International Journal of Industrial Eng. & Production Research (2008) pp. 47-56

Volume 19, Number 4, 2008

International Journal of Industrial Engineering & Production Research

Journal Website: http://ijieen.iust.ac.ir/

A Single Machine Sequencing Problem with Idle Insert: Simulated Annealing and Branch-and-Bound Methods

R. Tavakkoli-Moghaddam* & M. Vasei

R. Tavakkoli-Moghaddam, *The Department of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran*
M. Vasei, *The Department of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran*

Keywords

Single machine sequencing, Earliness/ tardiness, Idle insert, Simulated annealing, Branch-and-bound methods

ABSTRACT

In this paper, a single machine sequencing problem is considered in order to find the sequence of jobs minimizing the sum of the maximum earliness and tardiness with idle times (n/1/I/ET$_{max}$). Due to the time complexity function, this sequencing problem belongs to a class of NP-hard ones. Thus, a special design of a simulated annealing (SA) method is applied to solve such a hard problem. To compare the associated results, a branch-and-bound (B&B) method is designed and the upper/lower limits are also introduced in this method. To show the effectiveness of these methods, a number of different types of problems are generated and then solved. Based on the results of the test problems, the proposed SA has a small error, and computational time for achieving the best result is very small.

© 2008 IUST Publication, All rights reserved. Vol. 19, No. 4

1. Introduction

Various objective functions exist in the literature survey for a single machine scheduling problem. Most of these objective functions are introduced in terms of earliness and tardiness of jobs. Earliest due date (EDD) order is used for minimizing the maximum difference between the due date and the completion time of jobs (L_{max}) as well as the maximum tardiness (T_{max}) [1 and 2]. In most cases, the sum or mean of tardiness of all jobs is considered as a criterion to determine the job sequencing. The mean tardiness is represented by $\overline{T}$ and in overall, the traditional optimization methods, such as B&B and dynamic programming (DP), are used for minimizing this criterion [1 and 3].

These methods are generally inefficient for solving large-scale problems. Emmons [4] introduced essential conditions to find an optimal solution for $\overline{T}$ in a single machine problem, after proving some theorems. Sen and Borah [5] reduced the set of feasible solutions in order to find the optimal solution using Emmons' theorems. They obtained the optimal solution for up to 30-job problems by the B&B method. Horseback and Ressell [6] introduced a heuristic method for minimizing $\overline{T}$ according to Emmons' theorems. This heuristic method has been considered as a base for further research. Panwalker, *et al.*, [7] proposed a heuristic algorithm, Islam and Eksiglu [8] proposed a method based on tabu search (TS) in order to minimize $\overline{T}$.

Most researchers have been interested in a multi-objective function for sequencing and scheduling problems to adapt and satisfy managers' requirements. One of the most important objectives is to minimize the (weighted) sum of the earliness and tardiness of jobs. This matter is conformity to just-in-time (JIT) systems [9 and 10]. Tardiness and earliness causes penalties in losing customers and increasing inventory cost, respectively. Thus, none of these penalties is desirable. Most researchers are interested in various forms and with various assumptions for the due dates of jobs, allowing idle insert, and weighting of earliness and tardiness [11-14].

There are large values of earliness or tardiness for some jobs in results obtained from minimizing the sum of earliness and tardiness. Thus, this problem causes some difficulties in production systems. To identify this problem, consider a case that all jobs are done on

*Corresponding author: *R. Tavakkoli-Moghaddam*
E-mail: *tavakoli@ut.ac.ir*
Paper first received Nov. 10, 2006, and in revised form Feb. 02, 2008.

[Downloaded from ijiepr.iust.ac.ir on 2024-07-17]

machines exit from a firm as the batches built-up many parts. If all jobs of a batch could be produced on time but a job has tardiness, then other jobs of the batch must wait. Thus, their on time production is not an advantage. In such a situation, if the jobs are carried out earlier, they need some space and increase the inventory level. However, if there is earliness or tardiness, then their associated values should be almost the same for all jobs.

In other word, if a job in a batch has earliness, then other jobs of the batch will have earliness. Likewise, if a job has tardiness, then other jobs will have tardiness. Thus, the interval time between earliness and tardiness must be approximately zero. This aim is fulfilled by minimizing the sum of the maximum earliness and tardiness.

Amin-Nayeri and Moslehi [15] studied a single-machine sequencing problem to find an optimal sequence of jobs, in which the objective function is to minimize the maximum earliness and tardiness. In the above paper, some assumptions for the original and traditional model as well as the absence of idle insert for a job and a machine have been considered. Since ET_{max} (the sum of the maximum earliness and tardiness) is an irregular criterion, then it is possible to eliminate the assumption of unallowable idle insert and to define a new problem.

However, this paper introduces a new sequencing problem considering idle insert, namely $n/1/I/ET_{max}$, in which a search is carried out for finding the optimal sequence holding idle insert. If the objective function is to minimize the sum of the maximum earliness and tardiness with allowing idle insert, then the objective is to find the best value of idle insert in a known sequence for improving the objective function. Tavakkoli-Moghaddam, *et al.*, [16] proposed an optimal algorithm to obtain the best value of idle insert in the known sequence ($n/1/OI/ET_{max}$).

In the next section, we describe the symbols. The difference between $n/1/I/ET_{max}$ and $n/1//ET_{max}$ considering idle insert is presented in Section 3. Neighboring conditions, upper and lower bounds for a branch and bound method are introduced in Sections 4 and 5, respectively. In Section 6, the proposed B&B algorithm is presented. The proposed SA algorithm is introduced in Section 7. Computational results are reported in Section 8. Finally, Section 9 is a conclusion.

