▩ 연구논문

SIMULATED ANNEALING APPROACH FOR MINIMIZING THE MEAN SQUARED DEVIATION FROM A DUE DATE[1)]

-공통 납기로부터 편차의 평균 제곱을 최소화하는 모의 뜨임 접근 방법-

Kim, Chae Bogk[2)]
김 채 복
Lee, Dong Hoon[3)]
이 동 훈

요 지

본 연구는 공통의 납기로부터 완성 시간의 편차의 평균제곱을 최소화하는 문제를 비제약적인 경우와 제약적인 경우에 다루고 있다. 모의 뜨임 기법을 이용하여 Eilon과 Chowdhury의 [4] 논문에 있는 예제를 테스트하였다. 제안된 자기 발견적 기법은 대부분의 경우에 좋은 해를 제공하였으며, 작업의 수가 200인 경우에도 해를 1초안에 찾았다. 비제약적인 경우와 제약적인 경우의 계산 결과가 제시되었으며, 다른 자기 발견적 기법에 의한 계산 결과와 비교하였다.

1. INTRODUCTION

After the Just-In-Time (JIT) production system was introduced, the problem of minimizing the function of both earliness and tardiness in a job shop has received a great attentions because one of the concepts of JIT production system is that the parts (components) are supplied at the needed time point. Baker and Scudder [2], Kanet [7], and Sidney [13] decribed why it is desirable to reduce the earliness and tardiness simultanenously.

Baker and Scudder [2] surveyed various kinds of objective functions related to both earliness and tardiness in a single machine with a common due date. The problem of minimizing mean squared deviation (MSD) from a common due date, instead of mean absolute deviation (MAD), is attacked in this paper because large deviations from the common due date usually incur more cost than small deviations in most production systems. The MSD problem is classified into unconstrained and constrained problem in terms of the common due date.

For the unconstrained MSD problem, an enumeration algorithm with branch and bound technique which provides an optimal solution is developed by Bagchi et al.

---

1 This work was supported by KOSEF(#971-0907-047-2).
2 Department of Technology education, Korea National University of Education
3 Department of Computer Science, Korea University

[1]. In order to reduce the computational burden, they presented several properties which fathom branches. De et al. [3] presented a dynamic programming algorithm whose complexity is pseudo-polynomial. They reported that it takes about 75 seconds in VAX 8600 machine when the number of jobs are 100. Even though the algorithm obtains an optimal solution relatively quickly, it is still meaningful to develop a heuristic algorithm when there are large number of jobs because Kubiak [9] showed the mean squared deviation problem is NP-complete. For the NP-complete problems, most researchers tend to develop heuristic procedures in nature.

There are several heuristics for MSD problem in the literature (Eilon and Chowdhury [4], Kanet [7], and Vani and Raghavachari [15], Gupta et al. [5], Kim and Foote [8]). The following notation and symbols will be used throughout this paper.

$n \equiv$ Number of jobs to be scheduled
$p_j \equiv j^{th}$ smallest processing time ($p_1 \leq p_2 \leq \cdots \leq p_n$)
$J_j \equiv$ The job with j^{th} smallest processing time
$c_j \equiv$ Completion time of job j in a given sequence
$\bar{c} \equiv$ Mean completion time in a given sequence
$d \equiv$ Common due date
$\Pi \equiv$ All possible sequences
$Z(S) \equiv$ Objective function value of sequence S

The MSD problem is formulated mathematically as follows by using above notation:

$$\text{Minimize} \quad Z(S) = \frac{1}{n} \sum_{i=1}^{n} (c_i - d)^2, \ \forall S \in \Pi \tag{1}$$

After differentiating equation (1) with respect to d and equating to zero, solve for d. Then, the objective function becomes

$$\text{Minimize} \quad Z(S) = \frac{1}{n} \sum_{i=1}^{n} (c_i - c)^2, \ \forall S \in \Pi \tag{2}$$

where $d \geq d^* = \bar{c}$

Note that this is only true when the common due date d is greater than or equal to d^*, the minimum due date for optimal solution. If above condition is not satisfied, c is not the optimal due date that minimizes the variance of completion times. Therefore, The MSD problem can be divided into two problems: Unconstrained MSD problem and constrained MSD problem. The following assumptions will be used to solve the MSD problem.

(1) All processing times are deterministic and known.
(2) The set up time of each job is not affected by the sequence and included in the processing time.
(3) All jobs are available at the same time (at time zero).
(4) Preemption is not permitted.

