}	E_{i2}	E_{i3}	E_{i4}
J_1	0	4	8	15
J_2	0	3	7	11
J_3	0	3	7	13
J_4	0	2	8	12
J_5	0	4	9	14
J_6	0	4	10	15
J_7	0	2	6	11
J_8	0	3	6	8

Table 8. The planned Latest Start Time of the Operations of Exp.2

Job *i*	L_{i1}	L_{i2}	L_{i3}	L_{i4}
J_1	20	24	28	35
J_2	23	26	30	34
J_3	22	25	29	35
J_4	21	23	29	35
J_5	18	22	27	32
J_6	18	22	28	33
J_7	23	25	29	34
J_8	27	30	33	35

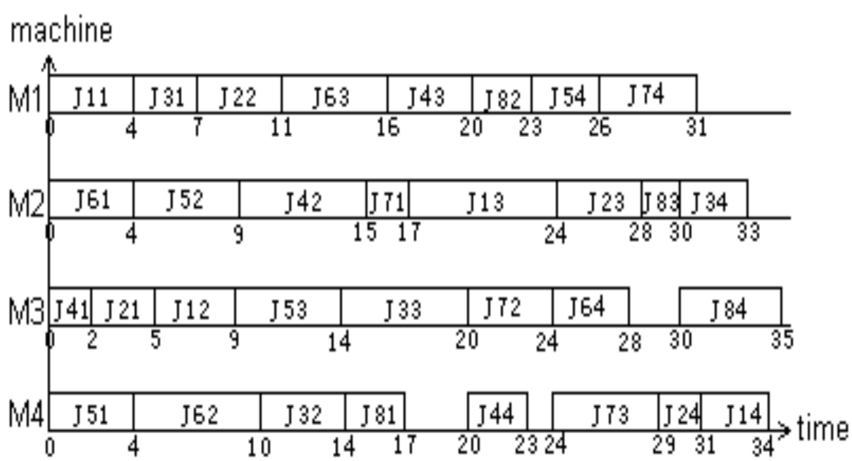

Fig. (2). The scheduling Gantt of Exp.2.

Secondly, gets the planned latest start time of the operations; the result is shown in Table **8**.

Finally, implement scheduling, the scheduling Gantt result is shown in Fig. (**2**).

From this Gantt, The best schedule was generated by the sequence list algorithm for the job shop scheduling problem. In this schedule, the processing cycle is the shortest, and the machine capacity factor is more than 95%. Actually, using the sequence list algorithm to solve large-scale the job shop scheduling problem can achieve more excellent result.

5. CONCLUSIONS

The sequence list algorithm for the job shop scheduling problem algorithms could be able to adapt to the characteristics of the job shop problem, so it can generate the good schedules always.

The sequence list algorithm for the job shop scheduling problem has the characteristics as follows:

(1) Ensure the linear order of every job:

Because of $S_{ij} = \max\{E_{ij}, F_{i,j-1}, F_{gh}\}$, $S_{ij} \geq F_{i,j-1}$, so operation J_{ij} is processed after operation J_{ij-1}, it ensures the linear order among operations of every job.

(2) Ensure the processing cycle shortest or shorter:

Because that the goal of the sequence list algorithm is $\min \max_{\substack{1\leq i\leq n \\ 1\leq j\leq Q_i}}\{F_{ij}\}$, it ensures all jobs can be processing finished as soon as possible, so the processing cycle is shortest or shorter.

(3) Ensure the machine capacity factor highest or higher.

Because that the processing task is certain and the processing cycle is shortest or shorter, it ensures the machine capacity factor highest or higher.

Generally, the sequence list algorithm is more suitable for solving large-scale the job shop scheduling problem.

The ability of the algorithm is verified through two examples; it is quite feasible and effective.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflicts of interest.

ACKNOWLEDGEMENT

Declared none.

REFERENCES

[1] S. Liu, S. Zhang, B. Wang, and C. Teng, "Improved virus evolutionary genetic algor ithm for Job-shop scheduling problem ", *Electr. Mach. Control*, vol.12, no. 2, pp. 234-238, 2008.

[2] S. Liu, S. Zhang, B. Wang, and C. Teng, "Patheno Genetic algorithm for solving job-shop scheduling problem", *Comput. Eng.,* vol. 35 no.5, pp. 188-190, 2009.

[3] S. Liu, S. Zhang, B. Wang, and C. Teng, "A Job-shop dynamic scheduling algorithm developed by considering uncertain factors", *J. Harbin Eng., Uni*., vol. 32, no.4, pp. 471-475, 2011.

[4] S.M. Mohammad, and F. Parviz, "Flexible job shop scheduling with tabu search algorithms", *Int. J. Adv. Manuf. Technol*., vol. 32, no. 5-6, pp. 563-570, 2007.

[5] E. Atabak, S. Maghsud, T. Seyda, and E. Afshin, "A simulated annealing algorithm for the job shop cell scheduling problem with intercellular moves and reentrant parts", *Comput. Ind. Eng*., vol. 61, no. 1, pp. 171-178, 2011.

[6] N. Yahyaoui, and F. Fnaiech, "A suitable initialization procedure for speeding a neural network job-shop scheduling", *IEEE Trans. Ind. Electron*., vol. 58, no. 3, pp. 1052-1060, 2011.

[7] L. Anna, "Advanced scheduling with genetic algorithms in supply networks", *J. Manuf. Technol. Manag*., vol. 22, no.6, pp.748-769, 2011.

[8] A. Udomsakdigool, and V. Kachitvichyanukul, "Multiple colony ant algorithm for job-shop scheduling problem", *Int. J. Prod. Res*., vol. 46, no.15, pp. 4155-4175, 2008.

[9] R. Tavakkoli, M. Azarkish, and A. Sadeghnejad, "A new hybrid multi-objective pareto archive PSO algorithm for a bi-objective the job shop scheduling problem", *Expert Syst. Appl*., vol. 38, no. 9, pp. 10812-10821, 2011.

[10] Ferrer, K. Susan Heath, and N. Dew, " An RFID application in large job shop remanufacturing operations", *Int. J. Prod. Eco*., vol. 133, no. 2, pp. 612-621, 2011.

Received: January 29, 2012 Revised: April 20, 2012 Accepted: October 23, 2012