International Journal of Operations Research

International Journal of Operations Research Vol. 4, No. 1, 32−41 (2007)

Early-Tardy Minimization for Joint Scheduling of Jobs and Maintenance Operations on a Single Machine

Syed Asif Raza[1,*], Umar Mustafa Al-Turki[2], and Shokri Zaki Selim[2]

[1]Department of Mechanical and Industrial Engineering, Concordia University, 1455 de Maisonneuve Blvd. W. Montreal, Quebec H3G 1M8 Canada

[2]Systems Engineering Department, King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia

Received July 2006; *Revised October 2006*; *Accepted November 2006*

Abstract**—**In this paper, we consider joint scheduling of jobs and preventive maintenance operations on a single machine with an objective to minimize the total earliness and tardiness of jobs about a common due date. The properties of an optimal schedule are identified and utilized to develop a constructive heuristic and a lower bound estimate. The properties are also utilized to hybridize Tabu search and Simulated Annealing algorithms. A numerical study with over 3200 randomly generated problems is reported to demonstrate the performance of the proposed solution methods. The study shows that the effectiveness of the proposed lower bound and constructive heuristic is sensitive to maintenance related parameters. We also show that hybridized Tabu search and Simulated Annealing algorithms are efficient approaches to solve the problem.

Keywords**—**Single machine scheduling, Maintenance, Early-Tardy, Common due date, Tabu search and simulated annealing

1. INTRODUCTION

The importance of Preventive Maintenance (PM) is well established in a manufacturing environment. A sound PM program results in reducing corrective maintenance cost and cost of defective production. It also increases the availability of the production facility.

Schmidt (1988) considered scheduling jobs with due dates on parallel machines having availability intervals. Qi et al. (1999) studied scheduling jobs on a single machine that requires preventive maintenance. The time elapsed between to maintenance activities can not exceed a given value. Lee and Chen (2000) considered the parallel machine version with the objective of minimizing the weighted total completion time. Lorigeon et al. (2002) studied two-machine open shops subject to availability constraint with the objective of minimizing total completion time. Chen and Powell (2003) considered a situation where jobs are classified into families. No set up is needed when processing jobs of the same family. A setup is however needed when there is a switch from one job family to another. They considered two problems involving identical machines. In the first problem the total weighted completion time is minimized and in the second problem the total weighted number of tardy jobs is minimized. Cassady and Kutanoglu (2003) consider the problem of joint scheduling of jobs and preventive maintenance such that total tardiness is minimized. Aggoune (2004) minimize the make span of a flow shop with availability constraints. Two maintenance policies are considered. In the first, maintenance starting times are fixed, where as in the second maintenance must be performed within a given time window. Chen (2004) considered a parallel machine scheduling problem involving job scheduling and resource allocation. The processing times are inversely related to funds allocated to resources. The objective is to minimize total cost of scheduling and resource allocation. Adzakpa et al. (2004) developed heuristics for scheduling jobs on parallel machine with the objective of minimizing weighted flow time. In another paper, Adzakpa et al. (2004) consider online scheduling and assignment of maintenance on a single machine with availability constraint, on a given time window with the objective of minimizing cost of discharge of maintenance or jobs. Akturk et al. (2003) consider the problem of tool change due to wear on a single computer numerical control machine. The objective is to minimize total completion time. Akturk et al. (2004) studied the performance of the Shortest Processing Time (SPT) rule for the same problem. Liao et al. (2005) considered a two parallel machines problem where one machine is not available during a fixed and known time period. The objective is to minimize the make span for both non-resumable and resumable cases. Sortrakul et al. (2005) considered an integrated optimization model for production scheduling and preventive maintenance planning.

Early-Tardy minimization is a non-regular performance measure. It is of particular importance in just-in-time manufacturing systems. Initial work on minimizing completion time variance was conducted by Merten and

*Corresponding author's email: asif_s@encs.concordia.ca

Muller (1972) and Schrage (1975). Eilon and Chowdhury (1977) extended the work of Schrage by proving the V-Shaped property. A sequence is V-Shaped if all jobs before the job with least processing time are sequenced according to Longest Processing Times (LPT) and those after it are sequenced according to SPT. Kanet (1981) considered a single machine problem where penalty is incurred for late and early jobs. The objective is to minimize the total penalty. The assumption here is that the due date is sufficiently large. Sundararaghvan and Mesbah (1984) extended this work to the case of identical machines. Mazzini and Armentano (2001) considered single machine scheduling with due dates, ready time and shut down constraint. Tardiness is not allowed, however, earliness is penalized. Due to the fact that shutdowns are known, this work is different from that of Qi et al. (1999). Bulbul et al. (2004) considered the problem of scheduling customer orders with the objective of minimizing tardiness and earliness.

In this paper, we discuss the problem of joint scheduling of independent jobs and preventive maintenance on a single machine with no pre-emption, no machine breakdowns and no setup times. The machine ready time and job release times are all zeros. The objective is to minimize the total un-weighted earliness and tardiness about a given common due date. This problem is an extension of that of Qi et al. (1999).

The rest of this paper is organized as follows; we define the problem, introduce notation and introduce some properties of an optimal schedule in Section 2. In Section 3, we show that the problem is NP-hard and give a lower bound estimation method of the objective function. A constructive heuristic and two meta-heuristics are proposed in Section 4. Computational experience with the proposed algorithms are reported in Section 5. Section 6 contains conclusion regarding the computational experience followed by suggested future research.

2. PROBLEM DEFINITION AND PROPERTIES OF AN OPTIMAL SCHEDULE

We first define the problem, P, for any performance measure then adopt the definition to the case of total earliness and tardiness minimization about a common due date. The problem is also equivalent to the objective of minimization of the total absolute deviation of the completion times of the jobs about a common due date. The problem is to schedule n jobs, $J_1, J_2, \ldots, J_n$, available at time zero on a single machine such that a given performance measure is optimized. The maintenance time is t. The machine can not be operated for a period exceeding T. The processing time p_i is deterministically known for a job, J_i and jobs are indexed in the non decreasing order of their processing times, i.e., $p_1 \le p_2 \le \ldots \le p_n$. No preemption is allowed, also $p_n \le T$. A typical schedule, S contains a sequence of jobs and maintenance has to be performed in between the jobs such that the total continuous operation time of the machine does not exceed the time limit T. Jobs are processed continuously in batches, denoted as H. Thus a schedule, S is denoted as $S = \{H_1, M, H_2, M, \ldots, M, H_L\}$, where M represents preventive maintenance. Each occurrence of M has a cost of time, t in performance measure. Each batch H_i, $i = 1, 2, \ldots, L$ contains jobs with a constraint that the sum of processing times of jobs in each batch does not exceed T.