2. LITERATURE REVIEW

Bagchi et al. [1] showed that the unconstrained MSD problem, equation (2), is equivalent to the completion time variance (CTV) problem. Therefore, we can use the results of the papers (Schrage [12], Eilon and Chowdhury [4], Raghavachari [11], Hall and Kubiak [6], and Kim and Foote [8]) in order to attack the unconstrained MSD problem.

Merten and Muller [10] introduced the CTV problem more than two decades ago for the file organization. They showed that the CTV problem is equivalent to the waiting time variance problem. Schrage [12] obtained the optimal sequences for $n \leq 5$.

Theorem 1 (Schrage 1975)
The schedule such that the job with the longest processing time is processed first minimizes completion time variance.

Eilon and Chowdhury [4] considered int erchanging two jobs in a given sequence and obtained the difference of objective function values incurred from interchanging two jobs. They presented five heuristics for the problem with relatively large number of jobs and computational experience was shown.

Theorem 2 (Eilon and Chowdhury 1977)
The schedule such that the job with the second longest processing time is processed last minimizes completion time variance.

Theorem 3 (Eilon and Chowdhury 1977)
The schedule that minimizes the completion time variance is V-shaped.

A schedule is V-shaped if jobs for which $c_j \leq d$ are sequenced in nonincreasing order of processing time and jobs for which $c_j > d$ are sequenced in nondecreasing order of processing time. Kanet [7] studied the problem of minimizing the flow time variation on single machine with n jobs. He showed that the CTV problem is equivalent to minimizing the sum of squared deviations of job completion times. He presented a heuristic algorithm and compared it to the heuristic of Eilon and Chowdhury [4].

Vani and Raghavachari [15] generalized an interchanging property and obtained optimal sequences for n = 6, 7. They presented a heuristic using the interchanging property and compared their results with the results of Eilon and Chowdhury [4] and Kanet [7].

Theorem 4 (Vani and Raghavachari 1987)
The schedule such that the job with the second longest processing time is processed last and the job with third longest processing time is processed second minimizes completion time variance when $n \leq 18$.

Theorem 5 (Hall and Kubiak 1991)
The schedule such that the job with the second longest processing time is processed last and the job with third longest processing time is processed second minimizes completion time variance.

For the large and difficult combinatorial optimization problems, the meta heuristic method is used to obtain good solution. Simulated annealing, one of the mera heuristics, comes from the analogy between combinatorial optimization and thermal equilibrium of the solid. It goes through a number of iterations when to be applied to the combinatorial optimization problem. Recently, simulated annealing method received a great attention in scheduling area (see Van Laarhoven et al. [14]).

The success of simulated annealing method are dependent on how to find an effective procedure moving from one solution to better solution. Generally, the simulated annealing method consists of initial configuration, initial temperature, neighborhood structure for pairwise interchanging, cooling temperature scheme, and the computation of objective function.

With a number of iterations, simulated annealing process looks for a solution like local search technique. However, it improves the weak point of local search by the possibility of accepting a change for a worse solution with probability. Many researchers proposed heuristics using simulated annealing for various scheduling problems (see Van Laarhoven et al. [14]). Most of them can be coded very easily and provide fairly good solution. But, the amount of computation time needed to obtain such a solution tends to be relatively long.

In this paper, heuristic based on the simulated annealing process is presented. It is applicable to both unconstrained and constrained MSD problem. The proposed heuristic recorded short computational time when to solve the large job set problem because of the special property (V-shape property). Since the V-shape property confines the set of neighborhoods, the simulated annealing process works very fast. Consider two jobs which may be exchanged for better solution. To pick such two jobs, for example, job k is selected first. Then, the jobs which can be exchanged with job k are only job $k+1$ or job $k-1$. Any other jobs will violate the V-shape property. That is the reason why the simulated annealing process is adopted for minimizing MSD problem in this paper.

3. SOLUTION APPROACH

3.1. Simulated annealing process for MSD problem

Generally, a number of iterations are needed to apply simulated annealing process to scheduling problem. At each iteration k, there are both a current schedule and a best schedule obtained so far. Then, a decision making is needed to select a new schedule from the neighborhoods of current schedule. After then, the objective function value of new schedule is compared with the best solution so far. If the new solution is better, it will be the current solution of next iteration and the best schedule is updated. Otherwise, the newly generated schedule is accepted with probability. This possibility of accepting the worse solution is different from any other local search techniques. The more the number of iterations in simulated annealing process are, the lower the probability of accepting worse solution. It is called cooling scheme or temperature control.

