VOL. 1, NO. 1, December 2012

International Journal of Economics, Finance and Management
©2012. All rights reserved.

http://www.ejournalofbusiness.org

Two Hybrid Algorithms for Solving the Multi Objective Batch Scheduling Problem on a Single Machine Based on Simulated Annealing and Clustering Methods

[1] **Hamidreza Haddad,** [2] **Payam Ghanbari**

[1,2] Iranian university of science and technology, Narmak, Tehran, Iran

ABSTRACT

Single machine batching scheduling is one of the most important problems in the manufacturing area which has applied many applications especially in field of supply chain management. In the real world industry, the manufacturers require a suitable plan so as to deliver the finished items to their customers with minimum delivery and tardiness cost.

This paper studies the present problem with objective of minimizing the total tardiness and maximizing the job values on a single machine when the deteriorated jobs are delivered to each customer in various size batches and proposed a mathematical model for this target. Furthermore, all jobs are not ready for process at time zero and each job is ready based on a predefined release date.

In order to solve the proposed model, two hybrid algorithms, based on simulated annealing and clustering methods are offered and the results are compared with the global optimums that are generated by Lingo 10 software. Furthermore, based on the effective factors of the problem, a number of sensitivity analyses are also implemented.

Computational study demonstrates that using clustering methods leads to specified improvements in batching process.

Keywords: *Batch scheduling; single machine; deterioration; job values; clustering; release date*

1. INTRODUCTION

The assumption of Batching is one of the most important assumptions that approach the problem of single machine scheduling in the real industries. By using this assumption, the scheduling models could be developed as a supply chain with several stages.

Generally, two types of batching problem have been studied in the literature of scheduling. The first model is related to the production systems in which jobs are grouped into separate batches before being processed on the machine and the generated batches are processed. This problem can be considered as a parallel or serial batching problem. In parallel batching problem, the completion time of each batch is equal to the maximum processing time of jobs that belong to that batch.

On the other hand, in serial batching problem, the completion time of each batch is considered as the completion time of the last job assigned to it.

Ng et al. (2002) studied the problem of serial batch scheduling on a single machine with consideration of precedence constraint and solved it by an $o(n^2)$ polynomial time algorithm.

Ishii et al. (2010) presented a Lagrange relaxation in order to solve the problem of single machine batch scheduling with three objectives involving makespan, maximum flow time and the satisfaction level. In the proposed model, the size of each batch was considered as a fuzzy number.

Cheng et al. (2001 b) studied the problem of single machine batch scheduling with resource dependent setup with objective of minimizing total weighed resource consumption. The proposed model was solved in polynomial run time.

Cheng and Kovalyov (2000) accomplished an extensive research about different cost functions in single machine batch scheduling, including maximum lateness, number of late jobs, total tardiness, total weighted completion time and total weighted tardiness. They also solved some of the proposed objectives by using a dynamic programming approach.

Nong et al. (2008) studied the impact of family setups and release date in the problem of single machine parallel batching with objective of minimizing makespan and a polynomial approximation scheme was developed by some simplifications.

Other researches that are related to the single machine batch scheduling are Ng et al. (2004 c), Coffman (1990), Albers and Bruker (1993) and Baptice (2000).

The second model is related to the distribution systems and develops the scheduling problems in a supply chain. In such problems that entitles batch delivery scheduling, jobs are processed independently on the machine and after completion are grouped into some batches and generated batches are sent to the costumers. Each completed job may be held for finishing the completion of next jobs or sent individually. Most of these researches consider the delivery cost in addition to a scheduling function for the problem.

©2012. All rights reserved.

Mahdavi Mazdeh et al. (2011 a) presented a simulated annealing for solving the problem of minimizing the number of weighted tardy jobs with delivery cost so as to find the near optimal solution.

Tian et al. (2007) studied the problem of minimizing the sum of total weighted flow times with delivery cost and proposed an optimal algorithm for solving it. Mahdavi Mazdeh et al. (2011 c) solved the problem of minimizing weighted sum of flow times with delivery cost analytically usinga branch and bound approach. Another research related to Mahdavi Mazdeh et al. (2007 b) presented a branch and bound to minimize flow time and delivery cost simultaneously. Hamidinia et al. (2012) studied a single batch delivery system to minimize total tardiness and earliness simultaneously. They proposed a mathematical model for describing the problem and presented a genetic algorithm for solving it.

Deterioration is one of the other assumptions which approach the scheduling problems in the real world industries and means that if processing a job becomes delayed, the processing time will be increased by a special function.

Mosheiov (1994) is known as the pioneer of presenting the deterioration in single machine problems. He investigated several functions involving minimizing 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 by considering a simple linear deterioration model.

Wang et al. (2009) studied the single machine scheduling problem with learning effect and deteriorating jobs simultaneously so as to minimize the total weighted completion time and the maximum lateness. The other researches in the field of the application of deterioration in single machine problems could be mentioned as Wang and Wang (2010), Husang et al. (2010) and Cheng and Ji (2010).

In the classic models of single machine, it was assumed that all jobs are ready for process before the starting of scheduling. But in real industries especially in supply chains, the orders enter to the system periodically. Hence, for any job a predefined release date can be considered that determines the time which that job is ready to be processed.

Although the assumption of release date has been studied in several single machine problems, there are few papers that consider it in batching problem.

Lu et al. (2010) considered the problem of single machine parallel batch scheduling with release date and rejection. The objective function was minimizing the makespan and total cost of rejected jobs. A polynomial run time method and an approximation algorithm were presented for solving the problem.