In this paper, we consider the problem of minimizing the total earliness and tardiness about a common due date. The due date d, is assumed unrestrictive, i.e., it is not a constraint for any job from being early. Next we introduce some notation that we use in the rest of the paper.

A_k:	Batch k containing tardy jobs
B_k:	Batch k containing early jobs
qa_k:	Sum of processing times of jobs in batch A_k
qb_k:	Sum of processing times of jobs in batch B_k
la:	Number of batches of tardy jobs
lb:	Number of batches of early jobs
na_k:	$\lvert A_k \rvert$
nb_k	$\lvert B_k \rvert$
SA:	$\{A_1, A_2, \ldots, A_{la}\}$
SB:	$\{B_{lb}, B_{lb-1}, \ldots, B_1\}$
SAM:	$\{A_1, M, A_2, M, \ldots, A_{la-1}, M, A_{la}\}$
SBM:	$\{B_{lb}, M, B_{lb-1}, M, \ldots, B_2, M, B_1\}$

SAM is the set of tardy jobs and maintenance operations performed after the due date. Likewise SBM is a set of early jobs and maintenance operations before due date. We consider a schedule $S = \{SBM, SAM\}$, in which jobs are either early or tardy and exactly one job completes on the due date. All maintenance operations are scheduled either before or after the common due date. The last job in SBM finishes on d. Later we prove that having such characteristics in any feasible schedule is essential for optimality. We define the objective function for the problem with a given schedule S.

$$f(S) = \sum_{i=1}^{n} \lvert C_i - d \rvert$$

where C_i is the completion time of J_i in a given schedule S. Lemma 1 due to Kanet (1981) shows that the minimum of $f(S)$ is achievable in polynomial time.

Lemma 1 (Kanet, 1981). A schedule is characterized as an optimal schedule for the objective of minimization of total absolute deviation of completion times of jobs from a common due date such that following conditions are satisfied:

1. All jobs completing before a common due date are sequenced using LPT rule and all jobs completing after the due date are sequenced using SPT rule.
2. If total number of jobs is even, then total number of tardy and early jobs is same. If total number of jobs is odd, then total number of early jobs exceed the total number of tardy jobs exactly by one job.

Now we identify some important properties of an optimal schedule. These properties are useful for identifying an optimal schedule. The properties can also be used to construct efficient heuristic algorithms or be embedded into meta-heuristics that make them able to solve the problem efficiently to near optimality with a significant CPU time saving.

In the following, an activity is either a job or a maintenance operation.

Property 1. There is an optimal schedule in which no activity starts before the due date and finishes after it.

Proof. The proof is by contradiction. Let S be an optimal schedule where an activity starts before the due date and finishes after it. Let t_B and t_A be the activity times before and after d respectively. Next we consider two cases.

Case I: $|SAM| \geq |SBM|$. Construct a schedule S_1 which has the same sequence as S, however its start time is delayed by an amount of time t_B. Then, $f(S) - f(S_1) = |SAM|t_B - |SBM|t_B \geq 0$, and S_1 is superior to S. This is a contradiction.

Case II: $|SBM| \geq |SAM|$. Construct a schedule S_2 which has the same sequence as S, however its start time is delayed an amount of time t_A. Then, $f(S) - f(S_2) = |SBM|t_A - |SAM|t_A \geq 0$, and S_2 is superior to S. This is also a contradiction. This completes the proof.

Property 2. In an optimal schedule, jobs in batches B_k, k = 1, 2, …, lb are sequenced in LPT order, and jobs in batches A_k, k = 1, 2, …, la are sequenced in SPT order.

Proof. Consider a schedule S where an arbitrary tardy batch has jobs in non-SPT sequence. Suppose J_i and J_m are adjacent and $p_i > p_m$. Construct a schedule S' which is the same as S except that the sequence of these two jobs is reversed. Let a be the start time of J_i. Then the sum of tardiness of J_i and J_m under $S = (a + p_i - d) + (a + p_i + p_m - d)$, and it is $(a + p_m - d) + (a + p_i + p_m - d)$ under S'. Thus, $f(S) - f(S') = p_i - p_m > 0$, and S can not be optimal. A similar argument can be used for early batches. This completes the proof.

Property 3. In an optimal schedule, the following inequalities hold:

$$T - qa_k < p_i, \ \forall J_i \in A_r, \ 1 < k < la \text{ and } r > k$$

$$T - qb_k < p_i, \ \forall J_i \in B_r, \ 1 < k < lb \text{ and } r > k$$

Proof. Consider a schedule S with an early set $SBM = \{B_1, M, ..., M, B_k, M, ..., M, B_{Lb}\}$, such that there is a batch B_k with J_m in the first position in the batch and J_h is the last job that satisfies $qb_k + p_h \leq T, J_h \in B_r$, for some $r > k$. Let Ψ be the set of jobs that are between J_h and J_m. Job J_h can be placed at the first position in B_k. The earliness of J_h will be decreased by an amount $(k-r)t + \sum_{i \in \psi} p_i$. However, the earliness of jobs in Ψ will be increased by an amount $|\Psi|p_h$. Since $p_h < p_i$, for all $i \in \Psi$, then the new schedule is superior to S. A similar proof can be used for the other inequality.

In the next, we prove a result similar to that of Qi et al. (1999). Consider a batch D that starts at time 0. Let C_{0i} be the completion time of job $J_i \in D$. Suppose now that D starts at time $a > d$, then the total tardiness of the jobs in D is given by $(a-d)|D| + Q_D$, where $Q_D = \sum_{J_i \in D} C_{0i}$.

