

A branch and bound for single machine stochastic scheduling to minimize the maximum lateness

Hamidreza Haddad^{a*}, Mohammad Ebrahim Arbabian^b and Keyvan Kamandani Pour^b

^a*School of Industrial Engineering, Iran University of Science & Technology, Iran*

^b*Department of Industrial Engineering, K. N. Toosi University of Technology, Iran*

ARTICLE INFO

Article history:

Received 20 September 2011
Received in revised form
November, 2, 2011
Accepted November, 3 2011
Available online
11 November 2011

Keywords:

Stochastic scheduling
Single machine
Tardy jobs
Deterioration

ABSTRACT

This paper studies the problem of single machine stochastic scheduling with random processing times, deterministic due dates and an independent setup time. The jobs are also deteriorated based on the position, which their processes are done. The objective function is to find a schedule of jobs, which minimizes the expected value of maximum lateness. A branch and bound scheme is presented to solve the problem analytically and a simulated annealing meta-heuristic (SA) is also provided for solving the problem in larger scales. Computational experiments demonstrate that the proposed SA is capable of finding near optimal solutions with very low gap.

© 2012 Growing Science Ltd. All rights reserved

1. Introduction

Generally, it is desired for the manufacturer to provide the orders on time and reduce the penalty costs of lateness in the production system. Therefore, a suitable sequence of processing jobs is required for all machines. Based on this regard, there are many researchers studying the problems of tardiness and lateness in deterministic scheme in which the values of all the parameters of the problem (including processing time, due date, etc.) are known in advance. However, under some conditions, especially in project management, it is not logical to consider all the parameters as deterministic values. Some researchers are interested in studying the problem in stochastic version that is more complex than the deterministic ones.

Gutjahr et al. (1999) studied the single machine stochastic problem to minimize the total weighted tardiness by using a stochastic branch and bound which portioned the solution space and could find the optimal solution based on the certain search of solution space. Gutjahr and Pflug (1996) considered the same problem and modified a stochastic simulated annealing to find near optimum solutions. Norkin et al. (1994) presented a stochastic branch and bound for solving the total tardiness scheduling problem. Soroush (2007 a) studied the problem of minimizing weighted numbers of early and tardy jobs in which the processing times were random variables and the due dates were

* Corresponding author Tel.: +9809126724147
E-mail: hamidrhaddad@gmail.com (H. R. Haddad)

considered as deterministic values. He showed that the problem is NP-hard, but developed the certain conditions where the problem was exactly solvable. Furthermore, he proposed an efficient heuristic, which was capable of finding the near optimum solutions in larger scales of problem.

Balut (1973) presented a chance constrained programming for the problem where the processing times and setup times were independent and followed the normal distribution. Pinedo (1983) considered the problem of minimizing stochastic tardiness when the due dates were deterministic and the processing times followed the exponential distribution. Soroush (2010 b) considered the stochastic problem in which an initial idle time was inserted before the first job of schedule and a quadratic objective function was proposed as minimizing the $\sum w_i l_i^2$ where l_i is the lateness of each job. He presented an exact algorithm based on a precedence relationship structure among adjacent jobs, which that could solve the problem quickly, even for large-scale problems.

Portougal and Trietsch (2006) studied the stochastic version of the setting due date problem by considering the random processing times and solved it by using an $O(n \log n)$ algorithm. The same problem by considering deterministic due date and random processing time was studied by Soroush and Fredendall (1994). Soroush (1999 c) studied the simultaneous due date setting and sequencing problems in stochastic environment in order to minimize earliness and tardiness penalties. He showed that the variations in processing time increase the costs of earliness and tardiness and affect the optimum solutions. He also solved the proposed problem analytically and offered two $O(n \log n)$ heuristic algorithms. Cheng (1986 a) used the total work due date assignment method in stochastic mode when the due dates were proportion to the processing times to minimize expected value of missed due date costs.