Li and Yuan (2006) studied the single machine parallel batching with release date and three objective functions including minimizing makespan, minimizing machine occupation time and minimizing stocking cost. The problem was solved by an time dynamic programming algorithm.

Rakrouki et al. (2012) extended a new Mehta heuristic based on genetic local search and recovering beam search for solving the problem of minimizing the total completion time.

One of the closest researches to our study is related to Karimi-Nasab et al. (2012) in which the problem of single machine batch scheduling with deterioration and precedence constraints was studied. The objective function was minimizing the total tardiness and maximizing the job values in makespan and a simulated annealing was used for finding near optimal solutions ap. The main difference of our research here is using the clustering algorithms with SA as two hybrids in order to improve the efficiency of objective function. The other difference is related to the release date assumption that is considered in this study.

In this paper, we consider the issue of serial batch scheduling problem with release date and deterioration, where the objective is to minimize the total tardiness and delivery costs and maximize the job values simultaneously, which, based on our knowledge is not considered in the literature.

MahdaviMazdeh et al. (2011 a) mentioned that the problems that combine a scheduling function with delivery costs are rather complex and solving them by simple optimization methods is not commercial. On the other hand, Lawler (1997) showed that problem of single machine scheduling with total tardiness function is strongly NP-hard. Therefore, the present problem that considers the objectives of total tardiness, delivery costs and job values simultaneously is NP-Hard too.

Two hybrid algorithms, based on simulated annealing and clustering algorithms, are offered for finding near optimal solutions. In order to check the verification of SA, some data sets of problem are solved optimally, using Lingo software, and the results are compared with each other. Some sensitivity analyses are also done based on important factors of the problem. The remaining parts of this paper are as follows: in section 2 the proposed model is presented and the variables and parameters are introduced. In section 3 the solution

International Journal of Economics, Finance and Management
©2012. All rights reserved.

http://www.ejournalofbusiness.org

approach is offered and in section 4 the computational studies are presented.

2. PROBLEM FORMULATION

Consider that there is a machine that processes n jobs with no preemption and each job is available for processing based on a predefined release date. The completed jobs can be delivered to the customer immediately after completion or be awaited next jobs to be delivered as a batch. According to the number of jobs, N batches are considered and the jobs are assigned to the batches. Clearly, any batch that does not have any jobs will be omitted and then the sequence of batches is determined so that the proposed objective function is minimized. As mentioned, the serial batching assumption is considered in this paper, therefore, the processing time of each batch is calculated as sum of all the processing jobs located on it. The due date and release date of each batch is also the maximum due date and release date of jobs which are assigned to it respectively.

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

N	Number of jobs ready to be scheduled
N_b	Number of batches containing at least one job
p_i	The normal processing time of job i
d_i	The due date of i-th job
r_i	The release date of i-th job
$p_{bat}(k)$	The processing time of batch k
$D_{bat}(k)$	The due date of batch k
$R_{bat}(k)$	The release date of batch k
n_k	Number of jobs assigned to batch k
t_j	The tardiness of batch where scheduled on jth position
c_j	The completion time of job where scheduled on jth position
~	The rate of deterioration
	The decreasing rate of job value
s_k	The shipping cost of batch k
$y_{ik} = \begin{cases} 1 \\ 0 \end{cases}$	If job i is assigned to batch k Otherwise
$x_{kj} = \begin{cases} 1 \\ 0 \end{cases}$	If batch k is located on j-th position in the sequence Otherwise
$a_k = \begin{cases} 1 \\ 0 \end{cases}$	If there is at least one job assigned to batch k Otherwise

And the proposed model is presented as follows:

$\min \ z = \sum_k s_k . a_k + \sum_k t_k - \sum_j e^{-\beta . c_j}$	(1)
St:	
$c_j = \sum_k x_{kj} .(p_{bat}(k). j^{\sim}) + \max\{ \sum_k x_{kj} .R_{bat}(k), c_{j-1} \}$	j=1,2,…,N_b (2)
$c_0 = 0$	(3)
$p_{bat}(k) = \sum_i y_{ik} . p_i$	K=1,2,…,N_b (4)
$R_{bat}(k) = \max_i \sum_i y_{ik} .r_i$	(5)
$n_k = \sum_i y_{ik}$	K=1,2,…,N_b (6)
$\frac{n_k}{n_k + 1} \le a_k \le (\frac{n_k}{n_k + 1}).(n_k + 1)$	K=1,2,…,N_b (7)
$t_j = \max \left\{ 0, c_j - \sum_k D_{bat}(k). x_{kj} \right\}$	j=1,2,…,N_b (8)

©2012. All rights reserved.

$D_{bat}(k) = \min \sum_i y_{ik} . d_i$	$y_{ik} \neq 0$, K=1,2,…,N_b (9)
$\sum_j x_{kj} = 1$	k=1,2,…,N_b (10)
$\sum_k x_{kj} = 1$	j=1,2,…,N_b (11)
$\sum_k y_{ik} = 1$	i=1,2,…,N_b (12)
$x_{kj} = 0 \ or \ 1$	(13)
$y_{ik} = 0 \ or \ 1$	(14)

Equation (1) introduces the objective function where the first term corresponds to the delivery cost, the second term corresponds to the sum of tardiness and the third term is related to the sum of job values. Constraint (2) states that how the completion time of each batch is calculated. Constraint (3) mentions that the machine is available at time zero. Constraint (4) mentions that the release date of each batch is equal to the maximum release date of jobs assigned to it. Constraint (5) declares that the processing time of each batch is equal to the sum of processing times of jobs assigned to it. Constraint (6) determines how many jobs are located on each batch. Constraint (7) states that whether batch k is empty or not. Constraint (8) clarifies that the tardiness of each batch is equal to the gap between the time that the processing of that batch has been completed and its due date. Clearly, the value of tardiness must be positive. Equation (9) shows how the due date of each batch is calculated. Constraint (10) states that each batch is processed only once at each sequence and constraint (11) determines that each position can be assigned to just one batch. Constraint (12) mentions that each job can be assigned to exactly one batch to be processed and constraint (13), (14) shows that x and y are binary variables.

