

## A new mathematical model for single machine batch scheduling problem for minimizing maximum lateness with deteriorating jobs

Hamidreza Haddad\*, Payam Ghanbari, and Ahmad Zeraatkar Moghaddam

Department of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran

### ARTICLE INFO

#### Article history:

Received 10 March 2010

Received in revised form

July, 25, 2011

Accepted 26 July 2011

Available online

27 July 2011

#### Keywords:

Batch scheduling

Single machine

Deterioration

Simulated annealing

### ABSTRACT

This paper presents a mathematical model for the problem of minimizing the maximum lateness on a single machine when the deteriorated jobs are delivered to each customer in various size batches. In reality, this issue may happen within a supply chain in which delivering goods to customers entails cost. Under such situation, keeping completed jobs to deliver in batches may result in reducing delivery costs. In literature review of batch scheduling, minimizing the maximum lateness is known as NP-Hard problem; therefore the present issue aiming at minimizing the costs of delivering, in addition to the aforementioned objective function, remains an NP-Hard problem. In order to solve the proposed model, a Simulation annealing meta-heuristic is used, where the parameters are calibrated by Taguchi approach and the results are compared to the global optimal values generated by Lingo 10 software. Furthermore, in order to check the efficiency of proposed method to solve larger scales of problem, a lower bound is generated. The results are also analyzed based on the effective factors of the problem. Computational study validates the efficiency and the accuracy of the presented model.

© 2012 Growing Science Ltd. All rights reserved

## 1. Introduction

This paper investigates the problem of a single machine batch scheduling to minimize the maximum lateness with delivery costs, where that all jobs are delivered to their customers in some batches. This problem depicts two stages supply chain scheme, which contains a manufacturer and some customers in which each job may be transmitted to its customer right after its completion or to be held in batch with more jobs. Batching the jobs not only may cause the job process to accelerate, but also it can decrease the delivery cost of job transmission to their customers.

Lu and Yuan (2007) proved that the single machine batch scheduling with the objective of maximum lateness is strongly NP-hard, therefore by adding a term that corresponds to the delivery cost, the problem remains NP-hard. The literature of single machine batch scheduling can be classified by variant concepts. For example, based on how the batches are processed on the machine, the problem would contain serial or parallel batching categories. In serial batching it is assumed that each batch is processed based on all the processing time of its jobs and the processing time of each batch is equal

\* Corresponding author Tel.: +009802122494417  
E-mail: hamidhaddad@gmail.com (H. Haddad)

to the sum of the processing jobs are located on it. Ng et al. (2002 b) studied the problem of single machine serial batch scheduling to minimize maximum lateness by considering the precedence constraints between the batches and presented an  $o(n^2)$  polynomial time algorithm. They (2003 a) also investigated the same problem by total completion time objective, release date and precedence constraints and presented an  $o(n^5)$  polynomial algorithm.

Yuan et al. (2007) considered the serial batching scheduling when each batch contained exactly  $k$  jobs and the objective was to minimize total weighted completion time. They showed that the problem is NP-hard and proposed a polynomial algorithm with some simplified rules that were:

- a. the jobs were inversely agreeable i.e; if  $p_i < p_j$  then  $w_i > w_j$
- b. the weight of jobs was a multiple of their processing times ( $w_i = \alpha \cdot p_i$ )

The other researches in the field of serial batching can be mentioned as:

An  $o(n \log n)$  algorithm for minimizing total completion time by Coffman (1990), an  $o(n^2)$  algorithm for minimizing total weighted completion time with precedence constraint by Albers and Bruker (1993) and an  $o(n^{14})$  algorithm for minimizing total completion time by Baptice (2000). On the other hand the processing time of a parallel batch is calculated by the maximum processing time of jobs are apparent to that batch. Nong et al. (2008 a) investigated the problem of single machine parallel batching to minimize makespan by considering family setups and release date. They developed a polynomial approximation scheme for their problem that yielded to an algorithm with worst case ratio of 2.5.

Potts and Kovalyov (2000) accomplished an extensive review in the field of parallel batching literature and gave details of basic solution methods. Another classification can be offered as on-line or off-line batching scheduling. On-line scheduling means that the job information, like processing time, is unknown before the arrival of jobs.

Nong et al. (2008 b) proposed an on-line scheduling problem and offered two models in which the first model was unbounded and machine could process infinite jobs and the other one was a bounded model. They also solved the models by a 2 worst case ratio algorithm. Zhang et al. (2001) studied the problem of minimizing makespan on a parallel batch by considering the delay effect and solved the problem with various sizes containing bounded and unbounded models. Chen et al. (2004) proposed an on-line scheduling problem to minimize total weighted completion time. Azizoglu and Webster (2001) provided a branch and bound method to solve the on-line parallel batched scheduling with the objective of minimizing total weighted completion time where the size of batches was limited.