In order to apply the simulated annealing method to the MSD problem, the definition of initial schedule, neighborhood structure, temperature control and cooling scheme of temperature are required. Especially, the structure of neighborhood is a very important aspect of the simulated annealing process because it greatly affects the computational time.

The selection of initial solution is also an important factor like the most iterative search methods because it affects the quality of final solution.

3.2. Unconstrained MSD problem

When $d \geq d^*$, the MSD problem is called as unconstrained MSD problem and $\bar{c}$ is the common due date which provides the optimal solution. Then, Theorem 1 to Theorem 5 can be used. There are 2^{n-4} sequences which satisfy Theorem 1 through 5 if all processing times are distinct.

Since the procedure (De et al. [3]) for the optimal solution in MSD problem requires much computational time when n is a large number, a heuristic approach is desirable when n is large. Ventura and Weng [16] used a Lagrangian relaxation of a quadratic integer programming formulation which sets an upper bound on deviation from the optimal solution. The algorithm consists of the modified method of De et al. and the pairwise interchanges. But, the CPU times are still much. There are several heuristic algorithms by Eilon and Chowdhury [4], Kanet [7], Vani and Raghavachari [15], and Kim and Foote [8].

In order to apply the simulated annealing process to the MSD problem, consider the initial solution since the simulated annealing process performs better with a good initial solution. We adopt the initial sequence of Vani and Ragavachari's algorithm as an initial configuration because it is simple to generate and it has a V-shaped property. It is determined as follows.

$$S = (J_n, J_{n-2}, J_{n-4}, \cdots, J_{n-5}, J_{n-3}, J_{n-1})$$

The neighborhood structure is consisted of those sequences that result from the current sequence of each iteration by the interchange of the position of two jobs. For the interchange operation, any two consecutive jobs (not the position in a given sequence but the order of processing times of two jobs) are selected because of the V-shaped property. This special neighborhood structure makes the effect of reducing computational time of proposed heuristic. Of course, the positions of three jobs with the first, the second and the third largest processing time are fixed in unconstrained case. That means the jobs on the first , last and second positions are not affected by the position interchanging operation. Then, evaluate the objective function value. If the interchange of two jobs improves the objective function value (total sum of mean squared deviations), the interchange is accepted and the solution is recorded. Otherwise, compute the difference of objective function values and set it to δ. Draw the random number X from the uniform distribution with the range of [0, 1]. If $X \leq e^{-\frac{\delta}{T}}$, the solution is accepted and try another interchanging operation. If not, try another interchanging of two jobs in the best sequence recorded.

In order to obtain the temperature value of each iteration, we generate 100 sequences with V-shape property and pick the best and worst objective function values. Set them δ_{min} and δ_{max}, respectively. Then, an initial temperature and final temperature are determined as

$$T_0 = \delta_{min} + \frac{1}{10}(\delta_{max} - \delta_{min}) \tag{3}$$

$$T_f = \delta_{min} \tag{4}$$

That means only the tenth from the bottom of the range of temperatures are used in the annealing scheme. When the user does not specify a temperature range, this temperature control approach is generally employed. The cooling scheme of temperature in each iteration is determined by the recurrence relationship such that

$$T_{i+1} = \frac{T_i}{1+\beta T_i} \tag{5}$$

where $\beta = \frac{(T_0 - T_f)}{M T_0 T_f}$ and M denotes the number of interchanging, 50K. The size of neighborhoods, K, is set to $\frac{1}{2}n(n-1)$. Note that there are at most $(n-1)$ neighborhoods because of the V-shape property even though we have $\frac{1}{2}n(n-1)$ number of pairwise interchange.

3.3. Constrained MSD problem

If $d < d^*$ in the objective function (1), then the MSD problem is called as a constrained MSD problem and c is not an optimal common due date which minimizes the objective function any more. Then, only Theorem 3 is effective in this case.

Unlike unconstrained MSD problem, there are not many algorithms in the literature. Bagchi et al. [1] discussed the procedure to find the optimal solution with branch and bound (enumeration) technique. Since the quality of solution by the simulated annealing algorithm is affected by the initial solution, we adopt a SPT (shortest processing time) sequence, $S = (J_1, J_2, J_3, \ldots, J_{n-1}, J_n)$, as an initial solution because of following Theorem. Other aspects of the algorithm except for the method of obtaining initial schedule are all the same as unconstrained case.