3. SOLUTION APPROACH

As mentioned, the considered problem is NP-hard; hence, it is not reasonable to use ordinary optimization methods. In this case, two hybrid algorithms, based on simulated annealing meta- heuristic (SA) and some clustering algorithms, in order to reach near Optimal solutions in suitable run times, are presented. The reason for using SA is that it can be considered as a Marcov chain (Van Laarhoven and Aarts 1988), so its sensitivity to the initial solution is much less than the other meta-heuristics and has a great ability to avoid getting in local optimum solutions.

3.1 Simulated Annealing

Simulated annealing (SA), which has been applied widely to solve many combinatorial optimization problems, is a class of optimization Meta heuristics and performs a stochastic neighborhood search through the solution space.

In this paper, the algorithm starts from an initial temperature termed as T_0, in which two random strings are generated as below.

2	2	1	3	4

1	3	4	2

Fig 1: The coding scheme of proposed SA

The first string corresponds to assigning the jobs to batches and the other is related to finding a suitable scheduling of generated batches.

Based on this fact, consider that there are five available jobs for processing on the machine. The first string shows that the first and the second jobs are assigned to batch 2, the third job is located on batch 1 and the fourth and fifth jobs are assigned to batches 3 and 4 respectively. Since number 5 is not mentioned in this string, therefore the fifth batch has remained empty and will be removed. Afterwards, in second string, a random schedule of all the batches that contain at least one job is generated.

This procedure continues to the point that the equilibrium condition for this temperature occurs. In this problem, the equilibrium condition is met when the gaps between proposed objective functions in consecutive iterations in a certain temperature are as less as possible. This condition can be demonstrated by:

$$\Delta Z = \sum_{i=k-j}^{k} Z_{[i+1]} - Z_{[i]} <= £ \qquad (14)$$

Where Z_{i+1} is the value of objective function in i+1-th iteration of algorithm, j is the number of objectives that are considered to calculate the total gap and £ is a very small number.

International Journal of Economics, Finance and Management
©2012. All rights reserved.

http://www.ejournalofbusiness.org

By reaching the equilibrium in the temperature, the temperature decreases and the procedure starts from the lower temperature and continues until reaching the next equilibrium.

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.

3.2 Clustering

Clustering is the assignment of a set of observations to subsets (called clusters) so that the observations in the same cluster are similar in some sense and the similarity of generated clusters is little. Clustering is a method of unsupervised learning, and a common technique for statistical data analysis which has been applied in many fields, including machine learning, data mining, pattern recognition, image analysis, information retrieval, and bioinformatics. Clustering contains several algorithms and methods. For more details about clustering see (Jain et al. 1999 and Kittaneh 2012)

In this paper the Hierarchical clustering and K-means clustering are considered in order to assign the jobs to a batch based on some similarities.

a. Hierarchical Clustering (HC)

Hierarchical Clustering, which is presented by Johnson (1967), contains agglomerative and divisive schemes. In agglomerative clustering, at first, N clusters are considered, in which N is equal to the number of observations. Each one of the N clusters is assigned to only one batch. . In each step, two items which have more similarities are merged with each other and a unit batch is generated. This procedure continues until the number of clusters is equal to the predefined number.

On the other hand, divisive scheme starts the assignment procedure by putting all the items in one batch. In each step, an item that has the least similarity with others is removed from the batch and assigned to another batch. This procedure continues until the number of clusters becomes equal to the desired number.

The identification of similarities between the items can be done in different ways, which is what distinguishes single linkage from complete linkage and average linkage clustering.

In single linkage clustering, also called the connectedness or minimum method, the distance between two clusters is considered equal to the shortest distance from any member of one cluster to any member of the other cluster. In complete linkage clustering, also called the diameter or maximum method, the distance between two clusters is considered equal to the greatest distance from any member of one cluster to any member of the other cluster.

In average linkage clustering, the distance between two clusters is considered equal to the average distance from any member of one cluster to any member of the other cluster.

Generally, divisive method is a more complex method in comparison to the agglomerative scheme and requires more run times to assign the items into separate clusters.

b. K-means clustering (KC)

The basic concept of K-means clustering (MacQueen, 1967) is very similar to the Hierarchical clustering. However, the K-means method considers the assignments as an optimization problem. In this regard, first a number is defined as the desired number of clusters. Then each batch is occupied with a random item. These items are correspondences to the center of their clusters and are termed as K-centroids. The next step is related to assigning the remaining items in the clusters according to the maximum similarity between the remaining items and the centeroids. By adding a new item in each cluster, the centeriod values are recalculated and this procedure continues until the centroid values no longer change.

Although it could be proved that the procedure will always terminate, the k-means algorithm does not necessarily find the most optimal configuration, which is corresponding to the global objective function which is minimum. The algorithm is also significantly sensitive to the initially randomly selected cluster centers. The k-means algorithm can be run multiple times to reduce this effect.

The presented clustering algorithms are efficient for assigning the items to batches, but have some disadvantages. One important disadvantage of the presented clustering algorithms is the incapability of them to calculate the optimal number of batches. Consequently, the batch numbers must be given to them as an input data. In order to overcome to this problem these methods are used by SA as hybrid methods which are presented in details in following sections.