In offline scheduling structure, all the details about the jobs are gathered before the process begins. Yuan et al. (2004) proved that the offline version of single machine batching problem is strongly NP-hard for bounded and unbounded models and proposed two dynamic programming and a heuristic algorithm for bounded models. In recent years, researchers have considered the delivery cost as a secondary objective called "combined optimization batch delivery problem". Mahdavi et al. (2011a) studied the problem of minimizing the number of weighted tardy jobs by considering delivery cost and used a simulation annealing approach to solve the problem. Tian et al. (2007) investigated the problem of minimizing the sum of total weighted flow times with delivery cost. They indicated that the problem is NP-hard and proposed an optimal algorithm. Mahdavi et al. (2011b) proposed a branch and bound approach to solve the problem of minimizing the weighted sum of flow times with delivery cost. Another research related to Mahdavi et al. (2007) presented a branch and bound to minimize flow time and delivery cost, simultaneously.

One of the most important problems considered in scheduling problem is job deterioration where there are wide applications in many real-world situations. Deterioration means that the processing time of jobs increases over time.

Browne and Yechiali (1990) introduced a job deterioration scheduling problem in which the processing time of a job was a function of its starting time. Mosheiov (1994) who was the first to apply simple linear deteriorating for single machine scheduling problem considered this problem to minimize the makespan, total completion time, total weighted completion time, total weighted waiting time, and total tardiness, number of tardy jobs, maximum lateness and maximum tardiness.

Wu et al. (2009) solved a single-machine problem by a branch-and-bound scheme and two heuristic algorithms to minimize the makespan under the piecewise linear deterioration model. Wang et al. (2009) solved the single machine scheduling problems with learning effect and deteriorating jobs simultaneously to minimize the total weighted completion time and the maximum lateness. Wang and Wang (2010) considered a single machine scheduling problem with deteriorating jobs in which the processing time of a job was defined as a simple linear function of its starting time to minimize the total weighted earliness penalty subject to no tardy jobs. Husang et al. (2010) considered a single machine scheduling problem with deteriorating jobs by a linear function of time to minimize the total weighted earliness penalty subject to no tardy jobs. Cheng and Ji (2010) proposed a single machine scheduling problem in which the jobs were processed in batches and the processing time of each job was a simple linear function of its waiting time to minimize the makespan. Al Anzi et al. (2007) studied the combination of deteriorating and scheduling objectives with application in computer communication and reverse logistics.

In this paper, we propose a new mathematical model for the problem of serial batched scheduling problem where the objective is to minimize the maximum lateness and delivery cost simultaneously. Since the problem is strongly NP-hard, a simulation annealing approach is used to get near optimal solutions. In order to check the verification of SA, some data sets of problem are solved using Lingo software optimally and the results are compared together. The remaining parts of this paper are as follows: in section 2 the mathematical model is presented and the variables and parameters are introduced. In section 3 the solution approach is offered and in section 4 the computational studies are presented.

## 2. Problem formulation

Consider a machine that processes  $n$  jobs with no preemption where all jobs are available to process at time zero. The completed jobs can be delivered to the customer immediately after completion or they are kept on for the following jobs to deliver as a batch. According to job numbers,  $n$  batches are considered and the jobs are assigned to the batches. Clearly, any batch with no job will be omitted and the sequence of batches is determined in such a way that the maximum lateness and delivery cost are minimized. As mentioned, the serial batching assumption is considered in this paper, so that the processing time of each batch is calculated as the sum of all the processing jobs located on and the due date of batch is the average of the due date of assigned jobs.

The decision variables and parameters of the problem are as below:

$N$	Number of jobs ready to be scheduled
$p_i$	The normal processing time of job $i$
$d_i$	The due date of $i$ -th job
$p_{bat}(k)$	The processing time of batch $k$
$D_{bat}(k)$	The due date of batch $k$
$n_k$	Number of jobs assigned to batch $k$
$L_j$	The lateness of job scheduled on $j$ -th position
$c_j$	The completion time of job where scheduled on $j$ -th position
$b$	The rate of deterioration

$$y_{ik} = \begin{cases} 1 & \text{If job } i \text{ is assigned to batch } k \\ 0 & \text{Otherwise} \end{cases}$$

$$x_{kj} = \begin{cases} 1 & \text{If batch } k \text{ is located in } j\text{-th position in the sequence} \\ 0 & \text{Otherwise} \end{cases}$$

$$a_k = \begin{cases} 1 & \text{If there is at least one job to batch } k \\ 0 & \text{Otherwise} \end{cases}$$