2. Symbols

To explain the lemmas and associated relationships, the general symbols are defined as follows. The number of jobs in a known sequence is n, in which the processing time and the due date of job i are represented by p_i and d_i, respectively. The completion time and the difference between the completion time and due date are represented by C_i and L_i, respectively. In a single machine sequencing, earliness (E_i) and tardiness (T_i) of job i, maximum earliness (E_{max}), maximum tardiness (T_{max}), and the sum of maximum earliness and tardiness (ET_{max}) in each sequence are obtained as follows:

$$E_i = \max\left(0, \bar{d}_i - C_i\right) \tag{1}$$

$$T_i = \max\left(0, C_i - d_i\right) \tag{2}$$

$$E_{\max} = \max_{1 \le i \le n}\left\{E_i\right\} \tag{3}$$

$$T_{\max} = \max_{1 \le i \le n}\left\{T_i\right\} \tag{4}$$

$$ET_{max} = E_{max} + T_{max} \tag{5}$$

Term id is the time of incremental idle insert in a sequence. The problem of an optimal sequence with the objective function ET_{max} considering idle insert is represented by $n/1/I/ET_{max}$.

3. $n/1/I/ET_{max}$ AND $n/1//ET_{max}$ With Idle Insert

In the problem $n/1//ET_{max}$, an optimal sequence of jobs is found by the proposed B&B algorithm [15], in which the objective function is to minimize ET_{max}. The concept, in which an optimal sequence of jobs with the absence of idle insert ($n/1//ET_{max}$) is found and then the best value of idle insert in the known optimal sequence for improving the objective function is obtained by the idle insert algorithm [16], which is different from $n/1/I/ET_{max}$. This subject is proved by a reversal example.

Reversal example. Jobs 1, 2, and 3 are considered with processing times 1, 6, and 2 and due dates 12, 5, and 4 respectively. By solving the above example with the absence of id, the optimal sequence 1-2-3 is obtained, in which the sum of the maximum earliness and maximum tardiness is equal to 6 ($n/1//ET_{max}$). By using the idle insert algorithm [16], the objective function reduces one unit and improves to 5, whereas sequence 2-1-3 with the objective function 4 is obtained by a complete enumeration of $n/1/I/ET_{max}$. In Table 1, all feasible sequences are given. As seen, the first sequence with the objective function value 6 has the best value of ET_{max} in a problem $n/1//ET_{max}$. In Table 2, the improvement values of the objective function for all feasible sequences are given. As seen, only in cases 1 and 4, the objective function can be improved by using the idle insert. In cases 1 and 4, the objective function can be improved 1 and 3 units, respectively.

As mentioned above, the best sequence for $n/1/I/ET_{max}$ is 2-1-3, in which the objective function value is equal to 4. Whereas, the best sequence for $n/1//ET_{max}$ is 1-2-3, in which the idle insert improves 1 unit and the associated objective function value is transformed into 4.

[Downloaded from ijiepr.iust.ac.ir on 2024-07-17]

Tab. 1. Calculation of maximum earliness and maximum tardiness for all feasible sequences

sequence	Maximum earliness	Maximum tardiness	Objective function value
1-2-3	3	3	6
1-3-2	9	4	13
2-3-1	5	5	10
2-1-3	3	4	7
3-1-2	11	4	15
3-2-1	11	5	16

Tab. 2. Comparison between "considering idle insert" and "without idle insert"

Sequence	Objective function value considering idle insert	Objective function value without idle insert	Best objective function value
1-2-3	5	6	5
1-3-2	13	13	13
2-3-1	10	10	10
2-1-3	4	7	4
3-1-2	15	15	15
3-2-1	16	16	16
Best value	4	6	4

4. Neighborhood Conditions

In this section, some lemmas, which are the basis of the proposed B&B algorithm, are presented. Lemmas 1 and 2 are used for determining the dominant set in the B&B method. Before determining the dominant set it is necessary to specify when the idle insert improves the objective function of the sequence. These four notes have been taken from [16].

Note 1. In a known sequence, if a job with E_{max} is positioned before a job with T_{max}, then the idle insert does not improve the objective function.

Note 2. In a known sequence, if a job with T_{max} is positioned before a job with E_{max} and idle insert is considered in the set B (set of jobs, which are positioned between the job with T_{max} and the job with E_{max}), then the idle insert may improve the objective function.

Note 3. If all jobs in a known sequence don't have earliness (they have tardiness greater than or equal zero), then the objective function may not be improved by the id.

Note 4. If the whole jobs in the known sequence have earliness, then the objective function is improved by considering the idle insert.

Lemma 1. In the problem $n/1/I/ET_{max}$, if the sequence is arranged in order of longest slack time (LST) and the last job in the sequence has tardiness, then positioning the idle insert in this sequence does not improve the objective function.

Lemma 2. In the problem $n/1/I/ET_{max}$, if the sequence is arranged in order of LST and the last job in sequence has earliness, then all jobs in every sequence will have earliness and the minimum earliness is obtained for the last job in the sequence with LST order.

By using lemmas 1 and 2, the following five principles can be used as a dominant set in the B&B method. Before introducing these five principles, a definition of the partial sequence σ and set σ' are presented. The associated positions of these two sets are shown in Figure 1. Elements of σ' are positioned before the elements of σ. The number of elements of each of these sets is equal or smaller than the total number of jobs, that is $\sigma+\sigma'$ is equal to *n*.