3.3 Description of Hybrid Algorithms

In this section, the performances of proposed hybrids are described. In both hybrids, the procedure begins with the proposed SA to determine the desired value for number of batches. Afterwards, one of the clustering algorithms is used in order to assign the jobs to suitable batches. Then the generated batches are scheduled by the second string presented in SA section. In computational study section, it is shown that using the

©2012. All rights reserved.

clustering methods cause immense improvement via the objective function.

a. SA-HC hybrid

First, the SA algorithm is run and the numbers of batches are determined. Then the jobs are assigned to batches using Hierarchical clustering. The linkage method is considered as the average between items and the distance measure is defined as squared Euclidean. As mentioned, the similarity of items is calculated based on the processing times and due dates of them. It means that the jobs that have close values of processing time and due date are joined to generate a unit batch. Afterwards, the generated batches are entered into the SA again and their schedule is specified. The performance of proposed hybrid is presented below in more detail:

```
1. Start with the SA and determine the number of batches. ( N_final)
2. Consider N virtual batches. (N is equal to the number of jobs)
3. Assign each job to a batch randomly so that each batch occupies only one batch.
4. while N < >  N_final DO
5.       Merge two batches that has the closest similarities
6.       N=N-1
7. loop
8. Enter the generated batches in the SA by a random schedule. (state k)
9. T=T0
10. While T<TF DO
11.       Do while the equilibrium condition has not occurred
12.           Generate a neighborhood ( state j)
13.           Z=f(j)-f(k)
14.           If z <0 then k=j
15.           Else if random (0,1) < exp(-Δz/T)
16.           K=j
17.     loop
18.   T= T.(cooling rate)
19.   Loop
20. End
```

Fig 2: The pseudo-code of SA-HC algorithm

b. SA-KC Hybrid

The beginning of SA-KC hybrid is similar to SA-HC hybrid; SA algorithm is used at first to determine the number of batches. Then the jobs, using K-means clustering, are assigned to batches. The similarity of items is calculated based on their processing times and due dates. It means that the jobs that have near values of processing time and due date are joined to generate a unit batch.

Afterwards, the generated batches are entered into the SA again and their schedule is specified. The performance of proposed hybrid is presented below in more detail:

```
1. Start with the SA and determine the number of batches. (N_final)
2. Assign N jobs to each batch randomly as the center of them. (center(i))
3. while the center^new(j) –center (j) < >0 do
4.       assign the remaining jobs to the generated batches based on the maximum similarity
         Recalculate the center of each batch (center^new(j))
5. loop
6. Enter the generated batches in the SA by a random schedule. (state k)
7. T=T0
8. While T<TF DO
9.       Do while the equilibrium condition has not occurred
10.           Generate a neighborhood ( state j)
11.           Z=f(j)-f(k)
```

©2012. All rights reserved.


```
        If z <0 then k=j
        Else if random (0,1) < exp(-Δz/T)
        K=j
    loop
  T= T.(cooling rate)
  Loop
End
```

Fig 3: The pseudo-code of SA-KC algorithm

3.4 Calibrating the SA Parameters

In order to calibrate the proposed SA, a Taguchi approach is presented in which attempts to identify controllable factors (control factors) which minimize the effect of the noise factors, have been made. During the experimentation, the noise factors are manipulated to force variability to occur and then to find the optimal control factor settings that make the process or product robust or resistant to variations from the noise factors.

In this paper, the S/N ratio which is considered to be nominal, is the best of the kind and is calculated by the following equation:

$$\frac{S}{N}\ ratio = 10\log_{10}(objective\ function)^2 \qquad (15)$$

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

Table 1: The Taguchi experiment inputs

Factor	**symbol**	**levels**	**Type**
Initial temperature	A	3	A(1)=500 A(2)=1500 A(3)=3000
Number of objectives that are considered to calculate the total gap in each temperature	B	3	B(1)=3 B(2)=4 B(3)=5

The associated degree of freedom for these two factors is equal to 8. According to Taguchi standard table of orthogonal array, the L_9 which fulfils all the minimum necessary requirements should be selected.

In order to analyze the Taguchi outputs, three important measures are considered as the S/N ratio (as robust measure), the average responses for each combination of control factors and the variability in the responses due to the noise (standard deviation).

The results are depicted in below figures.

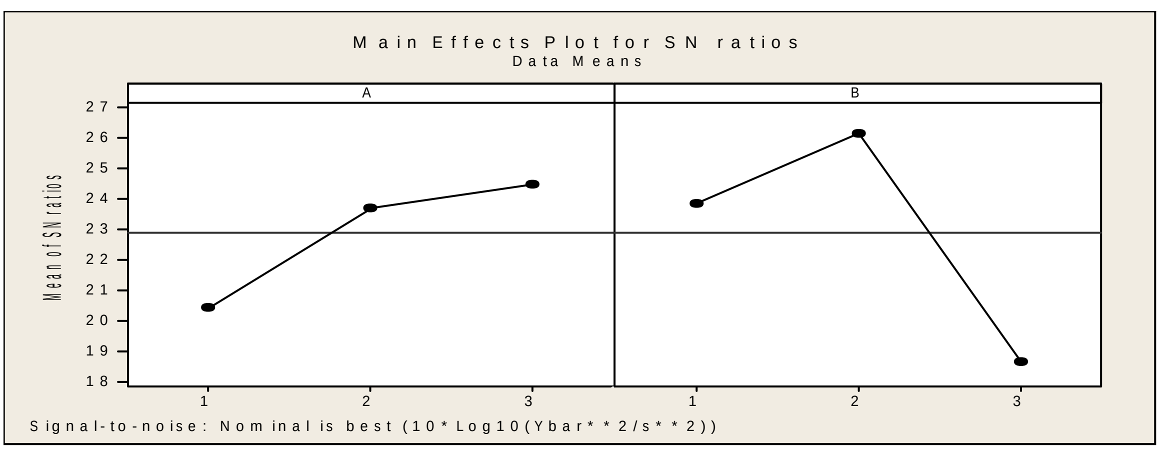

Fig 4: the results for responses based on S/N ratio

©2012. All rights reserved.

