**Journal of Algorithms and Computation**

journal homepage: http://jac.ut.ac.ir

# Minimizing the Number of Tardy Jobs on Single Machine Scheduling with Flexible Maintenance Time

Fatemeh Ganji*[1] and Amir Jamali†[2]

[1,2]Department of Industrial Engineering, Gopayegan University of Technology. Golpayegan, Iran

## ABSTRACT

In this study, single machine scheduling with flexible maintenance is investigated with non-resumable jobs by minimizing the weighted number of tardy jobs. It is assumed that the machine stops for a constant interval time during the scheduling period to perform maintenance. In other words, the starting time of maintenance is the decision variable. By reviewing the literature, we noticed that this problem has not been studied yet. Initially, it is proved that the problem is NP-hard. Then, a mathematical model is proposed and solved by the GAMS software. Because of the long time for solving the problem with an exact method, we develop a heuristic algorithm. To evaluate the efficiency of the

*Keyword:* Scheduling; Single machine; Flexible maintenance; Tardy job.

## ARTICLE INFO

*Article history:*
Received 15, May 2018
Received in revised form 18, October 2018
Accepted 30 November 2018
Available online 30, December 2018

AMS subject Classification: 05C79.

*Corresponding author: F. Ganji. Email: ganji@gut.ac.ir
†ajamali@gut.ac.ir

## 1 Abstract continued

proposed algorithm, 696 test problems with different sizes of the problem in the range from 1 to 2000 jobs, are generated. The computational results demonstrate that the average error of solution is 10.93%.

## 2 Introduction

Nowadays, the scheduling problems have found the wide applications in the production systems. Most of these scheduling problems suppose that the machine must work continuously during the planning horizon [2], while the unavailability periods appear often in the industry due to a machine breakdown (in stochastic case) or a preventive maintenance (in deterministic case) during the scheduling period. Therefore, a more realistic scheduling model can be taken into account in the period called unavailability constraint. This assumption is not true in many production environments. In fact, the machine may be not available during some periods like the maintenance operations [2].

The scheduling problems with availability constraint fall into two general categories based on the condition of unavailability constraint. In fact, it can be fixed or flexible. In the scheduling problem with one or periodic flexible maintenance, it is assumed that the starting time of the maintenance process is a decision variable [4]. In some problems with a flexible maintenance, it is assumed that the maintenance period $[U_n, L_n]$ is previously specified and the maintenance time does not fall outside of the maintenance period ($P_n \leq L_n - U_n$). The time $U_n$ ($L_n$) is as the earliest (latest) time in which the machine maintenance starts (stops). Whereas, there aren't many studies in this field, so this research is focused on flexible unavailability constraint.
Considering the first group of the problems with fixed availability constraint, Kacem and Chu [12] studied the problem by minimizing total weighted completion time with a fixed unavailability constraint (denoted by $1, h_1 || \sum c_i W_i$) and developed three heuristic algorithms for the problem. Kacem [10] showed that the worst case performance of the third heuristic proposed in Kacem and Chu [12] is 2. Kacem et al. [11] developed a mixed integer programming model (MIP), a dynamic programming model and a branch-and-bound algorithm for solving the problem $1, h_1 || \sum c_i w_i$.
Liao and Chen [14] studied the problem by minimizing the number of tardy jobs on a single machine with the periodic maintenance. They proposed a heuristic algorithm with $O(n^2 \sum p_i)$ and a branch-and-bound algorithm for solving the problem.
Considering the second group of the scheduling problems with flexible unavailabil-

ity constraint, Yang et.al [22] proved that the single machine scheduling problem with a flexible maintenance by minimizing the makespan is NP-hard. They provided a heuristic algorithm with complexity $O(n \log n)$ to find a good solution for the problem. Chen [7] proposed two mixed binary integer programming (BIP) models for solving the single machine scheduling problem with a flexible maintenance to minimize the total tardiness (denoted by $1, h_1|f|\sum T_i$), while Chen [5] developed two mixed BIP models for the problem $1, h_1|f|C_{\max}$. Also, Chen [6] developed two mixed BIP models for solving the problem $1, h_1|f|\bar{F}$ with the resumable and non-resumable jobs.