Lemma 2. At an optimal sequence for problem P, the following inequalities hold

$$\frac{t + q_{ak}}{|A_k|} \leq \frac{t + q_{ak+1}}{|A_{k+1}|}, \ 1<k<la$$

$$\frac{t + q_{bk}}{|B_k|} \leq \frac{t + q_{bk+1}}{|B_{k+1}|}, \ 1<k<lb$$

Proof. Consider two adjacent batches A_k followed by A_{k+1}. Their total tardiness is given by:

$$TT_{k,\, k+1} = (a-d)|A_k| + Q_k + (a + qa_k + t - d)|A_{k+1}| + Q_{k+1}$$

If the batches switch their sequence then their tardiness is:

$$TT_{k,\, k+1} = (a-d)|A_{k+1}| + Q_{k+1} + (a + qa_{k+1} + t - d) \times |A_k| + Q_k$$

For the sequence, $A_k - A_{k+1}$, to be superior to the sequence $A_{k+1} - A_k$, we must have $TT_{k,\, k+1} \leq TT_{k+1,\, k}$. This simplifies to the first inequality of this lemma. Similarly the second inequality of the lemma can also be proved.

Property 4. In an optimal schedule of problem P, the following inequalities hold:

$$|A_2| \geq |A_3| \geq ... \geq |A_{la}|$$

$$|B_2| \geq |B_3| \geq ... \geq |B_{lb}|$$

Proof. The proof is by contradiction. Consider two consecutive tardy batches A_k and A_{k+1}, $k > 1$. Then from Lemma 2 we have

$$T - qa_k \geq T - \left[|A_k| qa_{k+1} + (|A_k| - |A_{k+1}|)t \right] / |A_{k+1}|$$
$$= \left[(|A_{k+1}| - |A_k|)(T+t) + |A_k|(T - qa_{k+1}) \right] / |A_{k+1}|$$
$$\geq (|A_{k+1}| - |A_k|)(T+t) / |A_{k+1}|$$

If $|A_{k+1}| > |A_k|$, then $T - qa_k \geq T / |A_{k+1}| > p_{min}$

where p_{min} is the shortest processing time in batch A_{k+1}. This contradicts with Property 3 and thus proves Property 4. A similar argument can be used for the second inequality.

3. NP-HARDNESS AND LOWER BOUND

In this section, we show that problem P, is NP hard. Later we derive a lower bound for the cost of an optimal solution. In Proposition 1, we identify that the problem for minimization of total absolute deviation of completion times of jobs about a common degenerate due date i.e., $d = 0$, is the problem studied in Qi et al. (1999).

Proposition 1. The problem of minimizing total tardiness about a common due date at time zero is equivalent to the problem of minimizing total completion time.

Proof. Since all jobs are tardy and $d = 0$, then the tardiness of a job is also its completion time.

Corollary 1. The problem, P, with a common due date, $d = 0$, is NP-hard in strong sense.

Proof. Since minimizing the total completion time is NP-hard in the strong sense as shown by Qi et al. (1999), then from Proposition 1, Problem P has the same complexity.

We consider a superoptimal solution to the problem as a lower bound estimate for the cost of an optimal schedule. The lower bound may not satisfy all constraints and can be achieved at the best using any global optimization procedure. The main theme is to integrate properties of an optimal schedule identified in Section 2 and results of Lemma 1. We first state Proposition 2 that help us to establish a lower bound.

Proposition 2. A Schedule with preventive maintenance constraint is optimal if:

1. The jobs are sequenced using the rule of Lemma 1, and,
2. $qa_k = T,\ k = 1, 2, ..., la,$ and $qb_k = T,\ k = 1, 2, ..., lb.$

Proof. The problem of minimizing the total earliness and tardiness about a non-restrictive common due date is a relaxation of Problem P. Hence a solution of the former problem that satisfies the maintenance constraint is optimal for the problem P. This completes the proof.

Consider a schedule *SS* which is V-shaped. Construct tardy batches in which $qa_k = T$ or $qa_k - T \leq p_m$, also construct early batches in which $qb_k = T$ or $qb_k - T \leq p_m$. where J_m is the last job in the batch. The lower bound is thus given by $LB = f(SS)$.

4. PROPOSED SOLUTION ALGORITHMS

We propose three heuristic algorithms to solve problem P. The first algorithm is a constructive heuristic. It is based on using the properties in Section 2. The other two algorithms are meta-heuristics based on Tabu Search and Simulated Annealing algorithms. Both Tabu Search and Simulated Annealing algorithms utilize the aforementioned properties of an optimal schedule and thus these meta-heuristics are hybridized. We discuss the three proposed algorithms briefly as follows.

4.1 Heuristic solution algorithm

We use the properties of an optimal schedule to develop a Heuristic Algorithm (HA) that finds a near optimal solution in a single pass. The salient feature of HA is its combined use of V-shaped job sequencing properties and optimal maintenance scheduling rules. It mainly schedules the jobs in V-shaped about the common due date *d*. The last job in the early set finishes at the due date. The maintenance operations are scheduled starting from first tardy job until the last tardy job as late as possible. Similarly, the maintenance operations are scheduled starting from last early job until the first early job as late as possible. Using this scheduling policy we are able to minimize the total absolute deviation of the completion times of all jobs about a given common due date *d*. We mention the details of HA as follows:

Step 1. U = {Universal set of jobs}, $SA = \phi$, $SB = \phi$.

Step 2. If $U \neq \phi$, then remove a job J_k from U set where $p_k = \max\{p_i \mid p_i \in U\}$. Insert job J_k in the last position of *SB*. Else go to *Step* 3. Again check if $U \neq \phi$, then remove job J_k from U set where $p_k = \max\{p_i \mid p_i \in U\}$ and insert it in the first position of *SA*, repeat *Step* 2. Else go to *Step* 3.

Step 3. $i = 1, j = 1, qa_i = 0, qb_j = 0, A_i = \phi, B_j = \phi$.

Step 4. If $(qa_i + qb_j) + p_k \leq T$, where p_k is the processing time of the last job in *SB*, then remove job J_k from *SB* and insert it at the first position in B_j. Set $qb_j = qb_j + p_k$, $nb_j = nb_j + 1$. Else, $i = i + 1, j = j + 1$, and go to *Step* 6.