In another study, He (1991 b) used the Slack due date assignment method in order to minimize the values of waiting allowances and missed due dates. Seo et al. (2005) presented a new approach based on mathematical programming for the problem of minimizing expected number of jobs in stochastic environment in which the jobs follows normal distribution. By using the mentioned method, the stochastic problem was transmitted to a non-linear optimization problem.

Cheng and Yao (1993) presented a general approach to solve the stochastic scheduling problems. De et al. (1991) studied the problem of minimizing the expected weighted number of tardy jobs when the processing times followed general random variables and the common due date had exponential distribution. Sarin et al. (1991) considered the problem with a common due date and normal processing times to minimize the expected tardiness costs. Cai and Zhou (1997) studied a multi objective stochastic problem to minimize weighted earliness and tardiness penalties and flow time. Jang (2002) studied the problem of minimizing expected number of tardy jobs and presented a dynamic scheduling policy based on myopic heuristic.

In this paper, we consider the single machine stochastic problem with random processing times following the normal distribution, deterministic due dates and independent setup times to minimize the maximum lateness. We also assume that the jobs are deteriorated based on the position that their processes are accomplished, which causes the problem to be more complex in stochastic version. Based on our knowledge, there is no other research, which has studied the deterioration and maximum lateness in stochastic situation, simultaneously. The remaining part of this paper is organized as follows. In Section 2, we formulate and describe the proposed model. In Section 3, we offer the solution approach and in section 4, the computational section is presented.

2. Problem formulation

Consider a problem where one machine and n jobs are available for processing at time zero. The machine works with no idleness and without preemption. Each job has a random processing time, which follows the normal distribution and a deterministic due date. On the other hand, when a job is delayed, its processing time will be affected by a linear deterioration function. It is also assumed that

when the process of a distinct job is finished, an independent setup time is required to provide the settings for next job arrival. The objective function of the problem is to minimize the expected value of maximum lateness. The parameters and variables of the problem are presented as follows:

N	Number of jobs ready to be scheduled
μ_i	The mean of processing time of job i
δ_i	The standard deviation of processing time of job i
d_i	The due date of i -th job
l_j	The lateness of job scheduled on j -th position
μc_j	The mean of completion time of job where located in j -th position
δc_j	The standard deviation of completion time of job where located in j -th position
S	The independent setup time between the jobs
$Fr(.)$	The cumulative normal distribution
b	The rate of deterioration
$y_{ij} = \begin{cases} 1 \\ 0 \end{cases}$	If job i is assigned to position j in the sequence Otherwise

and the mathematical model is as given as follows:

$$\min z = \max l_j \quad (1)$$

subject to

$$\mu c_j = \mu c_{j-1} + \sum_{i=1}^N \sum_{j=1}^N y_{ij} \cdot \mu_i \cdot j^b + s \quad (2)$$

$$\delta c_j = \sqrt{\sum_{i=1}^N \sum_{j=1}^N (y_{ij} \cdot \delta_i \cdot j^{\frac{b}{2}})^2 + \delta c_{j-1}^2} \quad (3)$$

$$\mu c_0 = 0 \quad (4)$$

$$\delta c_0 = 0 \quad (5)$$

$$l_j = \sum_{i=1}^N \sum_{j=1}^N Fr \left| \frac{\mu c_j - d_i}{\delta c_j} \right| \cdot y_{ij} \cdot |\mu c_j - d_i| \quad (6)$$

$$\sum_{i=1}^N y_{ij} = 1 \quad j=1,2,\dots,N \quad (7)$$

$$\sum_{j=1}^N y_{ij} = 1 \quad i=1,2,\dots,N \quad (8)$$

$$y_{ij} = 0 \text{ or } 1 \quad j=1,2,\dots,N \quad i=1,2,\dots,N \quad (9)$$

Eq. (1) tries to find a schedule of jobs, which minimizes the expected value of maximum lateness. Constraint 2 demonstrates that the mean value of completion time of each job is affected by the deterioration and independent setup time and constraint 3 shows how the standard deviations of completion times are calculated. Constraints 4 and 5 declare that both the machine and all of the jobs are ready for processing at time zero. Constraint 6 demonstrates how the value of lateness in stochastic mode is calculated. Constraint 7 declares that each position could be occupied by only one

job and constraint 8 shows that each job is processed by the machine only once. Finally, constraint 9 demonstrates that y is a binary variable.