Low et.al [15] studied the single machine scheduling problem with the flexible periodic maintenance to minimize the makespan and developed a heuristic algorithm for this problem. Qi [20] studied $1, h_1|f|\sum C_i$ and $1, h_1|f|L_{\max}$ and showed that these problems are NP-hard. Sbihi and Varnier [21] considered the single machine scheduling problem with several flexible maintenance periods considering maximum allowed continues working time of the machine is previously determined. Graves and Lee [9] considered several single machine scheduling problems with semi-resumable jobs and the flexible maintenance scheduled under situation. Their objective functions are total completion time and maximum lateness. Also, they showed these problems are NP-complete.

Ganji et.al [8] considered the minimizing maximum earliness on single machine problem with flexible maintenance. Mashkani and Moslehi [16] studied the scheduling problem considering minimizing number of tardy jobs and flexible availability constraint.

Few researchers have taken into account the weighted number of tardy jobs. Whereas, this objective function has an extensive application in the industrial environment. Also, this objective has some effects on the external costs related to customer satisfaction. Considering this function, Moore [18] studied the problem $1||\sum U_i$ and provided an algorithm for solving the problem, optimally. Lee [13] considered the problem $1, h_1|r|\sum U_i$ and proved that the problem is NP-hard. In addition, he showed that the algorithm proposed by Moore can obtain the optimal solution for this problem. Also, Lee [13] showed that when the Moore and Hodjson's algorithm is applied for solving the problem $1, h_1|nr|\sum U_i$.

Molaee [17] studied the single machine scheduling problem with a fixed unavailability constraint for minimizing the maximum earliness in addition to the number of tardy jobs (denoted by $1, h_1||E_{\max}$, $1, h_1||\sum U_i$) and proposed a heuristic algorithm and a branch-and-bound method for solving the problem.

In this paper, the single machine scheduling problem with one flexible maintenance period and non-resumable jobs is considered by minimizing the weighted number of the tardy jobs. It is assumed there is a flexible maintenance period that its starting time is as the decision variable. Moreover, the idle time is not allowed.

The rest of this paper is organized as follows: the problem is described in Section 3. Notation and mathematical model is presented in Section 4. Section 5 is aimed at proposing heuristic algorithm. Computational experiments are then given in Section 6 to demonstrate the effectiveness of algorithm, followed by the conclusion in Section 7.

## 3 Preliminary

The problem $1|nr-fa|\sum w_iU_i$ is considered as scheduling $n$ jobs on a single machine with the target of minimizing the weighted number of the tardy jobs. It is assumed that no idle time is allowed on the machine and all jobs are non-resumable and available at zero time. Also, all data are integer. The time $U_n$ ($L_n$) is the earliest (latest) time in which the machine starts (stops) its maintenance. Meanwhile, the period $[U_n, L_n]$ is assumed to be specified in advance and time $P_n$ does not exceed the period (i.e. $P_n \leq L_n - U_n$). The number of the tardy jobs is calculated in a sequence as follows: $N_T = \sum_{i=1}^{n} U_i$, where $\sum U_i$ denotes the number of the tardy jobs. If $C_i > d_i$ ($i = 1, \ldots, n$), then $U_i = 1$; Otherwise, $U_i = 0$. If all jobs are processed before the start of maintenance activity, the problem converts to scheduling the single machine without the availability constraint and its optimal solution is obtained by applying the Moore and Hodjson's algorithm [18].
However, in this paper we assume that, the relation $L_n - P_n < \sum p_i$ is always true. Since the problem $1|nr-fa|\sum w_iU_i$ has not been addressed in the literature, first, we discuss its complexity. Yang et al. [22] considered the problem $1|nr-fa|C_{\max}$ and showed that this problem is NP-hard. Each problem with the objective function $C_{\max}$ can be reduced to $L_{\max}$ and this function also reduces to $\sum U_i$ [19]. Therefore, the complexity of the problem with the objective function $\sum U_i$ is not less than the complexity of $C_{\max}$. Also, the problem $1||\sum w_iU_i$ can reduce to $1||\sum U_i$, as well. Hence, the problem $1|nr-fa|\sum w_iU_i$ is also NP-hard.