Step 5. If $(qa_i + qb_j) + p_k \leq T$, where p_k is the processing time of the first job in *SA*, then remove job J_k from *SA* and insert it at the last position in A_i. Set $qa_i = qa_i + p_k$, $na_i = na_i + 1$, repeat *Step* 4. Else, $i = i + 1, j = j + 1$, and go to *Step* 6.

Step 6. If $qa_i + p_k \leq T$, where p_k is the processing time of the first job in *SA*, remove job J_k from *SA* and insert at the first position in A_i, $qa_i = qa_i + p_k$, $na_i = na_i + 1$. Else $i = i + 1$. Repeat this step until $SA = \phi$.

Step 7. If $qb_j + p_k \leq T$, where p_k is the processing time of the last job in *SB*, remove job J_k from *SB* and insert at the last position in B_j, $qb_j = qb_j + p_k$, $nb_j = nb_j + 1$. Else $j = j + 1$. Repeat this step until $SB = \phi$.

Step 8. $SAM = \{A_1, M, A_2, M, ..., A_{la-1}, M, A_{la}\}$, $SBM =$

$\{B_{lb}, M, B_{lb-1}, M, ..., B_2, M, B_1\}$, $S = \{SBM, SAM\}$.

4.2 Hybrid tabu search

Tabu Search (TS) was proposed by Glover (1986). It is a meta-heuristic that can be superimposed on another heuristic. TS begins by marching to a local minima. To avoid retracing previous steps, the method records recent moves in one or more Tabu lists. The intent of the list is not to prevent a previous move from being repeated, but rather to insure it is not reversed. Tabu lists are historical in nature and form the TS memory. The role of the memory can change as the algorithm proceeds. Tabu status of a move is overridden when certain criteria (aspiration criteria) are satisfied. More details about this method can be found in Glover and Laguna (1997). We call this implementation of TS as Hybrid Tabu Search (HTS) because TS procedure is hybridized with the use of the properties of an optimal schedule identified in Section 2. It results in convergence to "near" optimal solution with a significant saving in CPU time. In this implementation, the search starts with an arbitrary feasible schedule called seed solution. The seed solution is considered as current solution in the search. Several candidate solutions (feasible schedules) are generated using a neighborhood generation scheme. The moves of the best candidate solution are checked in the tabu list. If the moves of best candidate are found in tabu list but it satisfies the aspiration criterion then it is also accepted as current solution for the next search iteration, otherwise this step is repeated. The search terminates when a stopping criterion is reached. We discuss the detailed features of the proposed Hybrid Tabu Search (HTS) algorithm for this problem as follows:

- **Seed solution:** A seed solution is any sequence of jobs that satisfies the preventive maintenance requirement.

- **Neighborhood:** A neighborhood solution S' is obtained by swapping two randomly selected jobs. Two distinct policies are adopted with an equal chance in the process of random generation of neighborhood. In the first policy, we swap two jobs between any two distinct batches in an existing schedule. The second policy is to swap two randomly selected job, one job early and other job is tardy. If any of these swaps results in a infeasible schedule i.e., maintenance constraint is not met, then we use following procedure to achieve feasibility on the same job sequence.

  Step 1. Remove the maintenance i.e., M, from infeasible neighbor schedule, S'. Assign first $\upsilon = \lceil n/2 \rceil$ jobs to SB and remaining to SA in the same order.
  Step 2. Use *Steps* 3 onwards of HA.
  Step 3. Rearrange jobs in each early batch and tardy batch in LPT order and SPT order respectively.
  Step 4. Re-index the early and tardy job batches such the following rules are satisfied.

$$\frac{qa_2 + t}{na_2} \le \frac{qa_3 + t}{na_3} \le ... \le \frac{qa_{la} + t}{na_{la}}$$

$$\frac{qb_2 + t}{nb_2} \le \frac{qb_3 + t}{nb_3} \le ... \le \frac{qb_{lb} + t}{nb_{lb}}$$

Candidate list size: It is a list containing a subset of neighborhood moves examined. A candidate list size of 20 is selected for each iteration, after using the conclusions from a series of tests performed in Raza (2002).

Tabu restriction: In our implementation, attributes of a schedule are jobs swapped in a schedule which are recorded in tabu list. The tabu list can store a maximum of 7 moves and it is updated using First In First Out (FIFO) strategy.

- **Aspiration criterion:** It is satisfied when the best neighbor solution of the current iteration is found better than the best solution visited so far.

- **Stopping criterion:** The algorithm is stopped after 5000 iterations of no improvement.

4.3 Hybrid simulated annealing

Simulated Annealing (SA) was proposed by Kirkpatrick et al. (1983). SA follows an analogy from annealing of metal. During the search process SA not only accepts better solutions (Downhill move) but also accepts bad solutions (Uphill move) with some probability. This feature of SA enables the search to escape a local minimum. The SA algorithm requires a seed solution, metropolis criterion, cooling schedule, acceptance probability function and stopping criterion. The SA algorithm also makes use of properties of an optimal schedule aforementioned in this paper and hence we call it Hybrid Simulated Annealing (HSA) algorithm. The algorithm starts with a seed solution (feasible schedule) at a high temperature such that the most feasible neighborhood solutions of the seed solution are accepted. At a particular temperature the metropolis loop is executed for a fixed markov chain length in order to achieve the quasi equilibrium state is attained at that temperature. At each loop, the neighbor solution is accepted if it outperforms its generator solution, however a poor solution is also accepted but follows a probabilistic acceptance function. A temperature decrement rule is applied once quasi equilibrium state is reached at a particular temperature. The metropolis loop uses the neighborhood generation scheme same as suggested in HTS. The cooling schedule and acceptance probability functions in particular to this HSA implementation are described as follows:

- **Cooling Schedule**: The main parameters of a cooling schedule are: Initial temperature; temperature decrement rule; and final temperature at which the annealing process is stopped.
  Initial Temperature: We use the method for

estimating Y_0 proposed by White (1984). In this approach, the system is considered hot enough if $Y_0 \gg \sigma$ where σ is the standard deviation of the cost function at initial temperature Y_0. The following equation uses the stated criterion:

$$\psi = -\frac{3}{\ln R}$$

$$Y_0 = \psi\sigma$$

σ is determined based on 100 randomly generated neighbors of an arbitrary seed solution, and R is the percentage of accepted solutions.