The proposed model is presented as follows:

$$\min z = \sum_k c_k \cdot a_k + \max_k (|l_k|) \quad (1)$$

subject to

$$c_j = c_{j-1} + \sum_j x_{kj} \cdot (p_{bat}(k) \cdot j^b) \quad K = 1, 2, \dots, N \quad (2)$$

$$c_0 = 0 \quad (3)$$

$$p_{bat}(k) = \sum_i y_{ik} \cdot p_i \quad K=1,2,\dots,N \quad (4)$$

$$n_k = \sum_i y_{ik} \quad K=1,2,\dots,N \quad (5)$$

$$a_k \cdot n_k > n_k \quad K=1,2,\dots,N \quad (6)$$

$$L_j = c_j - \sum_j D_{bat}(k) \cdot x_{kj} \quad K=1,2,\dots,N \quad (7)$$

$$D_{bat}(k) = \frac{\sum_i y_{ik} \cdot d_i}{n_k} \quad n_k \neq 0, \quad K=1,2,\dots,N \quad (8)$$

$$\sum_j x_{kj} = 1 \quad k=1,2,\dots,N \quad (9)$$

$$\sum_k x_{kj} = 1 \quad j=1,2,\dots,N \quad (10)$$

$$\sum_k y_{ik} = 1 \quad i=1,2,\dots,N \quad (12)$$

$$x_{kj} = 0 \text{ or } 1 \quad (13)$$

$$y_{ik} = 0 \text{ or } 1 \quad (14)$$

$$L_j = \text{free} \quad j=1,2,\dots,N \quad (15)$$

Equation (1) introduces the objective function where the first term corresponds to the delivery cost and the second term determines a schedule so that the completion time of batches has minimal variations to their due dates. Constraint (2) states that the completion time of each batch is equal to the completion time of prior batch plus the sum of processing times of the assigned jobs. Constraint (3) mentions that the machine and all the jobs are available at time zero. Constraint (4) declares that the completion time of each batch is equal to the sum of processing times of each assigned job.

Constraint (5) determines how many jobs are located on each batch. Constraint (6) determines if the batch  $k$  is empty or not. Note that since  $a_k$  is a binary variable and is used for the objective function and the purpose of the objective function is minimization, the tendency of  $a_k$  is to reach to the zero value. However, this constraint imposes  $a_k$  to have the value of one if the batch is not empty. Constraint (7) clarifies that the lateness of each batch is equal to the difference between the time that the process of the batch and its due date have been completed. Clearly, the value of lateness is either positive or negative. Eq. (8) shows how the due date of each batch is calculated. Constraint (9) states that each batch is processed only one time at each sequence and constraint (10) determines that each position can be assigned to just one batch. Constraint (11) mentions each job can be assigned to exactly one batch and constraints (12) and (13) show that  $x$  and  $y$  are binary variables respectively and finally constraint (14) states  $L$  is a free variable that can take on positive, zero, or negative values.

### 3. Solution approach

#### 3.1 The proposed SA description

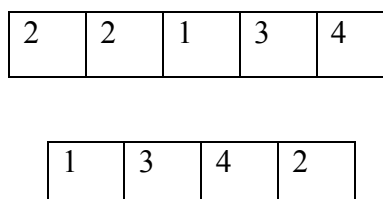
As mentioned in the literature, the problem of minimizing maximum lateness is strongly NP-hard and the delivery cost increases the complexity. Therefore, the use of ordinary optimization methods is not reasonable. In this case, a simulation annealing meta-heuristic is implicated to reach near optimal solutions in reasonable amount of running times.

Simulated annealing (SA) has been widely applied to solve many combinatorial optimization problems. It performs a stochastic neighborhood search of the solution space and it is considered as a Markov chain (1988), so its sensitivity to the initial solution is much less than the other meta-heuristics and has a great ability to avoid trapping to local optimum solutions. The basic principles of the SA can be described as follow:

Start from a current solution  $x$ , another solution  $y$  is generated by taking a stochastic step in some neighborhood of  $x$  while the degree of neighborhood is optional and apparently changes from one problem to another. If the new proposal improves the value of the objective function, then  $x$  is replaced by  $y$  as the new current solution. Otherwise, the new solution  $y$  is accepted along with a probability. This probability comes from the following equations,