$\sigma=\{i \mid \text{order of job } i \text{ is specified}\}$
$\sigma'=\{i \mid \text{order of job } i \text{ is not specified}\}$

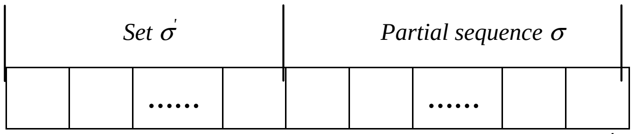

Fig. 1. Situations of the partial sequence σ and set σ'

Principle 1. According to lemma 1, if the last job in a sequence arranged by LST rule has tardiness, then the last job has maximum tardiness. Hence, if in the first sequence arrangement with LST rule, the last job has tardiness, then this job will have the maximum tardiness. Thus, the investigation of some sequences, in which the last job is the last job of LST rule and has tardiness, is avoided (see notes 1 and 3).

Principle 2. If all elements of partial sequence σ have tardiness and the last job of set σ' arranged with LST rule has tardiness, then the investigation of some sequences where all elements of partial sequence σ have tardiness and the last job of set σ' is the last job of LST arrangement and it also has tardiness are avoided (see note 1 and 3). The reason is that all jobs until the end of the sequence have tardiness and according to lemma 1, the remainder of jobs until the beginning of the sequence has a tardiness smaller than tardiness of the last job of LST rule for set σ',

Principle 3. If all elements of partial sequence σ have tardiness and the last job of set σ' arranged with LST rule has earliness, then maximum tardiness is positioned after maximum earliness and according to note 1, investigation of these sequences is avoided. Based on lemma 2, the remainder of elements of set σ' with any arrangement will not have an earliness greater than the earliness of the last job of LST rule of set σ' until the beginning of the sequence.

Principle 4. If all elements of partial sequence σ have earliness and/ or tardiness and the last job of set σ' arranged with LST rule has earliness and maximum earliness of partial sequence σ is smaller than earliness of the last job of LST rule of set σ', then according to lemma 2, any earliness smaller than earliness of the last

[Downloaded from ijiepr.iust.ac.ir on 2024-07-17]

job of LST rule of set σ' with any arrangement does not existed in set σ'. When maximum earliness of partial sequence σ is smaller than earliness of the last job of LST rule of set σ', this means that the job with maximum earliness is positioned before a job with maximum tardiness, and according to note 1, the investigation of these sequences is avoided.

Principle 5. If all elements of partial sequence σ have earliness and tardiness and the last job of set σ' arranged with LST rule has earliness and the maximum earliness of minimum slack time (MST) rule of set σ' is greater than the maximum earliness of partial sequence σ, then according to note 1, the minimum earliness of set σ', which is obtained by MST rule, is greater than the maximum earliness of partial sequence σ and this means that a job with maximum earliness is positioned before a job with maximum tardiness. Thus, the investigation of these sequences is avoided.

5. Upper And Lower Bounds For the Objective Function Value

In this section, lemmas 3 and 4 are presented to determine the proper upper and lower bounds, respectively.

Lemma 3. In the problem $n/1/I/ET_{max}$, the improved objective function for a solution obtained from the problem $n/1//ET_{max}$ by idle insert is an upper bound for $n/1/I/ET_{max}$.

Lemma 4. In the problem $n/1/I/ET_{max}$, the lower bound includes maximum earliness and maximum tardiness. Maximum earliness of a lower bound is $E_{max\sigma id}$ (i.e., maximum earliness of partial sequence σ after considering idle insert) and maximum tardiness of a lower bound is the maximum of $T_{max\sigma id}$ (i.e., maximum tardiness of partial sequence σ after considering *id*) as well as $T_{max\sigma pEDD}$ (i.e., maximum tardiness of the order EDD of σ').

6. Optimal Algorithm For Minimizing ET_{MAX} With Idle Insert

In this section, by combining the presented lemmas and a B&B method, an algorithm is proposed for minimizing ET_{max} considering idle insert as follows:

Stage 1. Computing the upper bound: In this stage, a feasible solution is presented as the upper bound. As shown in notes 1 and 3, if all jobs do not have earliness (they have tardiness greater than or equal zero), or a job with E_{max} is positioned before a job with T_{max}, then the idle insert does not improve the objective function. According to the mentioned notes, the explained dominant principles try to eliminate sequences that cause to create the two mentioned cases. Considering the property of dominant principles, it is possible that optimal solution is not searched. This subject occurs, when two conditions are done simultaneously. First, the solution of $n/1/I/ET_{max}$ is the same solution of $n/1//ET_{max}$ and the second, in the obtained optimal sequence of $n/1/I/ET_{max}$ and $n/1//ET_{max}$, which are the same, a job with E_{max} is positioned before a job with T_{max}. To avoid the elimination of the optimal solution with the B&B method in this case, the solution of $n/1//ET_{max}$ is considered as an upper bound for the problem $n/1/I/ET_{max}$. Thus, two properties of feasibility and correspondence with dominant principle are satisfied. After obtaining the optimal sequence of problem $n/1//ET_{max}$ by using the algorithm [15], the improvement value is obtained by using the idle insert algorithm [16]. Finally, the improved value of the objective function is considered as an upper bound.

Stage 2. Using the dominant principle: In this stage, the sequences satisfying the dominant principle are not searched. As proved in lemma 1, if set σ' is arranged with LST rule, in that the last job has tardiness, then by increasing the idle insert in the sequence, the objective function value is not improved. In other word's, if the last job in LST rule of set σ' has tardiness, then all jobs will not have earliness or a job with E_{max} is positioned before a job with T_{max}. This lemma is used as a dominant lemma. If a partial sequence σ is empty, then principle 1 is used. If a partial sequence σ is not empty, then principles 2, 3, 4, and 5 are utilized. Five principles resulted from lemmas 1 and 2 try to eliminate some sequences from the complete enumeration, which satisfy the dominant principle. Thus, the speed of the B&B method in achieving an optimal solution increases.