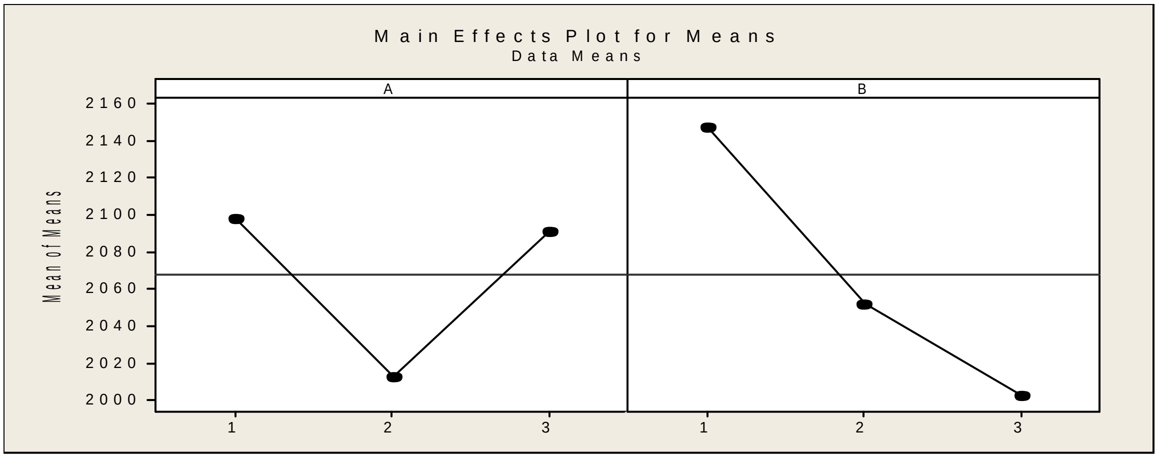

Fig 5: the results for responses based on means

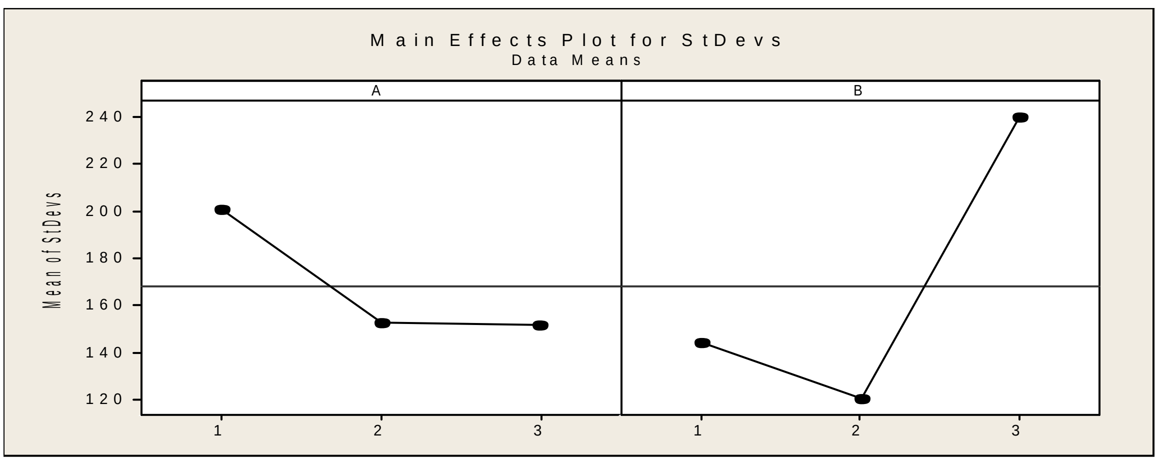

Fig 6: the results for responses based on standard deviations

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. 4 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(3) and B(2) are selected.

Fig. 5 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 (3) are selected.

Finally, Fig. 6 shows the variability in the responses due to the noise which is desired to be minimum, hence A(3) and B(2) are selected. Based on mentioned measures, the most efficient combination of the proposed factors are A(3) and B(2) which better satisfies the response values.

4. COMPUTATIONAL EXPERIMENT

All the instances for this problem were coded by Visual Basic 6 and were run on personal computer with CORE I 7 processor and 4 GB of RAM. The required data was 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 α is considered equal to 0.2 manually.

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

To make the sensitivity analysis, some important factors, including problem dimension (number of

©2012. All rights reserved.

considered batches) and rate of deterioration, have been considered.

4.1 Sensitivity Analysis Based On Problem Dimension

In order to implement this analysis first, some instances with small dimensions are considered and the results are compared to equivalent Lingo results. But Lingo is incapable of solving the problem for greater than 5 jobs by virtue of immense complexity of the problem. Rate of deterioration and rate of job values are also considered as 0.2 and 0.002 respectively in this section.
Table 2 depicts the comparison of the results of SA, hybrids and Lingo based on the problem dimension.