3. Solution approach

3.1 Simulated annealing

Simulated annealing (SA) is a class of metaheuristics optimization technique, which performs a stochastic neighborhood search through the solution space. SA has been applied to solve many combinatorial optimization problems, widely. The immense advantage of SA over classical local search methods is its ability to avoid getting trapped in local optima while searching for a global optimum (see Van Laarhoven (1998) for more details about the performance of SA). In this case, SA starts with a randomly generated solution in T temperature, which is a sequence of jobs. A sample string consists of four jobs is represented in Fig. 1.

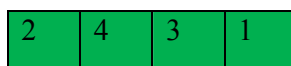


Fig. 1. Representation of a sample string

This procedure continues until the equilibrium condition in this temperature is met. In this problem, the equilibrium condition occurs when the gaps between the proposed objective functions in consecutive iterations in a certain temperature are in minimum possible level. This condition is stated as follows,

$$\sum_{i=Iter-E}^{Iter} |best_i - best_{i-1}| \leq \varepsilon, \quad (10)$$

where, $best_i$ is the best solution until i iterations after previous change in temperature are found, and ε is a predetermined small value. By reaching the equilibrium in the temperature, the temperature is decreased and the procedure starts from the lower temperature and continues until the next equilibrium. Neighborhood search is also implemented by swapping two randomly selected positions in the second string (batch string) to meet other nodes of solution space. The performance of the proposed SA is depicted in below figure.

1. Generate two strings randomly (state k)
2. $T = T_0$
3. Do while $T < T_F$
4. Do while the equilibrium condition has not occurred
5. Generate a neighborhood (state j)
6. $Z = f(j) - f(k)$
7. If $z < 0$ then $k := j$
8. Else if $\text{random}(0,1) < e^{\frac{-\Delta z}{T}}$
9. $K := j$
10. loop
11. $T := T \cdot (\text{cooling rate})$
12. Loop
13. End

Fig. 2. the pseudo-code of proposed SA

In order to calibrate the proposed SA, a Taguchi approach is presented. Taguchi designs recognize that not all factors that cause variability can be controlled in practice. These uncontrollable factors are called noise factors. Based on this fact, Taguchi attempts to identify controllable factors (control factors), which minimizes the effect of the noise factors. During experimentation, the noise factors

are manipulated to force variability to occur and then finding the optimal control factor settings, which makes the process or product robust, or resistant to variation from the noise factors.

Taguchi uses orthogonal arrays, which estimate the effects of factors on the response mean and variation. Orthogonal arrays allow investigating each effect independently of each other and may reduce the time and cost associated with the experiment when fractionated designs are used. In this paper the S/N ratio considered as nominal is the best and is calculated by:

$$\frac{S}{N} \text{ ratio} = 10\log_{10}(\text{objective function})^2 \tag{11}$$

The effective factors and their levels are also described in Table 1.

Table 1

Taguchi experiment inputs

Factor	symbol	levels	type	Degree of freedom
Number of total iterations	A	5	A(1)=100	4
			A(2)=200	
			A(3)=300	
			A(4)=400	
			A(5)=500	
The equilibrium condition (E)	B	5	B(1)=2	4
			B(2)=3	
			B(3)=4	
			B(4)=5	
			B(5)=6	

The associated degree of freedom for these two factors is equal to 8; therefore, according to Taguchi standard table of orthogonal array, the L₂₅ should be selected, which fulfils all the minimum necessary requirements. In order for the Taguchi experiments to be conducted, three important measures are considered containing the S/N ratio (as robust measure), average responses for each combination of control factors and the variability in the response due to the noise (standard deviation). The results are depicted in below figures.