Stage 3. Computing the lower bound: Considering the dominate principle for each sequence, the maximum improvement is created in the objective function of partial sequence σ by using the idle insert algorithm [16]. Then, the lower bound is computed from Equation (6).

$$LB=max\{T_{max\sigma id},T_{max\sigma pEDD}\}+E_{max\sigma id} \qquad (6)$$

$T_{max\sigma id}$: maximum tardiness of partial sequence σ after considering *id*

$E_{max\sigma id}$: maximum earliness of partial sequence σ after considering *id*

$T_{max\sigma pEDD}$: maximum tardiness of order EDD for set σ

Stage 4. If the lower bound for each node is smaller than the upper bound, then the upper bound is converted into the lower bound for this node. Otherwise, if the lower bound for each node is equal or greater than the upper bound, then the algorithm desists from continuing the node and positioning the arranged last job in the partial sequence σ. According to the above four stages, the steps of the proposed algorithm are listed below:

Step 1. Obtain the optimal sequence of the problem $n/1//ET_{max}$ without considering the idle insert, using the

[Downloaded from ijiepr.iust.ac.ir on 2024-07-17]

optimal algorithm [15], and compute the objective function value.

Step 2. Create the maximum improvement in the obtained objective function value, using the optimal algorithm [16], and assign the objective function value to the upper bound.

Step 3. Assign the obtained optimal sequence in step 1 to set *U*.

Step 4. Divide the jobs of set *U* into two sections; set σ' and the partial sequence σ. *i* is the first job of partial sequence σ, and *j* is an element of set σ', which is created in the new branch. *Ji* is a partial sequence and *j* is positioned before *i*.

Step 5. If all branches are investigated, then the upper bound would be the final solution, and the problem solving is terminated. Thus, go to step 14. Otherwise, create a separation in branch *j*, i.e. job *j* is entered from non-arranged set σ' to the arranged jobs set namely partial sequence σ.

Step 6. If *j* is not the last job of order LST of set σ', then go to step 12. Otherwise, go to step 7.

Step 7. If the partial sequence σ is empty or all jobs have tardiness, then eliminate the branch *ij* and go to step 8. Otherwise, go to step 9.

Step 8. If set σ' is empty, then assign the lower bound to the upper bound and go to step 5. Otherwise, go to step 5, directly.

Step 9. If *j* does not have earliness, then go to step 12. Otherwise, go to step 10.

Step 10. If the maximum earliness of partial sequence σ is smaller than the earliness of *j*, then eliminate branch *ij* and go to step 8. Otherwise, go to step 11.

Step 11. If the maximum earliness of set σ' with MST rule is greater than the maximum earliness of partial sequence σ, then go to step 12. Otherwise, eliminate branch *ij* and go to step 8.

Step 12. Create the maximum improvement in the objective function of partial sequence σ by using the idle insert. Consider the maximum of T_{max} of partial sequence σ and T_{max} of EDD rule of set σ' as the maximum tardiness of the lower bound. Moreover, consider the maximum earliness of partial sequence σ, after inserting idle insert, as the maximum earliness of the lower bound. The lower bound would be equal to the summation of the maximum earliness and the maximum tardiness of the lower bound.

Step 13. If the lower bound is smaller than the upper bound, then go to step 8. Otherwise, eliminate branch *ij* and go to step 8.

Step 14. Stop.

7. Proposed SA Algorithm

In this section, SA algorithm and the method applied to determine the input parameters are explained. To create a new neighborhood in this algorithm, a swap method is used, in which two adjacent jobs in the sequence are selected and their position is changed. All other parameters and the necessary conditions such as the initial temperature and the way it changes, equilibrium conditions and algorithm termination criteria are defined in this section. Following are the steps of the proposed SA algorithm:

Step 1. Assign value to the input parameters: ε, *e*, *m*.

Step 2. Calculate the initial feasible solution α_0 and T_{w0}, T_{wf} and set the values of *r*, *t* and *n* equal to zero.

Step 3. Calculate the objective function $f_0\ (T_r)$ for α_0. Set this value as the minimum value of the objective function in *E* (i.e., $E=f_0(T_r)$) and suppose the initial solution is the best answer until now: $\alpha^*=\alpha_0$.

Step 4. Generate new neighborhood by changing the position of two randomly selected adjacent jobs (swap method). The new solution obtained by this method is called α_{j}..

Step 5. Calculate both objective function value for α_j and $\Delta f(T_r)$ and $\Delta f(T_r)= f_j(T_r)-f_i(T_r)$. If $\Delta f(T_r)\leq 0$ then go to Step 7.

Step 6. Generate *Y* and $Y\sim u(0,1)$ and calculate $P(\Delta f)$ as follows:

$$P(\Delta f) = exp\left(\frac{-\Delta f(T_r)}{T_{wr}}\right) \tag{7}$$

If $Y\leq P(\Delta f)$, then go to Step 7, otherwise go to Step 4.

Step 7. Accept α_j and $n=n+1$. If the objective function value (OFV) for α_j ($f_j(T_r)$) is better than the best OFV found so far, then $E= f_j(T_r)$. If $n < e$ then go to Step 4, otherwise go to Step 8.