Temperature Decrement Rule: In most SA approaches, geometric temperature decrement rule is employed. If the temperature at iteration k is Y_k then the temperature at iteration $k + 1$ is given by:

$$Y_{k+1} = \alpha Y_k$$

where $0.8 \le \alpha \le 0.99$ in most SA applications (Sait and Youssef, 1999). In our implementation, $\alpha = 0.99$.

Final Temperature: In this implementation, the lowest allowable temperature is set to 10^{-3}.

- **Markov Chain Length:** The markov chain length describes the number of times the Metropolis loop is executed at a given temperature to attain quasi-equilibrium (Eglese, 1990). In this study, the markov chain length is 20.

- **Acceptance Probability Function:** In the present HSA algorithm, we use the statistical acceptance probability function (Sait and Youssef, 1999; Lyu et al., 1996). At a given temperature Y_i, the acceptance probability function p_a of a solution (schedule) S' is given as:

$$p_a = \begin{cases} 1, & \text{If } f(S') < f(S) \\ \exp(\Delta/Y_i), & \text{If } f(S') \ge f(S) \end{cases} \tag{1}$$

where $\Delta = f(S') - f(S)$

- **Stopping Criterion:** HSA stops if 5000 iterations result in no improvement in the best solution observed.

5. COMPUTATIONAL RESULTS AND ANALYSIS

The proposed HA, HTS, HSA algorithms and LB are coded in Compaq Visual Fortran version 6.6. The numerical experimentation is carried out on an Intel Pentium 4, 2.40 GHz processor based on a stand alone workstation with 256 MB RAM. The proposed heuristic HA is compared with the HTS and HSA. The processing times of jobs are randomly generated between 1 to 30 using uniform distribution. The common due date d, is determined by summing the processing times of all job and an estimate of least possible total time needed for maintenance and is reported in Eq. (2).

$$d = \sum_{i=1}^{n} p_i + \left\lceil \sum_{i=1}^{n} p_i / T \right\rceil t \tag{2}$$

Four distinct job sizes n = 15, 20, 25 and 30 are considered. The other relevant parametric values considered are, T = 50, 60, 70, and 80, similarly, t = 10, 20, 30, and 40. In each job size, 50 problems are solved for each combination of T and t. We conclude that there is a trend of improvement in the performance of HSA and HTS over HA when the job size increases. The fact is further reported in Figure 1 in the form of a main effect. In the figure, the number of problems are reported in which HTS and HSA are able to find a superior solution to the problem when compared with HA. Numerical experiments clearly demonstrate that both HTS and HSA perform better as the job size increases. Furthermore, HSA performance is found superior to HTS. In the experiment with n = 30, HSA was able to outperform HA in 736 problems out of 800. HTS improves 710 problems out of 800 problems in the same experiment. For n = 15, these findings for HTS and HSA are found 423 and 519 respectively. Similarly in Figure 2, the effect of the job size on relative improvement made by HTS and HSA over HA is reported. The relative improvement increases with an increase in the job size. The average relative improvement for the large job size, (n = 30), is at least 2.3%. The impact of increase in the job size on CPU time is reported in Figure 3, but it is considered insignificant as most of the problems are solved in less than 30 seconds of CPU time.

The effect of the maximum time limit of a continuous operation, T is also considered in the experiment. We noticed that a reduction in T given a fixed t also improves the performance of HTS and HSA. This trend is further studied considering the main impact of T in Figure 4. In the figure, the effect of T on relative deviation of HA, HTS and HSA from LB is reported. The main effect of T on improvement achieved by HTS and HSA over HA is also reported in Figure 5. It reveals that the performance of HA improves as T increases, and LB is also able to generate a tighter estimate on optimal solution with an increase in T. Among all the problems tested with T = 50, the average relative improvement of HTS and HSA over HA is 3.69% and 3.70% respectively. At T = 80, the average relative improvement of HTS and HSA decreases to 1.01 and 0.83% respectively.

The impact of maintenance time, t is opposite to that of T. From Figures 6 and 7, it can be inferred that the performance of HA deteriorates with an increase in maintenance time, t, thus HTS and HSA better improve over HA as t increases. At t = 10, average improvement of HTS and HSA over HA is 0.89 and 0.85% respectively, which increases to 3.18 and 3.27% respectively at t = 40.

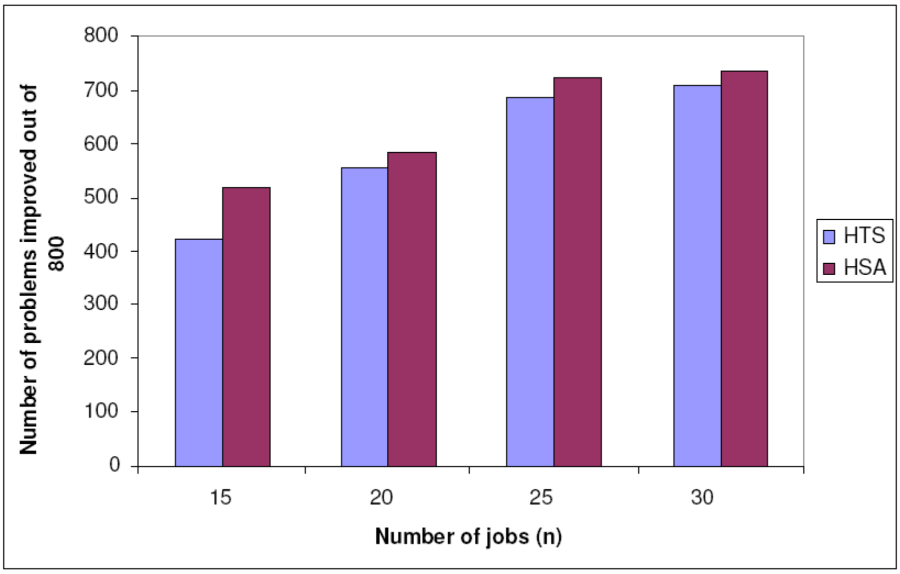

Figure 1. Effect of job size *n* on performance of proposed algorithms.