Table 2: Sensitivity analysis based on small problem dimensions

Number of tasks	Number of batches	Simple SA			SA-KC			SA-HC			Lingo	
		VOF	Run time (sec)	%GAP	VOF	Run time (sec)	%GAP	VOF	Run time	%GAP	VOF	Run time (sec)
4	1	232.34	3	0	232.34	-	0	232.34	-	0	232.34	90
5	2	371.56	12	2.37	371.56	1	2.37	371.56	1	2.37	362.93	348
6	2	420.97	17	16.20	362.26	3	0	365.20	3	0.08	362.26	1643
7	4	877.96	63	-	801.78	25	-	761.76	25	-	-	-
8	5	1284.6	57	-	1284.63	19	-	1233.75	16	-	-	-
9	4	1533.74	58	-	1438.94	9	-	1438.94	9	-	-	-
10	4	2096.07	52	-	1485.77	10	-	1825.22	10	-	-	-
15	8	5930.45	60	-	4952.93	13	-	5010.34	16	-	-	-

The second column indicates the number of batches and the third to fifth columns show the performance of single SA, including VOF (Value of

Objective Function), run time in seconds and the percentage of gap between them and global optimum.

For each instance, the SA was run 5 times and the best obtained solution was considered as the best VOF. The other columns also present the results of proposed hybrids and global optimums by Lingo.Based on the results from Table 2, using both clustering algorithms leads to decrease the required run time for solving the problem. On the other hand, the value of objective function is improved drastically. As mentioned in the introduction section, the proposed model is NP-hard and solving it optimally is not commercial in reasonable run time. In this regards, the Lingo solver is incapable to solve the problem with increasing the dimension. Therefore, the comparison of gaps between the results and optimum value objective function is not possible for larger scales.

The results gap between SA and global optimum is calculated by:

$$\%gap = \frac{VOF(SA) - VOF(global)}{VOF(global)} * 100 \qquad (16)$$

According to the results of table 2, by using the clustering methods not only the value of objective function is improved, but also the run time is decreased. For small dimensions of problem, both hybrids perform very efficiently and offer the solutions with low error in a reasonable time.

By increasing the number of jobs, does not seem that the hybrid of SA-KC performs faster, but in field of objective function, generally, there is no dominance.

The comparison of SA and proposed hybrids for solving the problem in larger scales are presented in table below.

International Journal of Economics, Finance and Management
©2012. All rights reserved.

http://www.ejournalofbusiness.org

Table 3: sensitivity analysis based on medium and large problem dimensions

Number of tasks	Number of batches	Simple SA		SA-KC		SA-HC	
		VOF	Run time (sec)	VOF	Run time (sec)	VOF	Run time
25	16	18966	66	14075	16	11864	22
30	14	20542	78	16706	16	14723	19
40	23	50101	106	41026	23	43033	24
50	30	73794	138	60552	25	66015	30

Based on the results from table 3, the efficiency of single SA declines by increasing the dimension of problem, but the hybrids offer better solutions in lower run time. For larger scales of the problem, it can also be mentioned that the SA-KC solves the instances faster than other algorithms, though it is not possible to suggest any comparison for the value of objective function for the hybrid algorithms.

4.2 Sensitivity analysis based on the rate of deterioration

The effect of deterioration is evaluated by dimension of 10 jobs and value of 0.002 for the rate of job values. Table 4 illustrates the results of SA for several rates of deterioration.

Table 4: sensitivity analysis based on rate of deterioration for N=10

Rate of deterioration	Number of batches	Simple SA		SA-KC		SA-HC	
		VOF	Run time(sec)	VOF	Run time(sec)	VOF	Run time
0.1	5	1818	33	1220	10	1208	10
0.2	5	1849	28	1554	10	1517	10
0.3	5	2322	33	1991	10	1818	10
0.4	5	2968	31	2495	10	2437	9
0.5	5	3153	32	3036	10	2979	10
0.6	5	3928	30	3698	8	3363	8
0.7	5	4489	31	4473	10	4283	8
0.8	5	5797	30	5180	10	5071	7
0.9	5	6699	31	5180	8	5748	8

According to table 4, the rate of deterioration is highly effective on the performance of SA and hybrids and clearly alters their final solutions.

On the other hand, does not seem to be any relationship between more rates of deterioration and the run times.

4.3 Sensitivity Analysis Based On The Rate Of Job Values

The sensitivity analysis also is conducted for the various values of reduction rate of job values and the results are shown in table 4. In this regard, rate of deterioration is considered equal to 0.2 and the number of jobs is 10.

Table5: sensitivity analysis based reduction rate of job values

Rate of job values	Number of batches	Simple SA		SA-KC		SA-HC	
		VOF	Run time (sec)	VOF	Run time (sec)	VOF	Run time
0.001	6	2413	32	1992	12	1999	10
0.002	6	2596	29	2065	11	1965	10
0.003	6	3024	30	2712	12	2896	11

©2012. All rights reserved.

0.004	6	2732	30	2478	12	1902	8
0.005	6	1965	35	1782	11	1886	10
0.006	6	1968	28	1912	9	1802	9
0.007	6	2002	28	1996	9	1996	8
0.008	6	1952	36	1756	8	1863	9
0.009	6	1978	39	1652	10	1600	8

Based on depicted plots it can be concluded that the proposed hybrids perform much better than simple SA for all the values of reduction rate and generally, there are no relation between this rate and the value of objective function.

4.4 The Comparison Of The Two Proposed Hybrid

In all of the previous sections, the results of both proposed hybrids are much better than single SA, especially in large scale problems. In this section, it is desired to compare the results of hybrid algorithms in a fair way. In order to do so, their performance is calculated in equal time intervals. Each algorithm is run 30 times and the best, average and the worst solutions are recorded. Table 6 summarizes this information.