Step 8. Set $n=0$ and investigate the equilibrium conditions. If the number of accepted solutions in a specific temperature (i.e., t_t) is more than the maximum number of solutions in any temperature (i.e., *m*) then go to Step 9; otherwise, investigate the following inequality:

$$\left|\frac{\bar{f}_e(T_r) - \bar{f}_G(T_r)}{f_G(T_r)}\right| \leq \varepsilon \tag{8}$$

If the above inequality is satisfied then go to the next step; otherwise, go to Step 4.

Step 9. Set t=0. If $T_{wr}<T_{wf}$ then go to Step 11, otherwise go to the next step.

Step 10. Calculate $T_{wr+1}= 0.85\ T_{wr}$ and put $r=r+1$ then go to Step 4.

[Downloaded from ijiepr.iust.ac.ir on 2024-07-17]

Step 11. Introduce the best solution (E, α^*)

7-1. Specifying Control Parameters in The Proposed SA Algorithm

The proposed SA algorithm is an efficient meta-heuristic method which plays an important role in combinatorial optimization problems. It is categorized in a class of improvement algorithms which are able to improve the quality of a given initial solution based on the objective function criteria and is able to exit from the local optimal points due to accepting some bad solutions under specific conditions.

Although this algorithm has strong ability to generate good solutions, it has been shown that it is sensitive to its control parameters. There is no specific algorithm to determine the parameter values, so defining such parameters to get qualified answers is difficult. In the following section some of these parameters and their assigned values are discussed.

7-1-1. Primary Solution

The proposed SA algorithm is sensitive to the initial solution. Starting with a good initial solution will result a better final solution. In this paper, to get initial solution, jobs are arranged randomly. Based on one of the available heuristic methods in the literature and considering only one of the two functions E or T as an objective function, jobs are arranged and the resulting sequence is selected as an initial solution for the proposed SA algorithm.

7-1-2. Initiating Temperature

The number of iterations during the annealing process is relatively dependent on the initial temperature. Methods for determining the initial temperature may be divided in to two categories; one of them considers the initial temperature as a fixed number which must be specified before the annealing process. The other method determines the initial temperature by using the information obtained from previous practice before starting the main SA algorithm. In the proposed method, the initial temperature is calculated by performing some tests before starting the process.

Calculation procedure: a number of 100 new neighbors are created without considering changes in the objective function. T_{w0} is considered as the maximum change in the OFV for this method, i.e.,

$$T_{w0}=max\ \{-\Delta f_w\} \tag{9}$$

Δf_w is the amount of change in the OFV after changing the arrangement of the sequence to get a new neighborhood solution (i.e., swap).

7-1-3. Procedure of Changing The Temperature

One of the essential aspects in the annealing process is the way the temperature changes during the execution of the SA algorithm. In fact, temperature impacts the probability of accepting the worste answer. In case of higher temperatures some of the bad solutions are accepted and this causes the algorithm not to be trapped in local optimums. On the other hand in the case of low temperatures, there is a high probability to stick in one of the local optimal points and moving from that toward the global optimum is a hard job. There are two general methods for reducing the temperature in the literature: One method uses a function to reduce the temperature according to annealing process and the second method applies the information obtained from performing some iterations of the algorithm before starting the main SA procedure [17].

In the literature, it is more usual to use a function for reducing the temperature in annealing process for the sequencing problems and the same strategy has been applied in this study. The algorithm starts with the initial temperature and while the system reaches the equilibrium state, the temperature reduces according to the following equation which is called the geometric temperature reduction function:

$$T_{wr+1}=\alpha_0\times T_{wr} \tag{10}$$

T_{wr} is the temperature in stage r and α_0 is the coefficient factor for temperature decrease.

In this problem, after various numerical experiments, coefficient factor α_0 was selected equal to 85%; meaning that in each stage the temperature is decreased by 15%.

7-1-4. Method of Neighborhood Generation

In general, there are two methods to generate the neighborhood solutions. One is a random selection of solutions from a set of feasible solutions and the other method is to generate the feasible solution using the swap or insertion procedures. According to Sridhar [18] the second method, i.e. swap or insertion, acts more efficiently comparing to the first one. In this paper, the swap method is used to generate the new neighboring solutions.

7-1-5. Equilibrium Conditions

In each SA method, after performing a specific number of iterations in each temperature it is necessary to investigate the equilibrium conditions to be assured that if the annealing process can be continued in the current temperature or it is good to stop and continue the process after decreasing the temperature. In most of the cases reviewed in the literature, the specific number of iterations in each temperature has been used as the equilibrium condition.

In some cases this number is constant and in some other it changes according to a function during different phases of the algorithm [17]. In the proposed SA, after doing some iteration in each temperature, the following relationship is investigated. If it is satisfied the temperature will be decreased and the procedure continues in the new temperature. If it is not satisfied,

[Downloaded from ijiepr.iust.ac.ir on 2024-07-17]

the second condition will be checked. This criterion terminates further investigation in the current temperature when the number of accepted solutions in the current temperature reaches its maximum level, which is pre-defined for the algorithm. The advantage of this method is that it is not necessary to specify a number as the maximum number of new neighborhoods (m) to be searched before termination on that temperature and it saves time for further research for the new feasible points while the first condition is satisfied.

$$\bar{f}_e(T_r) = \frac{\sum_{i=1}^{l} f_i(T_r)}{l} \tag{11}$$

$$\left| \frac{\bar{f}_e(T_r) - \bar{f}_G(T_r)}{\bar{f}_G(T_r)} \right| \le \varepsilon_1 \tag{12}$$

where,

l : is the number of accepted solutions in each iteration of the algorithm

ε_1 : A small positive number defined to investigate the equilibrium conditions in each temperature. This number is specified during the different iterations of the proposed SA and depends on the type of the given problem and the result of the experiments.