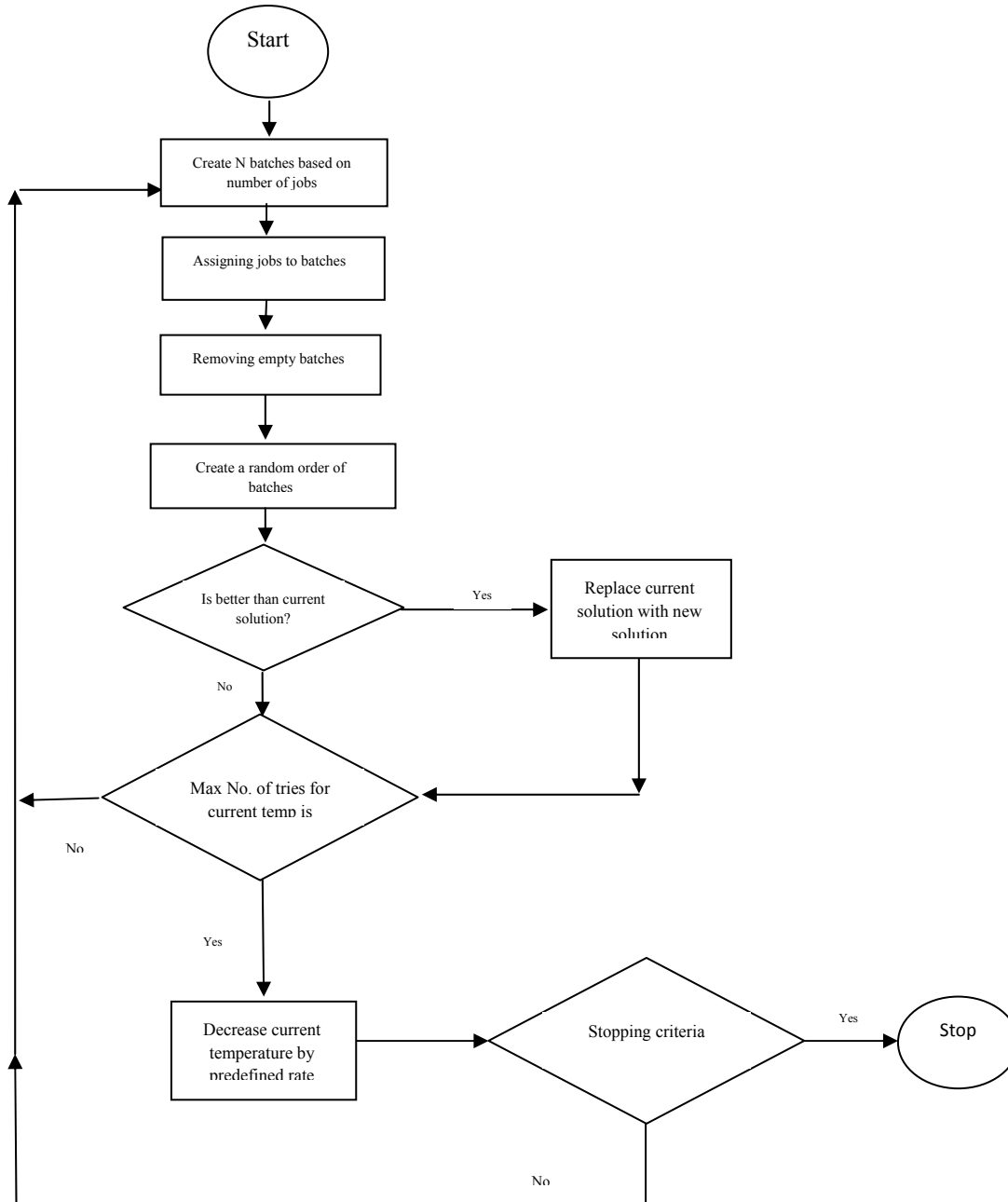
$$P(\Delta E) = \exp(\Delta E / T). \quad (16)$$

The value of  $T$  decreases as the number of iterations increases. Hence, the probability of accepting worse solutions will decrease too. In fact, any reduction in control parameter leads to maintain the current solution and just accepting the improving solutions while the algorithm is running in the last iterations. All the mentioned procedures enable SA to avoid getting trapped in local optima. In this case, two strings are proposed as shown in Fig 1. The first one corresponds to the assigning jobs to the batches and the second one is associate with scheduling the final batch. Therefore, jobs are randomly assigned to different batches and then the created batches are scheduled by swapping the batches. In each iteration, the best value of objective function is recorded. This procedure continues until the stopping condition of algorithm is achieved.



**Fig. 1.** The coding scheme of proposed SA

For example, suppose there are 5 jobs to be assigned to 5 batches. Fig. 1 shows that the first and the second jobs are assigned to batch 2, the third job is located on batch 1 and forth and fifth jobs are assigned to batches 3 and 4, respectively. Since batch 5 remains empty, it will be removed. Fig.2 depicts a sample of the scheduling of created batches where batch 1 is assigned to the first position, batch 3 is assigned to the second position and batch 4 and batch 2 are assigned to the third and the forth orders, respectively. Neighborhood search is also implemented by swapping two randomly selected positions in the second string (batch string) in order to meet other nodes of solution space. The performance of the proposed SA is depicted in Fig 2.