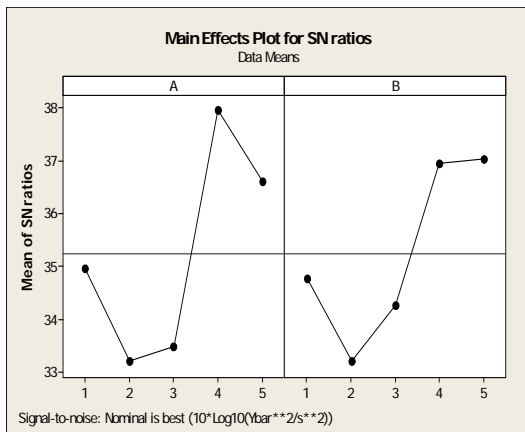


Fig. 3. The results for response based on S/N ratio

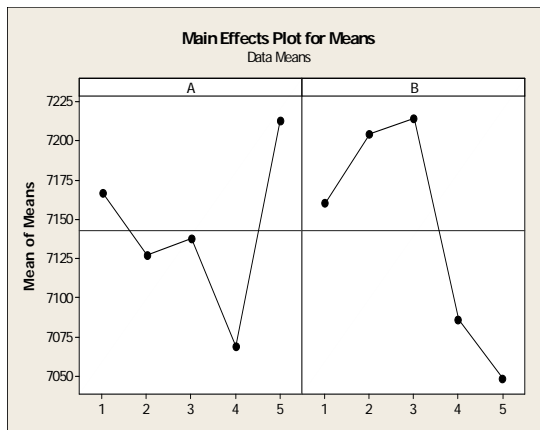


Fig. 4. The results for response based on means

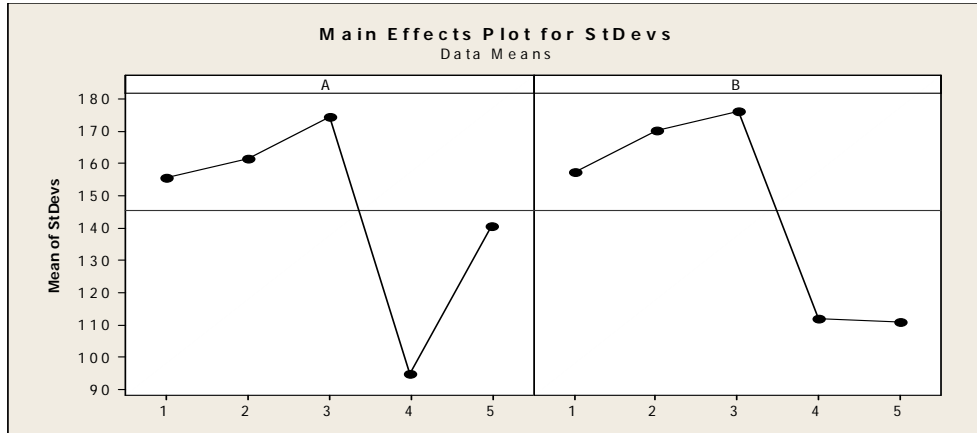


Fig. 5. The results for response based standard deviations

A measure of robustness is used to identify control factors, which reduces variability in a product or process by minimizing the effects of uncontrollable factors. Fig. 3 indicates the robustness of each combination of factors. Clearly, it is desired to select a pair of factors that generate the maximum robustness. Therefore, based on this figure, A(4) and B(5) should be selected. Fig. 4 shows the average responses for each combination of control factors. Since the objective function is in minimization form, the minimum value for this measure is desired; therefore, A (4) and B (5) are selected.

Finally, Fig. 5 shows the variability in the response due to the noise, which is desired to be minimal. Therefore, A (4) and B (5) should be selected. Based on the mentioned measures, the most efficient combination of proposed factors would be A (5) and B (5), which could satisfies the response values in all the mentioned measures.