$f_i(T_r)$: the objective function value for the i-th solution and in temperature T_r.

$\bar{f}_e(T_r)$: The average value of the objective functions for all accepted solutions during the current iteration and temperature (T_r).

$\bar{f}_G(T_r)$: The average of $f_i(T_r)$ for all of the previous iterations in temperature T_r.

After each iteration, the average of the objective functions is computed ($\bar{f}_e(T_r)$) and compared to the average of the previous iterations ($\bar{f}_G(T_r)$), if the above ratio is less than ε_1 then the equilibrium condition is satisfied and the procedure is terminated in the current temperature.

7-1-6. Algorithm Termination Conditions

There are many methods to stop the proposed SA. In this study, if the current temperature is less than the freezing state temperature T_{wf} the algorithm will be terminated and the best solution known until that time will be introduced as the final solution of the procedure.

7-1-7. Final Temperature

To calculate the final or freezing state temperature the following equation is used:

$$T_{wf} = \delta \times T_{w0} \tag{13}$$

T_{wf} : Final temperature

Final temperature is equal to δ times the initial temperature and this factor (δ) is considered equal to 0.04 through studies as well as analytical experiments.

8. Efficiency of The Proposed B&B and SA Methods

To show the efficiency of the B&B and the proposed SA methods in $n/1/I/ET_{max}$, it is necessary to design problems showing the strength of the proposed algorithms. In this section, a set of problems [15] is solved and the computational results are presented.

Many researchers have used random samples for test problems in the field of job earliness and tardiness. These researchers have considered two significant factors in these problems. The first factor is the tardiness represented by τ. This factor specifies the proportion of the average due dates of jobs to the sum of processing times in single machine problem. Ow and Morton [9], Kim and Yano [19], Yano and Kim [11] and James and Buchanan [14] have considered the above two factors and presented the following equation for τ:

$$\bar{d} = (1-\tau)\sum_{j=1}^{n} p_j \tag{14}$$

where $\bar{d}$ is the average due dates of jobs and p_j represents the processing time of job j. Processing times and τ are known as a priori and then $\bar{d}$ is obtained, accordingly. The second factor is the range of due date. According to Zegordi, *et al.* [10], processing times are generated by a uniform distribution in the range [5, 25]. $\bar{d}$ is obtained from Equation (14), first. Then, the due dates of jobs are defined by a uniform distribution as follows:

$$\left[\bar{d}\left(1-R/2\right), \bar{d}\left(1+R/2\right)\right] \tag{15}$$

In Equation (15), R is the range of due date and its value is known. Ow and Morton [9] have considered τ= 0.2, 0.6 and R= 0.6, 1.6. These standard values are used by most researches. Researchers use these values for generating test problems at random. To show the efficiency of the B&B method, four different types of problems are generated by combining two factors of τ and R.

These four types are first with τ= 0.2, R= 1.6, second with τ= 0.6, R= 1.6, third with τ= 0.2, R= 0.6 and fourth with τ= 0.6, R= 0.6. In each type, problems in sizes 5, 7, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, and 20 are considered. 15 iterations of each size in every type are solved. Thus, 195 problems for each type and 780 problems for four types are generated. In each size, if more than 80 percents of problems cannot be solved in

[Downloaded from ijiepr.iust.ac.ir on 2024-07-17]

a reasonable time, then the bigger sizes in that type are not generated. These problems are solved on a Pentium IV 1.2 GHz Processor.

Tables 3 to 6 show the computational results for type one, two, three and four, respectively. If the proposed B&B method succeeds to achieve the optimal sequence in time equal or smaller than 180 seconds, then the state of the solution is represented by "PBB". If the computational time exceeds the given time (180 seconds), then the algorithm is interrupted and the best solution up to this time is introduced.

This state of the solution is represented by "BF". The content of "time average of algorithm running" is the arithmetic mean of 15 iterations.

In Table 3, 32 and 163 problems have the states BF and PBB, respectively. As shown in Table 4, in problems of type one, all iterations until 14 jobs have the PBB state. The BF state is existed from 15 jobs upward, as for the problems with 15, 16, 17, 18, 19, and 20 jobs, the number of BF states is 3, 4, 6, 7, 7, and 10 items, respectively.

These numbers show that with increasing the number of jobs, the efficiency of the proposed method reduces, as in problems with 20 jobs only 1/3 of problems are achieved to a solution in a computational time less than 180 seconds. The computational time increases quickly, when the number of jobs increases, as shown in the mean computational time of Figure 2. In problems of type two, all iterations until 14 jobs have PBB state, but for the problems with 13 to 16 jobs, the number of BF states is 3, 5, 7, and 12 items, respectively. Thus, from 16 jobs upward, the efficiency of the proposed method reduces. In this type, the average of computational times for each size is greater than type one.

In problems of type three, all iterations until 10 jobs have PBB state, but for the problems with 11 and 12 jobs, the number of BF states is 1 and 15 items, respectively.

Thus, from 12 jobs upward, the efficiency of the proposed method reduces. In this type, the average of computational times for each size is greater than types one and two.

In problems of type four, all iterations until 10 jobs have PBB state, but for the problems with 11 and 12 jobs, the number of BF states is 3 and 15 items, respectively. Thus, from 12 jobs upward, the efficiency of the proposed method reduces. In this type, the average of computational times for each size is greater than all previous types. To show the efficiency of the proposed SA, four different types of problems are generated by combining two factors of tardiness and the range of due date, like the B&B method. In each type, problems in sizes 10, 20, 30, 40, 50, 60 and 70 are considered. 15 iterations of each size in every type are solved. Thus, 120 problems (15×7=105) for each type and 480 problems for four types (4×105=420) are generated. Tables 7 to 10 show the computational results for type one, two, three and four, respectively.