**Fig. 2.** The performance of proposed SA

### 3.2 Calibration the SA parameters

In order to calibrate the proposed SA, a Taguchi approach is presented. This approach is an experimental design methodology, which allows choosing a product or process that performs more consistently in the operating environment. 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) that minimize 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 that make 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 from the others and may reduce the time and cost associated with the experiment when fractionated designed are used. In this paper, the S/N ratio considered as nominal is the best of the kind and is calculated by the following,

$$\frac{S}{N} \text{ ratio} = 10 \log_{10}(\text{objective function})^2 \tag{17}$$

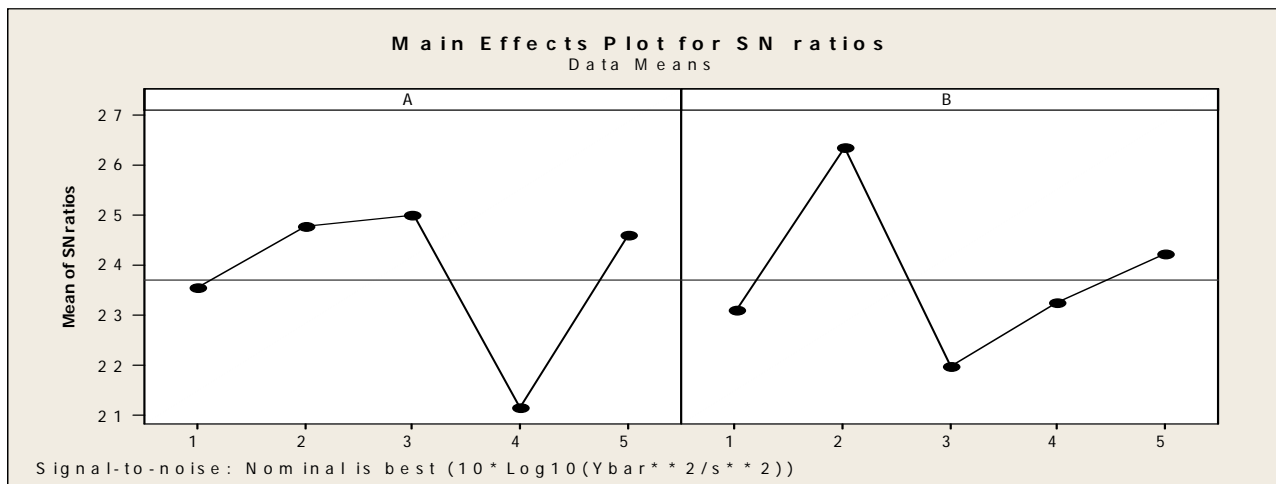
The effective factors and their levels are also described in Table 1.

**Table 1**  
The Taguchi experiment inputs

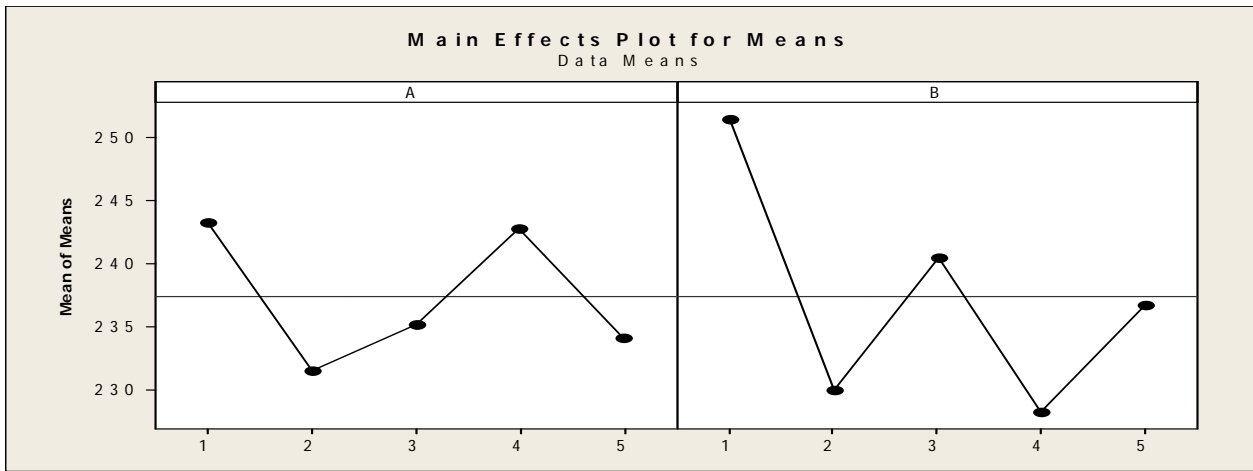
Factor	Symbol	Levels	Type	Degree of freedom
Number of total iterations	A	5	A(1)=250 A(2)=500 A(3)=1000 A(4)=2000 A(5)=5000	4
Number of iterations per temperature	B	5	B(1)=4 B(2)=6 B(3)=8 B(4)=10 B(5)=15	4