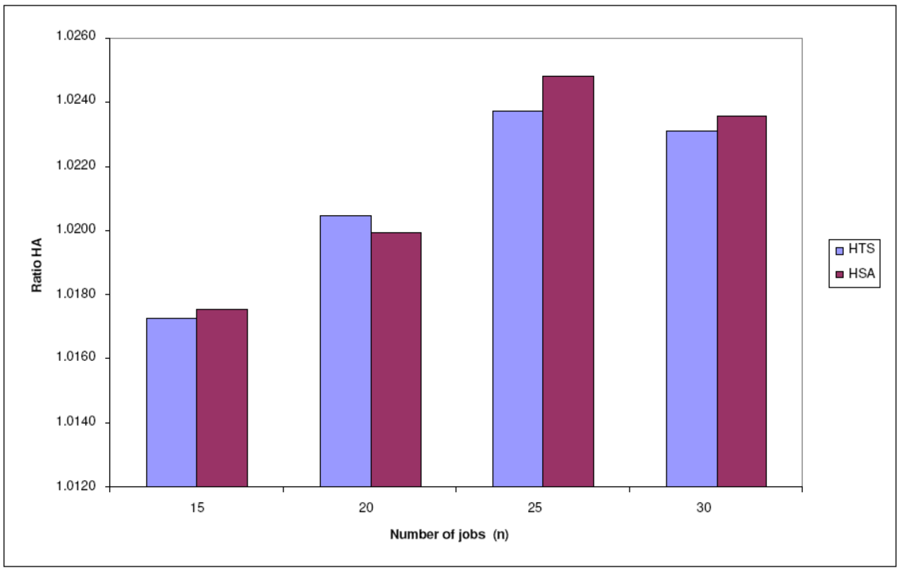

Figure 2. Effect of job size *n* on improvement over HA.

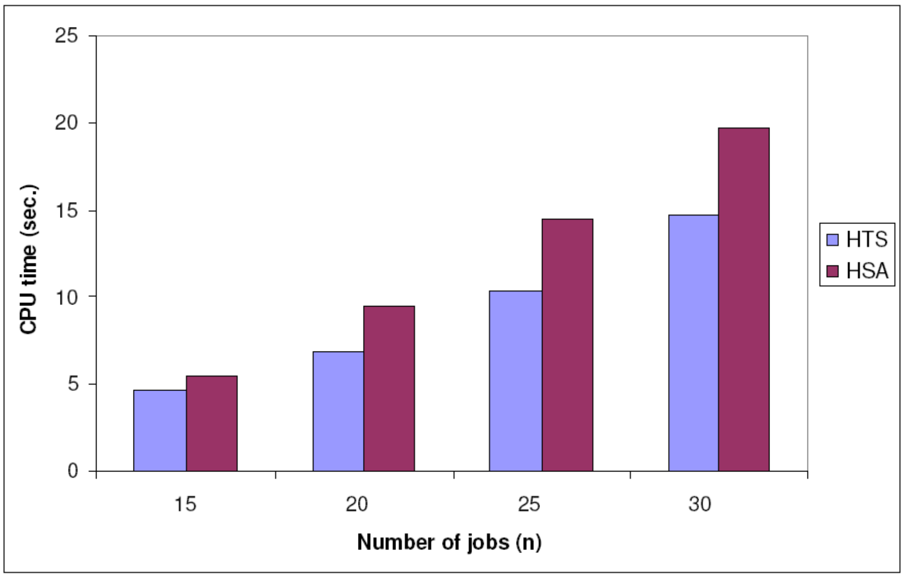

Figure 3. Effect of job size n on CPU time.

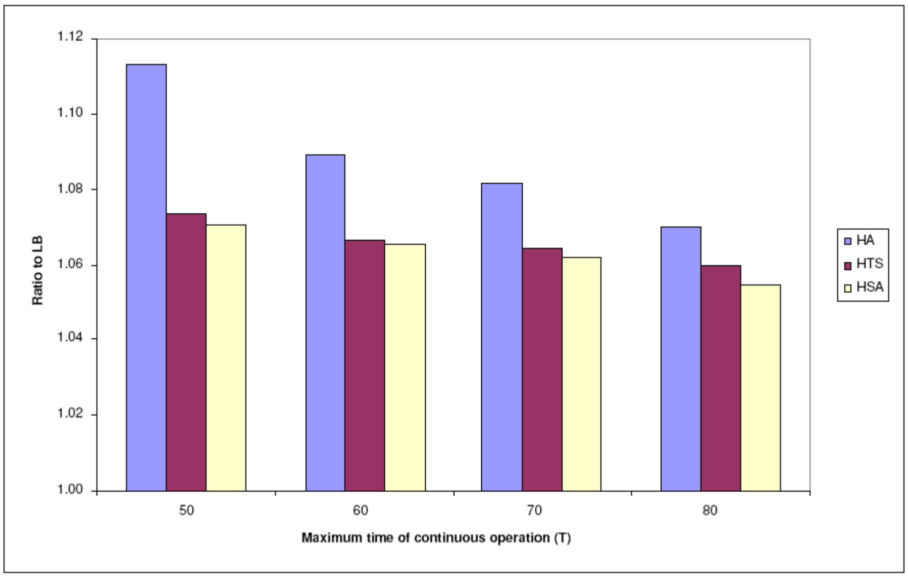

Figure 4. Effect of allowed continuous operation time T.