Theorem 6. (Bagchi, Sullivan and Chang 1987)
The SPT sequence is optimal if $d \le \frac{p_1 + p_2}{2}$

3.4. Computational results

We programmed proposed heuristic by using simulated annealing technique in FORTRAN 77 and ran on an IBM 6000 RISC system. Seven test problems taken from Eilon and Chowdhury [4] are run and Table 1, Table 2 and Table 3 show the input data and results. Comparing with other heuristic algorithms, it found better solutions. Heuristic algorithm based on simulated annealing presents 6 optimal solutions and 1 near-optimal solution. Even though the heuristic by Vani and Raghavachari [15] also presents 6 optimal solutions and 1 near-optimal solution, it requires more computational time as Baker and Scudder [2] and Ventura and Weng [16] depicted. In this case, the heuristic by Kim and Foote [8] also obtains 6 optimal solution and 1 near-optimal solution. In order to see the computational times of proposed method, 16 test problems are generated. Table 4 shows the computational time for large number of jobs. For each job size, random sampling with the uniform distribution to create processing times is used. According to the Table 4, as the number of jobs becomes large, the computational time of Kim and Foote heuristic increases much comparing with proposed heurictic. The computational time of proposed heuristic takes less than 1 second when n=200

Table 1. Input data and optimal sequences

problem	processing time	optimal sequence	$\bar{c}(d^*)$
1	2,6,9,12,19,21	6,4,3,1,2,5	43.17
2	2,3,6,9,21,65,82	7,5,4,3,2,1,6	121.00
3	2,4,6,7,8,9,10,16	8,6,5,2,1,3,4,7	38.63
4	1,2,8,9,10,12,13,16	8,6,4,3,1,2,5,7	43.63
5	1,2,5,8,9,10,13,16,18,19	10,8,7,5,2,1,3,4,6,9	60.00
6	2,3,6,9,21,23,34,65,82,92	10,8,7,4,3,2,1,5,6,9	209.00
7	5,7,8,9,10,13,21,25,41,100	10,8,7,5,2,1,3,4,6,9	165.60

Table 2. Sequences obtained by proposed heuristics

problem	simulated annealing
1	6,4,3,1,2,5
2	7,5,4,3,1,2,6
3	8,6,5,2,1,3,4,7
4	8,6,4,3,1,2,5,7
5	10,8,7,5,2,1,3,4,6,9
6	10,8,7,4,3,2,1,5,6,9
7	10,8,7,5,2,1,3,4,6,9

Table 3. Comparison of objective function values

problem	optimal	S.A.	K&F	V&R	Kanet	E&C
1	218.47	218.47	218.47	218.47	218.47	218.47
2	918.29	918.41	918.29	918.29	918.29	923.35
3	187.23	187.23	187.23	187.23	187.23	187.42
4	254.23	254.23	254.23	254.48	254.58	254.72
5	486.40	486.40	486.40	486.40	186.44	187.08
6	3584.00	3584.00	3584.00	3584.00	3584.00	3593.00
7	1336.24	1336.24	1336.24	1336.24	1336.24	1336.24

S.A. Simulated annealing heuristic

K&F Kim and Foote (1996) heuristic

V&R Vani and Raghavachari (1987) heuristic

Kanet Kanet (1981) heuristic

E&C Eilon and Chowdhury (1977) heuristic

Table 4. Computational times for unconstrained MSD problem

n	P_j	CPU(seconds)		P_j	CPU(seconds)	
		K&F	S.A.		K&F	S.A.
25	U(1,50)	0.27	0.19	U(1,50)	0.26	0.18
	U(1,50)	0.31	0.20	U(1,50)	0.29	0.19
50	U(1,50)	0.58	0.26	U(1,50)	0.53	0.25
	U(1,50)	0.61	0.30	U(1,50)	0.54	0.25
100	U(1,50)	2.74	0.47	U(1,50)	2.76	0.46
	U(1,50)	2.76	0.41	U(1,50)	2.73	0.44
200	U(1,50)	7.58	0.83	U(1,50)	7.84	0.80
	U(1,50)	7.83	0.82	U(1,50)	7.68	0.77

For constrained MSD problem, seven test problems (same as unconstrained MSD problems except a common due date) are run on IBM 6000 RISC system and common due dates are set to d = 0.2d^*, d = 0.5d^* or d = 0.8d^*, respectively. The optimal sequences and computational results are shown in Table 5 through Table 9.