The associated degree of freedom for these two factors is equal to 8. According to Taguchi standard table of orthogonal array, the  $L_{25}$  that fulfils all the minimum necessary requirements should be selected. In order to conduct the Taguchi experiments, three important measures are considered as the S/N ratio (as robust measure), the average responses for each combination of control factor and the variability in the response due to the noise (standard deviation).

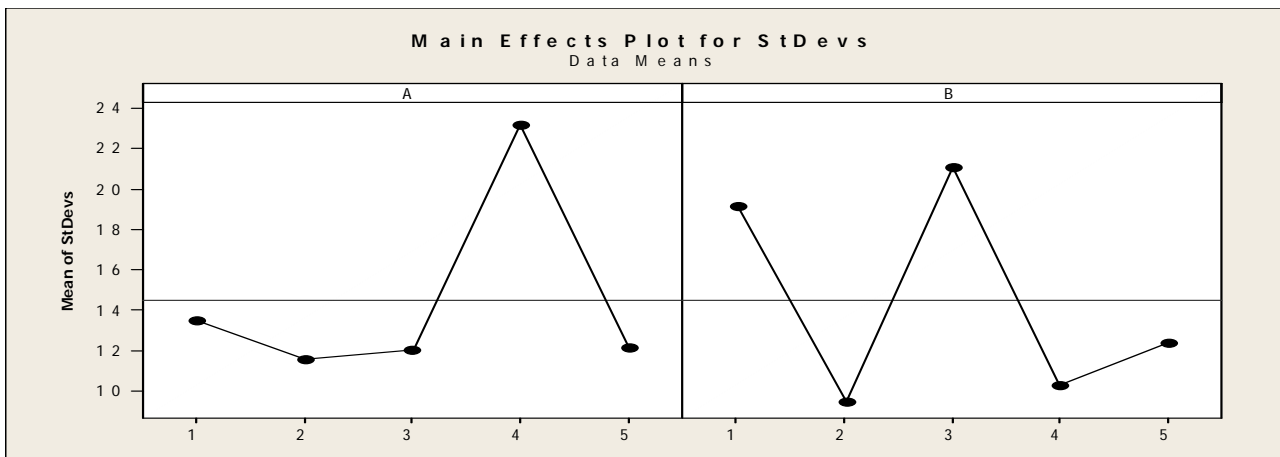
The results are depicted in below figures.



**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

In Taguchi design, a measure of robustness is used to identify control factors that reduce 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. So based on this figure A(3) and B(2) are selected.

Fig. 4 shows the average responses for each combination of control factors. Since the objective of the function is minimization the minimum value for this measure is desired, so A (2) and B (4) are selected. Finally Fig. 5 shows the variability in the response due to the noise that is desired to be minimal, so A(2) and B(2) are selected. Based on the mentioned measures, the most efficient combination of the proposed factors are as A(2) and B(2) that better satisfies the response values.

#### 4. Computational experiment

All the instances of this problem were coded by Visual Basic 6 and they were run on personal computer with CORE I 7 processor with 4 GB of RAM. The required data were generated randomly based on below scheme:

- Processing time of jobs from uniform distribution [1-100]
- Due date of jobs from uniform distribution  $[1- \alpha \sum p]$  where  $\alpha$  considered equal to 0.2 manually.



The instances also are solved using Lingo 10 software to determine the efficiency and capability of proposed SA to reach the global optimum.

To make the sensitivity analyze, two important factors include problem dimension, which are the number of considered batches and the rate of deterioration are considered.

#### 4.1 sensitivity analyze based on the problem dimension

In order to implement this analyze first, some small sizes of problem dimension are considered and the results are compared with equivalent Lingo results. However, Lingo is incapable of solving the problem for more than 7 jobs, therefore, the results of medium and large scale problems are offered individually. The rate of deterioration is-also considered as 0.2 in this section. Table 2 depicts the comparison of results of SA and Lingo based on the problem dimension.