Table 6: the comparison of hybrids performances

Run time (sec)	Alg.	Best	Avg.	Worst	SD	SD/Avg.
2.5	SA-HC	5515	6246	6690	445.61	0.07
	SA-KC	5447	6240	6657	484.65	0.07
5	SA-HC	4959	5237	5737	294.17	0.05
	SA-KC	5626	5651	5933	186.31	0.03
7.5	SA-HC	4959	5030	5189	104.06	0.02
	SA-KC	5177	5467	5660	188.60	0.03
10	SA-HC	4959	4974.2	5035	33.98	0.0068
	SA-KC	5177	5178	5180	1.64	0.0003

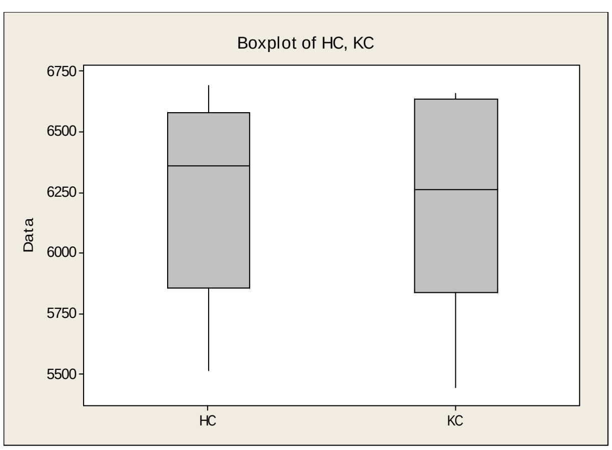

Fig 7: performances of hybrids in 2.5 seconds

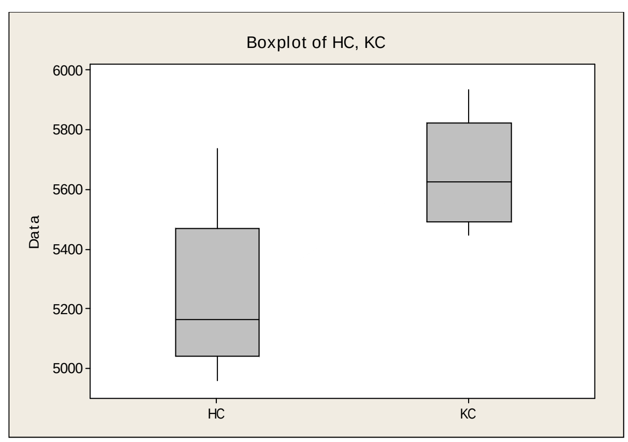

Fig 8: performances of hybrids in 5 seconds

©2012. All rights reserved.

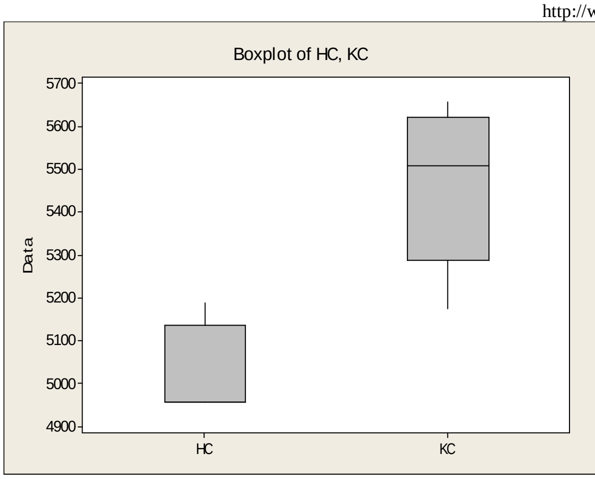

Fig 9: performances of hybrids in 7.5 seconds

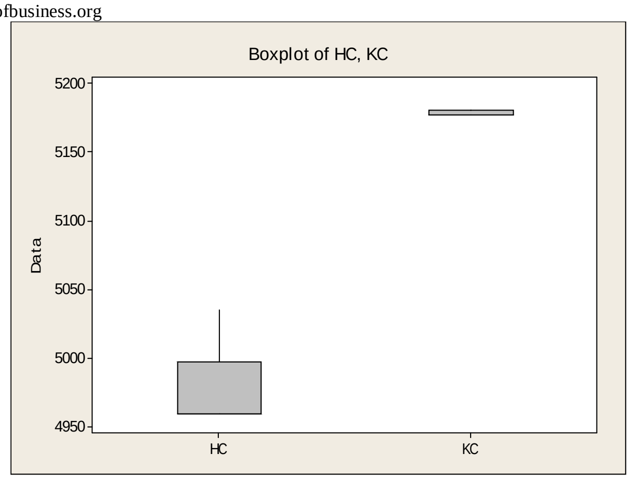

Fig 10: performances of hybrids in 10 seconds

Based on the results of table 6 and depicted plots, the performance of HC is better than KC in the field of objective function value, except in the case of 2.5 second run time; in this time interval KC performs better. On the other hand, the standard deviations of both algorithms decrease as time passes.

5. CONCLUSION AND FUTURE RESEARCH

This paper studied the problem of single machine batch scheduling with release date and deterioration. The objective function was minimizing the total tardiness and delivery costs and maximizing the job values simultaneously; in which a mathematical model was presented. In order to solve the proposed model, two hybrid algorithms were proposed and their performances was compared to global optimum for small dimensions of problem which were generated using Lingo software.