3.2 Branch and Bound scheme

In this section, a Branch and Bound algorithm (B&B) method is presented to possibly reach the global optimum solutions of the problem. Therefore, the solution space is searched by constructing a depth search fashion tree, in which each node represents assigning a job to a specific position in the schedule. The construction of this tree is shown as follows:

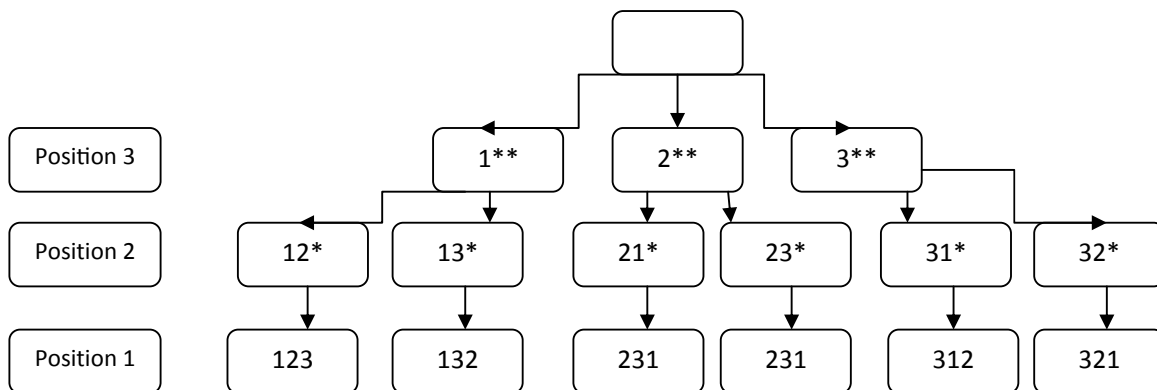


Fig. 6. The construction of the search tree

A node is fathomed if:

- It is a leaf node which means all the jobs have been scheduled.
- In this node the value of considered lower bound is equal or greater than upper bound.

By fathoming a node, the search continues based on the parent of current node, and the algorithm would be terminated if all the nodes have been fathomed.

3.2.1 upper bound

In this problem, the upper bound is calculated once for the root of tree based on the proposed simulated annealing. It is obvious that since SA is a class of meta-heuristic algorithms, its result is greater than the global optimum of the problem.

3.2.2 lower bounds

Lemma1.

During searching the tree, a lower bound value for the objective function of the problem can be found as follows:

- In each node, sequence the unscheduled jobs as EDD order,
- Put the constructed sequence after the scheduled jobs,
- Calculate the value of completion times for the scheduled jobs using the following,

$$\begin{aligned} \mu c_1 &= 0 \\ \delta c_1 &= 0 \\ \mu c_j &= \mu c_{j-1} + \sum_{i=1}^N \sum_{j=1}^N y_{ij} \cdot \mu_i \cdot j^b + s \quad j=2, \dots, n_s \\ \delta c_j &= \sqrt{\delta c_{j-1}^2 + \sum_{i=1}^N \sum_{j=1}^N (y_{ij} \cdot \delta_i)^2} \cdot j^b \quad j=2, \dots, n_s \end{aligned}$$

and n_s is equal to the number of scheduled jobs.

- Calculate the value of completion times for the unscheduled jobs using the following equation:

$$\begin{aligned} \mu c_j &= \mu c_{j-1} + \sum_{i=1}^N \sum_{j=1}^N y_{ij} \cdot \mu_i \quad j=n_s, \dots, n \\ \delta c_j &= \sqrt{\delta c_{j-1}^2 + \sum_{i=1}^N \sum_{j=1}^N (y_{ij} \cdot \delta_i)^2} \quad j=n_s, \dots, n \end{aligned}$$

- Calculate the value of lateness for each job using the following equation,

$$l_j = \sum_{i=1}^N \sum_{j=1}^N Fr \left| \frac{\mu c_j - d_i}{\delta c_j} \right| \cdot y_{ij} \cdot |(\mu c_j - d_i)|$$

- Consider L_{\max} as the maximum value of lateness in constructed sequence.

Proof

For the problem of minimizing maximum lateness without deterioration and setup time, an optimal order could be gathered by ordering the jobs as EDD rule.