**Table 2**  
Sensitivity analyze based on small problem dimension

instance number	Number of batches	Global optimum		SA		% gap
		VOF	Run time (sec)	VOF	Run time (sec)	
1	3	84.11	12	84.11	6	0
2	4	181	77	181	7	0
3	5	262	588	265.12	8	1.19
4	6	289.26	595	300.97	10	4.04
5	7	360.07	1194	387.08	16	7.50
6	8	-	-	591.18	23	-
7	9	-	-	653.32	25	-
8	10	-	-	669.68	27	-

The second column indicates the number of batches (jobs) and the third and fourth columns show the performance of Lingo software including value of objective function (VOF) and its run time to seconds.

For each instance, the SA was run 5 times and the best obtained solution was considered as the VOF recorded in column 5.

Finally, the gap of results between SA and global optimum is calculated by:

$$\%gap = \frac{VOF(SA) - VOF(global)}{VOF(global)} * 100 \quad (18)$$

Based on the results of Table 1, it is not possible to find the optimal solution for large-scale problems. Therefore, we propose a lower bound to compare the performance of the proposed SA for large-scale problems.

#### Theorem 1

Removing the deterioration effect yields a lower bound value for the objective function.

#### Proof

Let  $c_1$  and  $c_2$  be the completion time with and without deterioration, respectively.

Since  $c_2 \leq c_1$ , we show that  $L_2 \leq L_1$ . The proof is straightforward since

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

Since  $c_j$  is a positive variable it can be concluded that  $c_2 \leq c_1$  ■

Table 3 presents the performance of the proposed SA against the lower bound where the impact of deterioration has been disregarded.

**Table 3**  
Sensitivity analyze based on larger problem dimension

instance number	Number of batches	SA		Lower bound	
		VOF	Run time (sec)	VOF	Run time (sec)
1	5	265.12	8	244	26
2	10	565	62	488	480
3	12	691	74	612	501
4	14	774	87	687	563
5	15	1036	88	876.5	600
6	17	861	92	765	625
7	20	1424	110	1207.7	660
8	25	1859	132	1514.04	912

#### 4.2 Sensitivity analyze based on the rate of deterioration

The effect of deterioration is evaluated by dimension of 4 jobs. Table 3 illustrates the results of SA and Lingo for several rates of deterioration.

**Table 4**  
Sensitivity analyze based on rate of deterioration for N=4

instance number	Rate of deterioration	Global optimum		SA	
		VOF	Run time (sec)	VOF	Run time (sec)
1	0.1	169.51	28	180.19	8
2	0.2	181	84	181	8
3	0.3	184.46	155	184.46	7
4	0.4	198.21	43	198.21	8
5	0.5	214	42	221	8
6	0.6	221	298	221	7
7	0.7	221	57	221	7
8	0.8	221	73	221	7
9	0.9	221	30	221	7

According to the results presented in Table 4, the rate of deterioration impacts the performance of Lingo results, significantly. However, the performance of our SA does not seem to be sensitive to the rate of deterioration in terms of running time. Table 5 summarizes the results of SA for larger sizes of problems.

**Table 5**  
Sensitivity analyze based on rate of deterioration for N=50

instance number	Rate of deterioration	SA	
		VOF	Run time (sec)
1	0.1	4250.47	125
2	0.2	5582.53	103
3	0.3	7127.69	95
4	0.4	9185	85
5	0.5	12629	80
6	0.6	17080	71
7	0.7	23169	68
8	0.8	30939	65
9	0.9	41849	66

## 5. Conclusion and future research

This paper studied the problem of single machine batch scheduling with the objective of minimizing the maximum lateness. The proposed model of this paper considered a new mathematical model where jobs were deteriorated and the processing times depend on the accomplished sequence. In order to solve the proposed model, a calibrate simulation annealing was used and its results were compared with global optimum for small dimensions of problem. We also provided a lower bound to compare the performance of the proposed model for large-scale problems. For future research, the solution method can be considered as the hybrid of meta-heuristics, which could generate better results and we leave it for future research for interested researchers.