Afterwards, a number of sensitivity analyses were implemented based on effective factors of the problem, including problem dimension, rate of deterioration and rate of job values in makespan.

For future researches, the solution method can be considered as some exact algorithms like branch and bound, branch and cut and Lagrangian relaxation in order to reach the global optimum even for medium and larger scales of jobs. Furthermore, the problem could be considered in more nearest environments in real industry like job shop or parallel machine instead of single machine.

REFERENCES

[1] Albers, S. and Brucker, P. (1993). The complexity of one-machine batching problems. Discrete Applied Mathematics 46, 87–107.

[2] Baptiste, P. (2000). Batching identical jobs. Mathematical Methods of Operations Research 52, 355–367.

[3] Cheng, TCE. and Kovalyov, MY. (2000). Single machine batch scheduling with sequential job processing. IIE Transactions 33 (5), 413-420, DOI: 10.1023/A:1011005314354.

[4] Cheng, TCE. Janiak, A. and Kovalyov, M. (2001). Single machine batch scheduling with resource dependent setup and processing times. European Journal of Operational Research 135 (1), 177-183.

[5] Coffman, E.G. Yannakakis, M. Magazine, MJ. and Santos, C. (1990). Batch sizing and sequencing on a single machine. Annals of Operations Research 26, 135–147.

[6] Hamidinia, A. Khahabimamaghani, S. Mahdavi Mazdeh, M and Jafari, M. (2012) .A genetic algorithm for minimizing total tardiness/earliness of weighted jobs in a batched delivery system. Computers and industrial engineering 62 (1), 29-38.

[7] Husang, H. Wang, JB. and Wang, XR. (2010). A generalization for single-machine scheduling with deteriorating jobs to minimize earliness penalties. Int J Adv Manuf Technol 47, 1225–1230.

[8] Ishii, H. Xuesong, Li. and Masuda, T. (2010). Single machine batch scheduling problem with fuzzy batch size. 40th International Conference on Computers and Industrial Engineering. Awaji, 1-5

©2012. All rights reserved.

[9] Jain, A.K. Murty, M.N. and Flynn, P.J. (1999) Data clustering: A review. ACM Computing Surveys 31, 3.

[10] Johnson, S. (1967). Hierarchical Clustering Schemes. Psychometrika, 2, 241-254.

[11] Karimi-Nasab M, Haddad H.R, Ghanbari P. A simulated annealing for the Single machine batch scheduling with deterioration and precedence constraints. Asian Journal of industrial engineering 4(2012) 1-16.

[12] Kittaneh R, Abdullah S, Abuhamdah A. Iterative simulated annealing form medical clustering problems. Trends in applied sciences research 7 (2): 103, 117, 2012.

[13] Lawler EL. " A pseudo polynomial' algorithm for sequencing jobs to minimize total tardiness". Annals of Discrete Mathematics 1(1997)331–342.

[14] Li W, Yuan J. Single machine parallel batch scheduling problem with release dates and three hierarchical criteria to minimize makespan, machine occupation time and stocking cost. International Journal of Production Economics 102 (2006) 143-148.

[15] Lu S, Feng H, Li X. Minimizing the makespan on a single parallel batching machine. Theoretical Computer Science 411 (2010) 1140-1145.

[16] MacQueen, J. (1967). Some Methods for classification and Analysis of Multivariate observations, Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability. Berkeley, University of California Press, 281-297.

[17] Mahdavi Mazdeh, M. Hamidinia, A. and Karamouzian, A. (2011). A mathematical model for weighted tardy jobs scheduling problem with a batched delivery system. International Journal of Industrial Engineering Computations 2.

[18] Mahdavi Mazdeh, M. Sarhadi, M. and Hindi, KS. (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.

[19] Mahdavi Mazdeh, M. Shashaani. S. Ashouri, A. and Hindi, KS. (2011), Single-machine batch scheduling minimizing weighted flow times and delivery costs. Applied Mathematical Modelling 35, 563–570.

[20] Mosheiov, G. (1994). Scheduling jobs under simple linear deterioration, Computers and Operations Research 21, 653–659.

[21] Ng, CT. Cheng, TCE. and Kovalyov, M. (2004) Single machine batch scheduling with jointly compressible setup and processing times. European Journal of Operational Research 153 (1), 211-219.

[22] Ng, CT. Cheng, TCE. and 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.

[23] Nong, Q. Ng, CT. and 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.

[24] Rakrouki M.A, Ladhari T, T'kindt V. Coupling Genetic Local Search and Recovering Beam Search algorithms for minimizing the total completion time in the single machine scheduling problem subject to release dates. Computers & Operations Research 39 (2012) 1257-1264.

[25] Tian, J. Fu, R. and Yuan, J. (2007). On-line scheduling with delivery time on a single batch machine. Theoretical Computer Science 374, 49–57.

[26] Van Laarhoven, P.J.M. and Aarts, E.H. (1988). Simulated Annealing: Theory and Applications, Kluwer Academic Publishers, Dordrecht.

[27] Wang, D. and Wang, JB. (2010). Single-machine scheduling with simple linear deterioration to minimize earliness penalties, Int J Adv Manuf Technol 46, 285–290, DOI 10.1007/s00170-009-2086-8.

[28] Wang, JB. Huang, X. Wang, X. Yin, N. and Wang L. (2009). Learning effect and deteriorating jobs in the single machine scheduling problems, Applied Mathematical Modeling 33, 3848–3853.

[29] Cheng, TCE. and Ji M., (2010). Batch scheduling of simple linear deteriorating jobs on a single machine to minimize makespan. European Journal of Operational Research 202, 90–98.