Tab. 3. Computational results of a B&B method in type 1 (τ=0.2, R=1.6)

Number of jobs	Number of problems	Time average of algorithm running (Sec.)	Branch and Bound with BF state	Branch and Bound with PBB state
5	15	0.00	0	15
7	15	0.00	0	15
10	15	0.11	0	15
11	15	0.59	0	15
12	15	1.48	0	15
13	15	6.02	0	15
14	15	13.74	0	15
15	15	44.53	3	12
16	15	90.00	4	11
17	15	98.70	6	9
18	15	116.80	7	8
19	15	128.71	7	8
20	15	152.00	10	5
Total	195		37	168

Tab. 4. Computational results of a B&B method in type 2 (τ=0.6, R=1.6)

Number of jobs	Number of problems	Time average of algorithm running (Sec.)	Branch and Bound with BF state	Branch and Bound with PBB state
5	15	0.00	0	15
7	15	0.00	0	15
10	15	0.43	0	15
11	15	0.63	0	15
12	15	13.79	0	15
13	15	41.89	3	12
14	15	83.00	5	10
15	15	103.60	7	8
16	15	155.70	12	3
Total	195		27	108

Tab. 5. Computational results of a B&B method in type 3 (τ=0.2, R=0.6)

Number of jobs	Number of problems	Time average of algorithm running (Sec.)	Branch and Bound with BF state	Branch and Bound with PBB state
5	15	0.00	0	15
7	15	0.03	0	15
10	15	13.66	0	15
11	15	86.96	1	14
12	15	180.00	15	0
Total	195		16	59

Tab. 6. Computational results of a B&B method in type 4 (τ=0.6, R=0.6)

Number of jobs	Number of problems	Time average of algorithm running (Sec.)	Branch and Bound with BF state	Branch and Bound with PBB state
5	15	0.00	0	15
7	15	0.02	0	15
10	15	14.03	0	15
11	15	152.94	3	12
12	15	180.00	15	0
Total	195		18	57

Tab. 7. Computational results of a SA method in type 1 (τ=0.2, R=1.6)

Number of jobs	Time average of B&B running (Sec.)	Time average of SA running (Sec.)
10	0.11	0.10
20	152.00	15.65
30	180.00	56.54
40		62.25
50		77.56
60		112.25
70		180.00

[Downloaded from ijiepr.iust.ac.ir on 2024-07-17]

Tab. 8. Computational results of a SA method in type 2 (τ=0.6, R=1.6)

Number of jobs	Time average of B&B running (Sec.)	Time average of SA running (Sec.)
10	0.43	0.20
20	180.00	17.50
30		60.23
40		70.23
50		95.25
60		126.35
70		180.00

Table 9. Computational results of a SA method in type 3 (τ=0.2, R=0.6)

Number of jobs	Time average of B&B running (Sec.)	Time average of SA running (Sec.)
10	13.63	4.12
20	180.00	56.43
30		85.25
40		90.35
50		134.00
60		180.00

Table 10. Computational results of a SA method in type 4 (τ=0.6, R=0.6)

Number of jobs	Time average of B&B running (Sec.)	Time average of SA running (Sec.)
10	14.03	8.19
20	180.00	89.76
30		96.84
40		102.86
50		180.00

The sensitivity of the computational time respect to the problem size, in these four types, is shown in Figures 2 and 3 respectively. According to these figures, it is concluded that the slope of chart is increased from type one to type four, and the speed of increasing in computational time is increased too, i.e. problem solving from type one to type four will be further difficult by using the proposed B&B and SA method. In every type (with constant τ and R), the computational time increases, when the number of jobs increases. Among different types, the problem solving will be more difficult, when R reduces. Furthermore, the difficulty of the proposed B&B and SA methods has a reverse relation with R and a direct relation with τ.

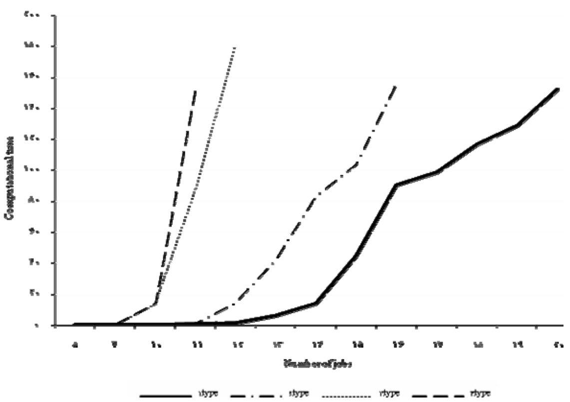

Fig. 2. Diagram of computational time with respect to the problem size using the B&B method

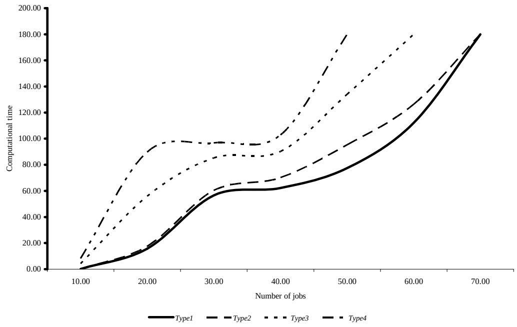

Fig. 3. Diagram of computational time respect to the problem size using the proposed SA method

The solutions obtained by SA algorithm are compared with solutions reported by the B&B method. The final solution is approximately optimal and computational time for achieving the best result is very small showing that the proposed algorithm is a quite suitable tool for solving the above problem.