First, we show that the objective function of maximum lateness without deterioration and setup time is a lower bound for the objective function of maximum lateness with presence of mentioned factors.

Let's consider c_2 as the value of completion time with no deterioration and setup time effect and c_1 as the value of completion time with considering the deterioration and setup time.

It is obvious that $c_2 \leq c_1$. We must show $L_2 \leq L_1$.

$$\begin{aligned} L_2 \leq L_1 &\rightarrow |c_2 - d| \leq |c_1 - d| \rightarrow (|c_2 - d|)^2 \leq (|c_1 - d|)^2 \\ &\rightarrow c_2^2 - 2 \cdot c_2 \cdot d + d^2 \leq c_1^2 - 2 \cdot c_1 \cdot d + d^2 \rightarrow c_2^2 \leq c_1^2 \end{aligned}$$

Since c_j is a positive variable, so it can be concluded that:

$$c_2 \leq c_1.$$

Therefore

$$l_2 \leq l_1. \tag{12}$$

On the other hand, let's consider $L_{\max}(\text{sch})^*$ and $L_{\max}(\text{usch})^*$ as the optimal values of maximum lateness for scheduled jobs and values of maximum lateness for unscheduled jobs, respectively.

if $L_{\max}(\text{sch})^* < L_{\max}(\text{usch})^*$ then the optimal value is equal to $L_{\max}(\text{usch})^*$

if $L_{\max}(\text{sch})^* > L_{\max}(\text{usch})^*$ then the value is equal to $L_{\max}(\text{sch})^*$

Therefore, from Eq. (11) we have,

$$L_{\max}(\text{usch})^* \leq L_{\max}(\text{sch})^* \tag{13}$$

Since $L_{\max}(\text{sch})^*$ is associated with a partial sequence, its value is less than or equal to the optimal value of the problem.

$$L_{\max}(\text{sch})^* \leq L_{\max}^* \tag{14}$$

From Eq. (13) and Eq. (14) it can be concluded that:

$$L_{\max}(\text{usch})^* \leq L_{\max}^*$$

and the proof is completed ■.

4. Computational study

All the computations for this problem were coded by the Visual Basic 6 software and were run on a core I 7 Intel CPU with 4 GB of RAM. The required data was also generated randomly based on Seo et al. (2005) as follows:

- The means of processing times were selected randomly from uniform [20,30],
- The standard deviations were selected randomly from uniform $[1, \frac{\mu_i}{2.33}]$,
- The due dates were selected from uniform $[0.1 \sum \mu_i, 0.4 \sum \mu_i]$.

The instances were solved by Lingo 10 solver as well as the proposed SA and B&B to determine the capability of proposed methods in solving the problem. For SA, each instance was solved 5 times and the best result was recorded as the value of objective function.

Table 2
Sensitivity analysis based on small problem dimension

instance number	Number of jobs	Lingo solver		SA			B&B		% gap
		VOF	Run time (sec)	VOF	Run time (sec)	% gap	VOF	Run time (sec)	
1	3	32.11	2	32.18	6	0.002	32.11	0.01	0
2	4	54.22	25	54.22	9	0	54.22	0.03	0
3	5	51.49	236	51.66	10	0.003	51.49	0.04	0
4	6	71	256	71.59	2.30	0.008	71	0.14	0
5	7	84.41	507	85.24	5.83	0.009	84.41	0.39	0
6	8	97.77	331	98.77	5.69	0.01	97.77	0.64	0
7	9	116.72	365	117.44	6.16	0.006	116.72	1.38	0
8	10	-	-	141.47	5.92	-	139.32	1.93	-
9	11	-	-	171.29	6.39	-	161.11	4.08	-
10	12	-	-	165.20	9.93	-	165.20	1.94	-
11	13	-	-	99.12	5.58	-	91.18	19.08	-

Table 2 presents the results of proposed SA via the Lingo solver where the column 3, shows the properties of Lingo, including the value of objective function (VOF) and the run times and column 4 presents the results of SA. Lingo solver is incapable of solving the instances, which consist of greater than seven jobs in a reasonable amount of time. As shown in Table 2, the proposed SA performs very well in order to find a near optimal solution in a reasonably low running time. The results of SA are also presented for greater dimensions, which are recorded in Table 3.