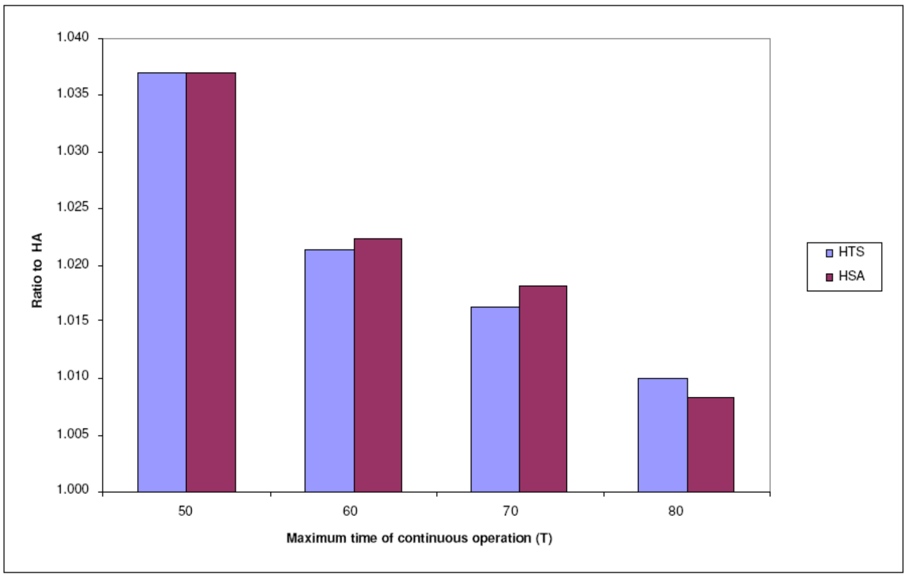

Figure 5. Effect of allowed continuous operation time T on improvement over HA.

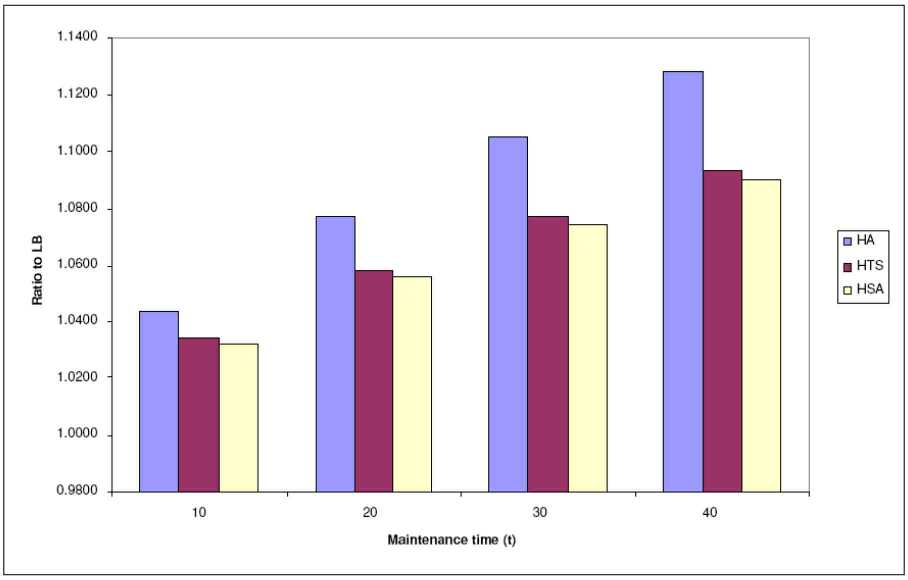

Figure 6. Effect maintenance time on the performance of proposed algorithms.

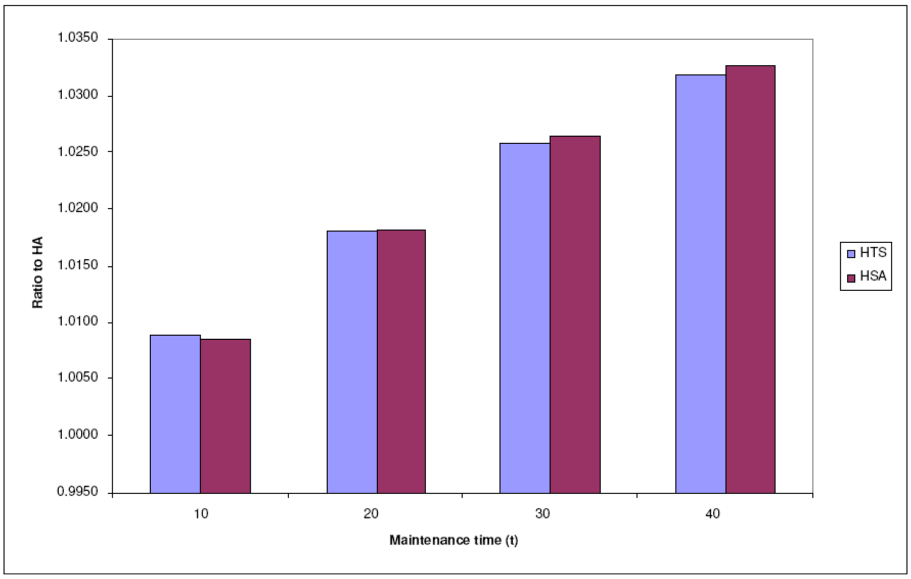

Figure 7. Effect of maintenance time t on improvement over HA.

As mentioned earlier, the proposed HA and lower bound sequence jobs in V-shaped and then maintenance is scheduled on this job sequence. The maintenance requirement increases: with an increase in the job size (*n*); increase in maintenance time (*t*); and with a decrease in maximum allowable time for a continuous operation (*T*). The objective function has contribution both from the jobs and the maintenance operations. With an increase in the maintenance requirement, the contribution to the objective function from maintenance becomes significant. As both the HA and lower bound have fixed V-shaped sequencing for jobs, it is less likely to minimize the contributions from maintenance to the objective function, while it is getting more substantial. Unlike HA and LB, both the HTS and HSA can use other job sequences, not only minimizing the contributions from maintenance operations but also the contributions from the jobs.