## 4 Notations and mathematical modeling

In this section, we define notations and decision variables to model the problem $1|nr-fa|\sum w_iU_i$. Then a mathematical model is proposed. The following notations are used in this section:

### Problem parameters:

$n$: Number of the jobs
$J$: Set of all jobs

$p_j$: Processing time for job $j$
$d_j$: Due date for job $j$
$w_j$: weight of job $j$
$M$: large number
$L_n$: Earliest maintenance starting time for the maintenance activity
$U_n$: Latest maintenance completion time for the maintenance activity
$P_n$: Maintenance time

## Decision variables:

$S_j$: start time of job $j$
$S_n$: start time of maintenance activity
$C_j$: completion time of job $j$
$C_n$: completion time of maintenance activity
$X_{jl}$: 1 if job $l$ scheduled after job $j$, 0, otherwise
$X_{0l}$: 1 if job $l$ scheduled at first position in sequence, 0, otherwise
$X_{j(n+1)}$: 1 if job $j$ scheduled at last position in sequence, 0, otherwise
$Y_j$: 1 if maintenance activity scheduled after job $j$, 0, otherwise
$U_j$: 1 if $C_j > d_j$ , 0, otherwise

## The mathematical model

A mixed integer programming is proposed as follows:

$$
\begin{array}{llll}
\min Z : \sum_{j=1}^{n} w_j U_j & & & (1) \\
st : & & & \\
\sum_{j=1}^{n} x_{jl} = 1 & l = 1, 2, \ldots, n & & (2) \\
\sum_{l=1}^{n+1} x_{jl} = 1 & j = 1, 2, \ldots, n & & (3) \\
\sum_{l=1}^{n} x_{0l} \leq 1 & & & (4) \\
\sum_{j=1}^{n} x_{j(n+1)} = 1 & & & (5) \\
x_{jj} = 0 & j = 1, 2, \ldots, n & & (6) \\
\sum_{j=1}^{n} x_{jl} = \sum_{l=1}^{n+1} x_{jl} & l = 1, 2, \ldots, n & j = 1, 2, \ldots, n & (7) \\
x_{jl} \in \{0, 1\} & j = 0 & l = n + 1 & (8) \\
S_l \geq 0 & l = 1, 2, \ldots, n & & (9) \\
S_l \geq C_j + (x_{jl} - 1)M & \forall j, l & j \neq l & (10) \\
S_n \geq L_n & & & (11) \\
C_n \leq U_n & & & (12) \\
S_n \geq C_j + (Y_j - 1)M & j = 1, 2, \ldots, n & & (13) \\
S_j \geq C_n - Y_j M & j = 1, 2, \ldots, n & & (14) \\
C_j = S_j + P_j & j = 1, 2, \ldots, n & & (15) \\
C_n = S_n + P_n & & & (16) \\
C_j - d_j \geq M U_j & j = 1, 2, \ldots, n & & (17) \\
Y_j, U_j \in \{0, 1\} & j = 1, 2, \ldots, n & & (18)
\end{array}
$$

Objective function (1) minimizes total weight of number of delays. Constraint (2) to (8) shows that schedules are feasible. Constraint (2) and (3) ensure that only one job is assigned to each position of the sequence of jobs. Constraint (4) and (5) ensure that only one assignment is assigned in the initial and final positions of the job sequence. Constraint (6) shows that a job cannot be processed twice. Constraint (7) creates an inconsistent and compatible sequence. Constraint (8) also determine decision variable as a binary variable. Constraint (9) and (10) find starting time of each job. Constraint (9) ensure non-negative time of starting each job. Constraint (10) is also a set of individual constraints, which shows that if job l is assigned after work j, the start time of job l should be at least as long as the completion time of job j. Constraint (11) and (12) ensure that the maintenance operation (net) will be performed within the permitted time interval [Ln, Un]. Constraint (13) implies that if maintenance operation assign after job j starting time is at least as long as completion time of the job. Otherwise constraint will be activated and shows start time of job j after the completion of maintenance operation. Constraint (15) determine completion time of each job. Constraint (16) determine the completion time of the maintenance operation as the sum of the start and processing time. Constraint (17) will be also activated in the event of delay in completion dead line and assign unit penalty to that job. Finally Constraint (18) explains binary type of decision variable