Table 3
The results of SA for medium and large problem dimension

instance number	Number of jobs	SA	
		VOF	Run time (sec)
1	20	5890	1.89
2	30	22155	1.75
3	40	40290	1.99
4	50	64129	2.30
5	60	102556	2.41
6	70	136850	2.94
7	80	205559	2.48

As mentioned in solution approach section, the SA is used as an upper bound in the branch and bound procedure. In order to determine the effect of proposed upper bound in pruning the solution space, the problem is also solved without considering the upper bounds. The results are depicted in Table 4, where the column 3 shows the properties of B&B without upper bound, and column 4 is related to B&B when the proposed SA acts as an upper bound. It can be concluded from Table 4 that by considering a suitable upper bound, a greater number of nodes are pruned and branch and bound is able to search the solution space much faster. On the other hand, even without using upper bound, the B&B has the ability to find the global optimum; however, it would require more searching and running times. To make the sensitivity analysis, a number of effective factors of the problem are considered, including the rate of deterioration and common setup time.

Table 4

The impact of SA as an upper bound in B&B procedure

instance number	Number of jobs	B&B		B&B + SA			
		VOF	Run time (sec)	Number of visited nodes	VOF	Run time (sec)	Number of visited nodes
1	3	32.11	0.01	10	32.11	0	6
2	4	54.22	0.03	19	54.22	0.03	10
3	5	51.49	0.04	25	51.49	0.03	15
4	6	71	0.14	94	71	0.12	59
5	7	84.41	0.39	163	84.41	0.34	100
6	8	97.77	0.64	267	97.77	0.50	193
7	9	116.72	1.38	536	116.72	0.54	173
8	10	139.32	1.93	703	139.32	1.09	342
9	11	161.11	4.08	1207	161.11	2.08	576
10	12	165.20	1.94	310	165.20	0.84	106
11	13	91.18	19.08	9088	91.18	4.94	1991

Table 5 presents the results of SA and branch and bound for various rates of deteriorations for six numbers of jobs. In this case, the common setup time is considered equal to 3.

Table 5

Sensitivity analysis based on rate of deterioration for N=6

instance number	Rate of deterioration	Branch and Bound		SA	
		VOF	Run time (sec)	VOF	Run time (sec)
1	0.1	344.83	1.93	349.08	1.25
2	0.2	396.37	1.77	405.08	1.01
3	0.3	453.22	1.96	462.35	0.98
4	0.4	518.05	1.76	518.25	0.84
5	0.5	591.91	1.93	601.11	0.89
6	0.6	674.37	1.69	685	0.73
7	0.7	768.94	1.73	770.48	0.71
8	0.8	876.41	1.77	885.75	0.74
9	0.9	995.62	1.74	1013.16	0.61

Based on the results from Table 5, the value of objective function is sensitive to the rate of deterioration for both methods. However, there is no relationship between the running times and the rate of deterioration. The sensitivity is also checked based on the common setup times and the results are presented in Table 6. In this case, the rate of deterioration is considered equal to 0.1.

Table 6

Sensitivity analysis based on setup time for N=6

instance number	Common setup time	Branch and Bound		SA	
		VOF	Run time (sec)	VOF	Run time (sec)
1	1	291.71	1.87	300.697	1.32
2	3	344.83	1.93	349.08	1.25
3	10	534.82	1.83	538.27	0.84
4	25	943.60	1.96	958.86	0.61
5	50	1619.85	1.87	1619.85	0.52
6	100	2969	1.90	2975.86	0.5

The results of Table 6, indicates that by increasing the setup time, the value of objective function would increase clearly, but the run time is not affected by changing the setup time.