## Acknowledgment

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

## References

- Al-Anzi, F. S., Allahverdi, A., & Kovalyov, M.Y. (2007). Batching deteriorating items with applications in computer communication and reverse logistics. *European Journal of Operational Research*, 182(3), 1002–1011.
- Albers, S., & Brucker, P. (1993). The complexity of one -machine batching problems. *Discrete Applied Mathematics*, 47(2), 87–107.
- Azizoglu, M., & Webster, S. (2001). Scheduling a batch processing machine with incompatible job families. *Computers & Industrial Engineering*, 39, 325–335.
- Baptiste, P. (2000). Batching identical jobs. *Mathematical Methods of Operations Research*, 52, 355–367.
- Browne, S., & Yechiali, U. (1990). Scheduling deteriorating jobs on a single processor. *Operations Research*, 38(3), 495–498.
- Chen, B., Deng, X.T., & Zang, W.A. (2004). On-line scheduling a batch processing system to minimize total weighted job completion time. *Journal of Combinatorial Optimization*, 8, 85–95.
- Cheng, T.C.E., & Ji, M. (2010). Batch scheduling of simple linear deteriorating jobs on a single machine to minimize makespan. *European Journal of Operational Research*, 202(1), 90–98.
- Coffman, E., Yannakakis, M., Magazine, M.J., & Santos, C (1990). Batch sizing and sequencing on a single machine. *Annals of Operations Research*, 26, 135–147.
- Huang, X., Wang, J.B., & Wang, X. R. (2010). A generalization for single-machine scheduling with deteriorating jobs to minimize earliness penalties. *International Journal of Advance Manufacturing Technology*, 47(9-12), 1225–1230.
- Lu, L.F., & Yuan, J.J. (2007). The single machine batching problem with identical family setup times to minimize maximum lateness is strongly NP-hard. *European Journal of Operational Research*, 177, 1302–1309.
- Mahdavi Mazdeh, M., Hamidinia, A., & Karamouzian, A. (2011a). A mathematical model for weighted tardy jobs scheduling problem with a batched delivery system. *International Journal of Industrial Engineering Computations*, 2, 491-489.
- Mahdavi Mazdeh, M., Sarhadi, M., & Hindi, K.S. (2007). A branch-and-bound algorithm for single-machine scheduling with batch delivery minimizing flow times and delivery costs. *European Journal of Operational Research*, 183, 74–86
- Mahdavi Mazdeh, M., Shashaani, S., Ashouri, A., & Hindi, K.S. (2011b). Single-machine batch scheduling minimizing weighted flow times and delivery costs. *Applied Mathematical Modeling*, 35, 563–570.
- Mosheiov, G. (1994). Scheduling jobs under simple linear deterioration, *Computers and Operations Research*, 21, 653–659.

- Ng, C.T., Cheng, T.C.E., Yuan, J.J., & Liu, Z.H. (2003). On the single machine serial batching scheduling problem to minimize total completion time with precedence constraints, release dates and identical processing times. *Operations Research Letters*, 31, 323 – 326.
- Ng, CT., Cheng, TCE., & Yuan, JJ. (2002). A note on the single machine serial batching scheduling problem to minimize maximum lateness with precedence constraints. *Operations Research Letters*, 30, 66 – 68.
- Nong, Q., Ng, CT., & Cheng, TCE. (2008). The bounded single-machine parallel-batching scheduling problem with family jobs and release dates to minimize makespan. *Operations Research Letters*, 36, 61 – 66.
- Nong, Q., Yuan, JJ., Fu, R., Lin, L., & Tian, J.I. (2008). The single-machine parallel-batching on-line scheduling problem with family jobs to minimize makespan. *International Journal of Production Economics*, 111, 435–440.
- Potts, C.N., & Kovalyov, M.Y. (2000). Scheduling with batching: a review. *European Journal of Operational Research*, 120, 228–249.
- Tian, J., Fu, R., & Yuan, J. (2007). On-line scheduling with delivery time on a single batch machine. *Theoretical Computer Science*, 374, 49–57.
- Van Laarhoven, P.J.M., & Aarts, E.H.(1988). *Simulated Annealing: Theory and Applications*. Kluwer Academic Publishers, Dordrecht.
- Wang, D., & Wang, J.B. (2010). Single-machine scheduling with simple linear deterioration to minimize earliness penalties, *International Journal of Advance Manufacturing Technology* 46, 285–290.
- Wang, J.B., Huang, X., Wang, X.Y., Yin, N., & Wang, L. (2009). Learning effect and deteriorating jobs in the single machine scheduling problems. *Applied Mathematical Modeling*, 33, 3848–3853.
- Wu, C.C., Shiau, Y.R., Lee, L.H., & Lee, W.C. (2009). Scheduling deteriorating jobs to minimize the makespan on a single machine. *International Journal of Advance Manufacturing Technology*, 44, 1230–1236.
- Yuan, J.J., Lin, X.Y., Cheng, T.C.E., & Ng, C.T. (2007). Single machine serial-batching scheduling problem with a common batch size to minimize total weighted completion time. *International Journal of Production Economics*, 105, 402–406.
- Yuan, J.J., Liu, Z.H., Ng, C.T., & Cheng, T.C.E. (2004). The unbounded single machine parallel batch scheduling problem with family jobs and release dates to minimize makespan. *Theoretical Computer Science*, 320, 199–212.
- Zhang, G., Cai, X., Lee, C.Y., & Wong, C.K. (2001). Minimizing makespan on a single batch processing machine with no identical job sizes. *Naval Research Logistics*, 48, 226–240.