## 5 Heuristic algorithm

In this section, Heuristic algorithm is presented with the name of WSFTA for the problem $1|nr - fa|\sum w_i U_i$, due to the floating of the jobs and maintenance activity.
Before starting the algorithm, some of the symbols and required definitions are explained as follows:
$EJ_j$: Completion time of job $j$
$SJ_j$: Start time of job $j$
$EJ_{net}$: Start time of maintenance activity
$SJ_{net}$: Completion time of maintenance activity
$Block1$: The interval from the first time of scheduling period to the latest time of doing maintenance activity
$Block2$: The interval from the earliest time of doing maintenance to the end of scheduling period
$S_j$: The floating which is defined in block 1 for the job $j$
$S_j$: The floating which is defined in block 2 for the job $j$

The floating of the job $j$ is calculated in various conditions as follows:

- floating of maintenance activity:

$$S = U_n - \sum_{SJ_j \geq L_{net} \ \& \ EJ_j < U_{net}} P_j - ( {}_{SJ_k < L_{net} \ \& \ EJ_k < U_{net}} EJ k - {}_{SJ_k < L_{net} \ \& \ EJ_k < U_{net}} EJ net), S' = 0$$

- floating the job $j$ if $d_j \leq U_n$

$$S = EJ_j - P_j - \sum_{EJ_n < EJ_j} P_n, S' = 0$$

- floating the job $j$ if $d_j < U_n$

$$S = EJ_j - \sum_{EJ_j < EJ_j \ \& \ EJ_j < U_n} P_n - P_{net}$$

$$S' = EJ_j - P_j - L_n - \sum_{EJ_{net} < EJ_{net} \ \& \ EJ_j > L_n \ \& \ SJ_j > EJ_{net}} P_n - ( {}_{SJ_k < L_{net} \ \& \ EJ_k < U_{net}} EJ k - {}_{SJ_k < L_{net} \ \& \ EJ_k < U_{net}} EJ net)$$

## WSFTA Algorithm

**Step 1** Put the maintenance activity at the end of the permitted maintenance period, $[SJ_{net}, EJ_{net}]$. As $U_n = EJ_{net}$, $SJ_{net} = U_n - P_n$, Go to step 2.

**Step 2** if there is a job that $P_j > d_j$, put this job out of the scheduling, Go to step 3.

**Step 3** calculate $r_j$ as $r_j = w_j/P_j$. Sort them in descending order. (A job with more quantity of $r_j$ has higher priority for enter scheduling), go to step 4.

**Step 4** if all of the jobs are in scheduling $(j > n)$ go to step 8. Otherwise choose job $j$ from according priority. Then go to step 5.

**Step 5** put job $j$ at the end of its due date $[SJ_j, EJ_j] = [d_j - P_j, d_j]$. If no job has been assigned in this period, go to step 6; Otherwise go to step 7.

**Step 6** update floating mode of all jobs that are located in scheduling based on the calculated floats-section and then go to step 4.

**Step 7** if there is another activity with a higher priority in which is called job $k$, scheduling will be updated from floating $S$, $S$ as follows:

- If $S'_k \geq P_j$ then using floating $S_k$ should put job $j$ in scheduling.
- If $S'_k \leq P_j \&\ S_k \geq P_j$ then using floating $S_k$ should put job $j$ in scheduling.
- If $S'_k \leq P_j \&\ S_k < P_j$ then Put job $j$ out of the scheduling.

  Update $EJ, SJ$ of all jobs that are located in scheduling. And go to step 6.