5. Conclusion and future researches

This paper considered the single machine stochastic scheduling in which the processing of jobs were random variables and the due dates were considered as deterministic values. The jobs were also deteriorated based on the position in which their processes were executed. The objective function of the problem was to find a schedule of jobs by minimizing the expected value of maximum lateness. A branch and bound scheme was presented to solve the problem analytically and a simulated annealing meta-heuristic (SA) was offered to solve the problem in larger scales. Some sensitivity analyses were accomplished for a number of effective factors, including rate of deterioration and the common setup time.

Computational experiments demonstrated that the proposed SA acted highly capable of finding near optimal solutions with the least gap. Furthermore, the role of SA was checked as an upper bound for the proposed branch and bound, which expedite the speed of direct solution. For future research, the objective function could be considered as a multi objective along with other objectives, including maintenance, batch delivery costs, etc.

Acknowledgment

The authors would like to thank the anonymous referees for their comments on earlier version of this work.

References

- Balut, S. (1973). Scheduling to minimize the number of late jobs when set-up and processing times are uncertain. *Management Science*, 19, 1283–1288.
- Cai, X., & Zhou, S. (1997). Scheduling stochastic jobs with asymmetric earliness and tardiness penalties. *Naval Research Logistics*, 44, 531–557.
- Chang, C., & Yao, D. (1993). Rearrangement, majorization and stochastic scheduling. *Mathematics of Operations Research*, 18, 658–684.
- Cheng, T. (1986). Optimal due-date assignment for a single machine sequencing problem with random processing times. *International Journal of Systems Science*, 17, 1039–1144.
- Cheng, T. (1991). Optimal assignment of slack due-dates and sequencing of jobs with random processing times on a single machine. *European Journal of Operational Research*, 51, 348–353.
- De, P., Ghosh, E., & Wells, E. (1991). On the minimization of the weighted number of tardy jobs with random processing times and deadline. *Computers and Operations Research*, 18, 457–463.
- Gutjahr, W., & Pflug, G. (1996). Simulated annealing for noisy cost functions. *Journal of Global Optimization*, 8, 1–13.
- Gutjahr, W., Hellmayr, A., & Pflug, G. (1999). Optimal stochastic single-machine-tardiness scheduling by stochastic branch-and-bound. *European Journal of Operational Research*, 117, 396–413.
- Jang, W. (2002). Dynamic scheduling of stochastic jobs on a single machine. *European Journal of Operational Research*, 138, 518–530.
- Norkin, I., Ermoliev, Y., & Ruszczyński, A. (1994). *On optimal allocations of indivisibles under uncertainty*. Austria: Laxenburg.
- Pinedo, M. (1983). Stochastic scheduling with release dates and due dates. *Operations Research*, 31, 559–572.
- Portougal, V., & Trietsch, D. (2006). Setting due dates in a stochastic single machine environment. *Computers & Operations Research*, 33, 1681–1694.

- Sarin S. C, Erel, E., Steiner, G. (1991). Sequencing jobs on a single machine with a common due dates and stochastic processing times. *European Journal of Operational Research*, 51, 188–198.
- Seo, D., Klein, C., & Jang, W. (2005). Single machine stochastic scheduling to minimize the expected number of tardy jobs using mathematical programming models. *Computers & Industrial Engineering*, 48, 153–161.
- Sorouch, H., & Fredendall, L. (1994). The stochastic single machine scheduling problem with earliness and tardiness costs. *European Journal of Operational Research*, 77, 287–302.
- Sorouch, H. (2007). Minimizing the weighted number of early and tardy jobs in a stochastic single machine scheduling problem. *European Journal of Operational Research*, 181, 266–287.
- Sorouch, H., (2010). Solving a stochastic single machine problem with initial idle time and quadratic objective. *Computers & Operations Research*, 37, 1328–1347.
- Sorouch, H., (1999). Sequencing and due-date determination in the stochastic single machine problem with earliness and tardiness costs. *European Journal of Operational Research*, 113, 450-468.
- Van Laarhoven, P., & Aarts, E.H.(1988). *Simulated Annealing: Theory and Applications*. Kluwer Academic Publishers, Dordrecht.