Table 5. Optimal sequences of the problems

problem	optimal sequences of the problems		
	d = 0.2d^*	d = 0.5d^*	d = 0.8d^*
1	3,1,2,4,5,6	4,3,1,2,5,6	6,3,2,1,4,5
2	5,2,1,3,4,6,7	6,1,2,3,4,5,7	7,4,3,1,2,5,6
3	3,1,2,4,5,6,7,8	6,4,2,1,3,5,7,8	7,6,4,2,1,3,5,8
4	3,1,2,4,5,6,7,8	6,3,2,1,4,5,7,8	7,6,3,2,1,4,5,8
5	5,2,1,3,4,6,7,8,9,10	7,6,3,2,1,4,5,8,9,10	9,7,6,3,2,1,4,5,8,10
6	6,4,3,2,1,5,7,8,9,10	8,6,4,3,1,2,5,7,9,10	10,7,6,4,3,2,1,5,8,9
7	6,5,3,1,2,4,7,8,9,10	9,7,5,3,1,2,4,6,8,10	10,6,4,3,1,2,5,7,8,9

Table 6. Sequences obtained by proposed heuristic

problem	optimal sequences of the problems		
	d = 0.2d^*	d = 0.5d^*	d = 0.8d^*
1	3,1,2,4,5,6	4,3,1,2,5,6	6,3,2,1,4,5
2	5,1,2,3,4,6,7	6,1,2,3,4,5,7	7,4,3,1,2,5,6
3	3,1,2,4,5,6,7,8	6,4,2,1,3,5,7,8	7,6,4,2,1,3,5,8
4	3,1,2,4,5,6,7,8	6,3,2,1,4,5,7,8	7,6,3,1,2,4,5,8
5	5,1,2,3,4,6,7,8,9,10	9,5,2,1,3,4,6,7,8,10	9,7,6,1,2,3,4,5,8,10
6	7,1,2,3,4,5,7,8,9,10	9,5,1,2,3,4,6,7,8,10	9,8,5,1,2,3,4,6,7,10
7	6,1,2,3,4,5,7,8,9,10	9,7,2,1,3,4,5,6,8,10	10,6,4,3,1,2,5,7,8,9

Table 7. Optimal objective function values

problem	optimal objective function values		
	d= 0.2d^*	d= 0.5d^*	d= 0.8d^*
1	947.37	515.86	274.14
2	4839.75	2759.96	1326.93
3	716.24	376.62	231.24
4	963.99	516.58	298.93
5	1755.40	964.90	561.30
6	14724.00	7972.25	4484.26
7	6037.11	3125.54	1930.97

Table 8. Objective function values by proposed heuristic

problem	objetive function values by K&F			objetive function values by S.A.		
	d = 0.2d^*	d = 0.5d^*	d = 0.8d^*	d = 0.2d^*	d=0.5d^*	d=0.8d^*
1	947.37	515.86	274.20	947.37	515.86	274.14
2	4839.75	2759.96	1326.93	4839.95	2759.96	1326.93
3	716.24	376.62	232.01	716.24	376.62	231.24
4	963.99	516.58	300.17	963.99	516.58	299.03
5	1755.60	969.90	562.70	1755.70	991.80	566.10
6	14724.00	7972.25	4484.26	14764.48	8217.15	4508.44
7	6037.11	3125.54	1930.97	6053.17	1337.72	1930.97

Table 9. Computational time for constrained MSD problem

n	p_j	due date	CPU(seconds)		p_j	due date	CPU(seconds)	
			K&F	S.A			K&F	S,A
25	U(1,50)	25	0.27	0.18	U(1,100)	50	0.26	0.22
	U(1,150)	75	0,20	0.22	U(1,200)	100	0.23	0.20
50	U(1,100)	50	1.74	0.38	U(1,200)	100	1.68	0.41
	U(1,300)	150	1.70	0.38	U(1,400)	200	1.64	0.35
100	U(1,200)	100	6.26	0.57	U(1,400)	200	6.06	0.56
	U(1,600)	300	6.28	0.55	U(1,800)	400	6.16	0.55

According to the computational results the proposed heuristic provides 11 optimal solutions and 10 near-optimal solutions. But, when the number of jobs are less than or equal to 8, the proposed heuristic presents 10 optimal solutions and 2 near-optimal solutions. Table 9 shows computational times with large problem on IBM 6000 RISC system. For each job size, the processing times are generated by random sampling with the uniform distribution like unconstrained case. Note that proposed heuristic takes less than 1 second in constrained MSD problem when there are 100 jobs (dynamic programing algorithm needs about 75 seconds even in VAX 8600 system).