9. Conclusion

This paper presents a single machine sequencing problem to determine the sequence of a set of jobs on a single machine. The associated objective function is to minimize the sum of maximum earliness and tardiness (ET_{max}). This objective can be adapted by any production system, in which the optimal sequence of a set of jobs is presented for a single machine with maximum earliness and tardiness considering the idle insert. In the general case, for $n/1/I/ET_{max}$, the neighborhood conditions were developed and the dominant set for the optimal solution is determined. The simulated annealing (SA) and branch-and-bound (B&B) methods are also applied to solve the above problem.

Future research can include other applications of this objective function (ET_{max}) in other types of sequencing problems, such as job shop, flow shop, and so forth, faster and more effective solving methods, as well as whatever changing in assumptions. Furthermore, the other subjects are listed as follows:

- Considering the utilization of the first depth in the branch and bound method. The utilization of the jump method can also be considered.
- Utilizing other optimization methods, such as dynamic programming.

Considering the utilization of the idle insert algorithm, using a branch-and-bound method and also generating the optimal sequence with a branch-and-bound method without using the idle insert, and finally comparing the results by obtaining the value of improvement in the objective function, using the idle insert algorithm.

References

[1] Baker, K.R. "*Introduction to Sequencing and Scheduling*," John Wiley, NY, 1974.

[Downloaded from ijiepr.iust.ac.ir on 2024-07-17]

[2] French, S., *"Sequencing and Scheduling: an Introduction to the Mathematics of the Job-Shop*," Ellis Horwood Ltd, Chichester, 1986.

[3] Ben-Daya, M., Al-Fawzan, M., *"A Simulated Annealing Approach For the One-Machine Mean Tardiness Scheduling Problem,"* European Journal of Operational Research, vol. 93, 1996, pp. 61-67.

[4] Emmons, H., *"One-Machine Sequencing to Minimize Certain Function of Job Tardiness*," Operation Research, Vol. 17, 1969, pp. 605-612.

[5] Sen, T., Borah, B.N., "*On the Single-Machine Scheduling Problem With Tardiness Penalties*," Journal of the Operational Research Society, Vol. 42, 1991, pp. 695-702.

[6] Holsenback, J.E., Russell, R.M., *"A Heuristic Algorithm for Sequencing on One Machine to Minimize Total Tardiness*," Journal of the Operational Research Society, Vol. 43, 1992, pp. 53-62.

[7] Panwalker, S.S., Smith, M.L., Koulamas, C.P., *"A Heuristic for the Single Machine Tardiness Problem,"* European Journal of Operational Research Society, Vol. 70, 1993, pp. 304-310.

[8] Islam, A., Oksioglu, M., *"A Tabu Search Approach for the Single Machine Mean Tardiness Problem*," Journal of the Operational Research Society, Vol. 48, 1997, pp. 751-755.

[9] Ow, P.S., Morton, T.E., "*The Single Machine Early/tardy Problem*," Management Science, Vol. 35, 1989, pp. 177-191.

[10] Zegordi, S.H., Itoh, K., Enkawa, T., "*A Knowledgeable Simulated Annealing Scheme for the Early/tardy Flow Shop Scheduling Problem,"* International Journal of Production Research, Vol. 33, 1995, pp. 1449-1466.

[11] Yano, C.A., Kim, Y.D., *"Algorithms for a Class of Single-Machine Weighted Tardiness and Earliness Problems*," European Journal of Operational Research, Vol. 52, 1991, PP. 167-175.

[12] Szwarc, W., Mukhopadhyay, S.K., "*Optimal Tming Schedules in Earliness-Tardiness Single Machine Sequencing*," Naval Research Logistics, Vol. 42, 1995, pp. 1109-1114.

[13] Dileepan, P., *"Common Due Date Scheduling Problem with Separate Earliness and Tardiness Penalties*," Computer and Operational Research, Vol. 20, 1993, pp. 179-184.

[14] James, R.J.W., Buchanan, J.T., *"A Neighborhood Scheme With a Compressed Solution Space for the Early/tardy Scheduling Problem,*" European Journal of Operational Research, Vol. 102, 1997, pp. 513-527.

[15] Amin-Nayeri, M.R., Moslehi, G., "*Optimal Algorithm for Single Machine Sequencing to Minimize Early/tardy Cost*," Esteghlal, Isfahan University of Technology, Vol. 1, 2000, pp. 35-48. (in Farsi)

[16] Tavakkoli-Moghaddam, R., Moslehi, G., Vasei, M., Azaron, A., *"Optimal Scheduling For a Single Machine to Minimize the Sum of Maximum Earliness and Tardiness Considering Idle Insert*," Applied Mathematics and Computation, Vol. 167, No. 2, 2005, pp. 1430-1450.

[17] Souilah, A., "*Simulated Annealing for Manufacturing Systems Layout Design*," European journal of operational research, Vol. 82, No. 3, 1995, pp. 592-614.

[18] Sridhar, J., Rajendran, C., "*Scheduling in a Cellular Manufacturing System: A Simulated Annealing Approach*," International journal of production research, Vol. 31, No. 12, 1993, pp. 2927-2945.

[19] Kim, Y.D., Yano, C.A., *"Minimizing Mean Tardiness and Earliness in Single-Machine Scheduling Problems With Unequal Due Date,"* Naval Research Logistics, Vol. 41, 1994, pp. 913-933.

[Downloaded from ijiepr.iust.ac.ir on 2024-07-17]