6. CONCLUSION AND FUTURE RESEARCH SUGGESTIONS

In this paper, we address the problem of joint scheduling of maintenance operations and jobs on a single machine with an objective to minimize total earliness and tardiness about a common due date. We present some important properties of an optimal schedule. The properties are also used for development of a lower bound estimate and a constructive heuristic to the problem. Two efficient meta-heuristics that make use of the properties developed are also proposed. Numerical experiments with over 3200 problems are carried out, and three major criteria are used to calibrate the performance of the proposed solution methods that include deviation from lower bound, improvement over proposed constructive heuristic and CPU time. Numerical experiments have resulted in the following conclusions:

- Lower bound and constructive heuristics are sensitive to maintenance parameters. They are observed better in performance as the maximum allowable delay between two maintenance operations increases.
- The impact of maintenance time is opposite to the maximum allowable delay between two maintenance operations. The performance of both the lower bound and constructive heuristic improve as maintenance time decreases.
- Embedding the characterization of optimal schedule into Tabu Search and Simulated Annealing algorithms significantly reduces the CPU time. It also helps in the convergence of the algorithms.
- The performance of the Hybrid Tabu Search and Simulated Annealing algorithms improves when compared to the proposed constructive heuristic, as the job size increases.

There are several directions in which this research can be extended. In this study, the common due date is considered unrestrictive, however a restrictive due date or a window due date can also be considered. An assumption in this work is that the machine does not fail. An extension can be to address both problems jointly. A direct but more complicated direction of future work is to consider more than one machines, i.e., joint scheduling of operations and maintenance activities in a flowshop environment. Some other opportunities of research may include the consideration of a stochastic behavior. The processing times and maintenance related parameters can also be considered as random parameters following some probability distribution. The objective can be to minimize the expected value of the objective function. A multi-objective scheduling problem subjected to preventive maintenance and machine failures can also be considered for future work, as most machine scheduling problems need to satisfy more than one criteria.

ACKNOWLEDGMENTS

The first author would like to thank Dr. Ali Akgunduz and Dr. Nadia Bhuiyan for their helpful comments. The second and third authors acknowledge the support of King Fahd University. The authors are also grateful to anonymous referees.

REFERENCES

1. Adzakpa, K.P., Adjallah, K.H., and Lee, J. (2004). A new effective heuristic for the intelligent management of the preventive maintenance tasks of the distributed systems. *Advanced Engineering Informatics*, 17: 151-163.
2. Adzakpa, K.P., Adjallah, K.H., and Yalaoui, F. (2004). Scheduling with tool changes to minimize total completion time: Basic results and spt performance. *Journal of Intelligent Manufacturing*, 15: 131-140.
3. Aggoune, R. (2004). Minimizing the makespan for the flow shop scheduling problem with availability constraints. *European Journal of Operational Research*, 153: 534-543.
4. Akturk, M.S., Ghosh, J.B., and Gunes, E.D. (2003). Scheduling with tool changes to minimize total completion time: A study of heuristics and their performance. *Naval Research Logistics*, 50: 15-30.
5. Bulbul, K., Kaminsky, P., and Yano, C. (2004). Flow shop scheduling with earliness, tardiness, and intermediate inventory holding costs. *Naval Research Logistics*, 51: 407-445.
6. Cassady, C.R. and Kutanoglu, E. (2003). Minimizing job tardiness using integrated preventive maintenance planning and production scheduling. *IIE Transactions*, 35: 503-513.
7. Chen, Z.L. (2004). Simultaneous job scheduling and resource allocation on parallel machines. *Annals of Operations Research*, 129: 135-153.
8. Chen, Z.L. and Powell, W.B. (2003). Exact algorithms for scheduling multiple families of jobs on parallel machines. *Naval Research Logistics*, 50: 823-840.
9. Eglese, R.W. (1990). Simulated annealing: A tool for operational research. *European Journal of Operational Research*, 46: 271-281.
10. Eilon, S. and Chowdhury, I.G. (1977). Minimizing waiting time variance in the single machine problem. *Management Science*, 23: 567-575.
11. Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, 5: 533-549.
12. Glover, F. and Laguna, M. (1997). *Tabu Search*, Kluwer Academic Publishers.
13. Kanet, J.J. (1981). Minimizing the average deviation of the job completion times about a common due date. *Naval Research Logistics*, 28: 643-651.
14. Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P. (1983). Optimization by simulated annealing. *Management Science*, 220: 671-680.
15. Lee, C.Y. and Chen, Z. (2000). Scheduling jobs and maintenance activities on parallel machine. *Naval Research Logistics*, 47: 145-165.
16. Liao, C.J., Shyur, D.L., and Lin, C.H. (2005). Makespan minimization for two parallel machines with an availability constraint. *European Journal of Operational Research*, 160: 445-456.
17. Lorigeon, T., Billaut, J.C., and Bouquard, J.L. (2002). A dynamic programming algorithm for scheduling jobs in a two-machine open shop with an availability constraint. *Journal of the Operational Research Society*, 53: 1239-1246.
18. Lyu, J., Gunasekaran, A. and Ding, J.H. (1996). Simulated annealing algorithm for solving the single machine early/tardy problem. *International Journal of Systems Science*, 27: 605-610.
19. Mazzini, R. and Armentano, V. A. (2001). A heuristic for single machine scheduling with early and tardy costs. *European Journal of Operational Research*, 128: 129-146.
20. Merten, A.G. and Muller, M.E. (1972). Variance minimization in single machine sequencing problems. *Management Science*, 18: 512-528.
21. Qi, X., Chen, T., and Tu, F. (1999). Scheduling the maintenance on a single machine. *Journal of the Operational Research Society*, 50: 1071-1078.
22. Raza, S.A. (2002). *Simultaneous Maintenance and Jobs Scheduling on Single Machine under Various Maintenance Policies*, Master's thesis, King Fahd University of Petroleum and Minerals, Saudi Arabia.
23. Sait, S.M. and Youssef, H. (1999). *Iterative Computer Algorithms with Applications in Engineering*, IEEE Computer Society.
24. Schrage, L. (1975). Minimizing the time-in-system variance for a finite jobset. *Management Science*, 21: 540-543.
25. Sortrakul, N., Nachtmann, H., and Cassady, C.R. (2005). Genetic algorithms for integrated preventive maintenance planning and production scheduling for a single machine. *Computers in Industry*, 56: 161-168.
26. Sundararaghvan, P.S. and Mesbah, U.A. (1984). Minimizing the sum of absolute lateness in single machine and multi-machine scheduling. *Naval Research Logistics*, 31: 325-333.
27. White, S.R. (1984). Concept of scale in simulated annealing. *Proceedings of the IEEE International Conference on Computer Design*, pp. 646-651.