3. CONCLUSION

In this paper, we reviewed the literature concerning the function of earliness and tardiness. Emphasis is on unconstrained/constrained MSD problem. To solve the MSD problem, we presented a heuristic algorithm based on the simulated annealing technique. We compared the results obtained by proposed heuristic with other heuristics in the literature (Eilon and Chowdhury [4], Kanet [7], Vani and Raghavachari [15], and Kim and Foote [8]) about unconstrained MSD problem. Proposed heuristic provided better solutions than any other heuristics.

For the constrained MSD problem, the proposed heuristic also performs well. We obtained 11 optimal solutions and 10 near-optimal solutions in 21 problems. When the number of jobs are less than or equal to 8, 10 optimal solutions and two near-optimal solutions are obtained.

Proposed heuristic also require fairly small computational time. It takes less than 1 second when n = 100. Future study will be to consider the earliness and tardiness penalty problem when jobs are dependent on their sequence. In this case, Theorem 1 through Theorem 5 do not hold. Another problem to address is on MSD with different weights on earliness and tardiness.

REFERENCES

[1] Bagchi, Uttarayan, Sullivan, Robert S., and Chang, Yih-Long, "Minimizing mean squared deviation of completion times about a common due date", *Management Science*, Vol. 33, pp. 894-906, 1987

[2] Baker, Kenneth R. and Scudder, Gary D., "Sequencing with earliness and tardiness

penalties: A review", *Operations Research*, Vol. 38, pp. 22-36, 1990

[3] De, Prabuddha, Ghosh, Jay B. and Wells, Charles E., "On the minimization of completion time variance with a bicriteria extension", *Operations Research*, Vol. 40, pp. 1148-1155, 1992

[4] Eilon, Samuel and Chowdhury, I. G., "Minimizing waiting time variance in the single machine problem", *Management Science*, Vol. 23, pp. 567-575, 1977

[5] Gupta, M. G., Gupta, Y. P. and Bector, G. R., "Minimizing the flow time variance in the single machine systems", *Journal of Operational Research Society*, Vol. 41, pp. 767-779, 1990

[6] Hall, Nicholas G. and Kubiak Weislaw, "Proof of a conjecture of Schrage about the completion time variance problem", *Operations Research Letters*, Vol. 10, pp. 467-472, 1991

[7] Kanet, John J., "Minimizing variation of flow time in single machine systems", *Management Science*, Vol. 27, pp. 1453-1459, 1981

[8] Kim, C. and Foote, B. L., "Minimizing the mean squared deviation from a common due date: Unconstrained and constrained cases", *Production Planning & Control*, Vol. 7, pp. 492-502, 1996

[9] Kubiak, W., "Completion time variance minimization on a single machine is difficult", *Operations Research Letters*, Vol. 14, pp. 49-59, 1993

[10] Merten, Alan G. and Muller, Mervin E., "Variance Minimization in single machine sequencing problems", *Management Science*, Vol. 18, pp. 518-528, 1972

[11] Raghavachari, M., "A V-shaped property of optimal schedule of jobs about a common due date", *European Journal of Operational Research*, Vol. 23, pp. 401-402, 1986

[12] Schrage, Linus, "Minimizing the time in system variance for a finite job set", *Management Science*, Vol. 21, pp. 540-543, 1975

[13] Sidney, Jeffrey B., "Optimal single machine scheduling with earliness and tardiness penalties", *Operations Research*, Vol.25, pp. 62-69, 1977

[14] Van Laarhoven, P. J. M., Aarts, E. H. L. and Lenstra, J. K., "Job shop scheduling by simulated annealing", *Operations Research*, Vol. 40, pp. 113-125, 1992

[15] Vani, Vina and Raghavachari, M., "Deterministic and random single machine sequencing with variance minimization", *Operations Research*, Vol. 35, pp. 111-120, 1987

[16] Ventura, Jose A. and Weng, Michael X., "Minimizing single-machine completion time variance", *Management Science*, Vol. 41, pp. 1448-1455, 1995