**Step 8** if possible shift all the job towards the origin of scheduling and go to step 9.

**Step 9** among jobs which are out of the scheduling, put those with higher $r_j$ at the end of the scheduling.

**Step 10** exit.

## 6 Computational results

In this section, in order to evaluate the performance of the proposed algorithm, a numerical experiment is carried out on a PC with 3GB RAM, Core 2Duo, CPU P8400, and P4 in a Windows 7 environment. We generate the parameter values in set $[3, 7, 8, 11, 15]$. Moreover, 24 series of the instances are generated as follows:
$N \in [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 2000]$
$P_i \in [1, 10]$ using discrete uniform distribution
$d_i \in [(1 - C - Q/2)\sum_{k=1}^{n} P_k, (1 - C + Q/2)\sum_{k=1}^{n} P_k]$ using discrete uniform distribution
$U_n \in \{1/4\sum_{i=1}^{n} P_k, 2/4\sum_{i=1}^{n} P_k, 3/4\sum_{i=1}^{n} P_k\}$
$L_n = U_n + 30$
$P_n \in [1, 15], [16, 30]$ using discrete uniform distribution.
Following the above assumption, we derive 696 test problems and distribute them to each individual group. For solving these test problems, 3600 seconds time limit is considered. It must be mentioned that, from references [1, 6], the processing times of the test problems are generated by the discrete uniform distribution in interval [1, 100].
Table 1,2,3 and 4 present the result of the test problems that they solved by exact and heuristic algorithm (WSFTA). Column "time of GAMS" and "time of WS-FTA" showed the time of solving the test problems by the GAMS AND WSFTA method respectively. For solving these instance problems, 3600 seconds time constraint is applied. The sign "*" in the table means that the associated problems haven't reached optimal solution.
According to the tables 1,2,3 and 4, it is observed that exact method can't solve problems optimally with number of jobs greater than 10 in time limit considered, but heuristic method, can solve the problems optimally greater than 400 jobs in all 24 series and it can solve the problems up to 1000 jobs in some series like series 17 to 24. The cause of this fact, is the maintenance activity assign to the end of its period and more jobs can scheduled in Block1 and the floating S and complexity of problems is decreased. Overall, these results present some good performance of heuristic algorithm in feasible time. Table 5 presents the specifications of series and the error bound of performance between exact and proposed heuristic algorithm in based on the percentage of optimality solved samples. These results show that the maximum error bound between exact and heuristic methods in optimality performance is 10.93.

Table 1: the time of solving problems with GAMS and WSFTA

| | Series1 | | Series2 | | Series3 | | Series4 | | Series5 | | Series6 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| n | time of GAMS | time of WSFTA | time of GAMS | time of WSFTA | time of GAMS | time of WSFTA | time of GAMS | time of WSFTA | time of GAMS | time of WSFTA | time of GAMS | time of WSFTA |
| 1 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 2 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 3 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 4 | 0.01 | 0.00 | 0.02 | 0.00 | 0.01 | 0.00 | 0.01 | 0.00 | 0.01 | 0.00 | 0.02 | 0.00 |
| 5 | 0.21 | 0.00 | 0.15 | 0.00 | 0.19 | 0.00 | 0.13 | 0.00 | 0.13 | 0.00 | 0.16 | 0.00 |
| 6 | 3.44 | 0.00 | 2.80 | 0.00 | 2.87 | 0.00 | 3.75 | 0.00 | 3.73 | 0.00 | 3.66 | 0.00 |
| 7 | 17.80 | 0.00 | 20.18 | 0.00 | 19.68 | 0.00 | 0.00 | 0.00 | 24.84 | 0.00 | 24.22 | 0.00 |
| 8 | 79.31 | 0.00 | 88.46 | 0.00 | 84.04 | 0.00 | 81.51 | 0.00 | 79.76 | 0.00 | 75.65 | 0.00 |
| 9 | 243.36 | 0.00 | 390.56 | 0.00 | 160.62 | 0.00 | 339.80 | 0.00 | 199.89 | 0.00 | 240.89 | 0.00 |
| 10 | 677.93 | 0.00 | * | 0.00 | 409.84 | 0.00 | * | 0.00 | 418.00 | 0.00 | 1688.75 | 0.00 |
| 20 | * | 0.00 | * | 0.00 | * | 0.00 | * | 0.00 | * | 0.00 | * | 0.00 |
| 30 | * | 0.02 | * | 0.01 | * | 0.01 | * | 0.01 | * | 0.01 | * | 0.01 |
| 40 | * | 0.05 | * | 0.04 | * | 0.04 | * | 0.04 | * | 0.03 | * | 0.03 |
| 50 | * | 0.21 | * | 0.14 | * | 0.13 | * | 0.12 | * | 0.12 | * | 0.08 |
| 60 | * | 0.33 | * | 0.24 | * | 0.35 | * | 0.27 | * | 0.27 | * | 0.27 |
| 70 | * | 0.56 | * | 0.43 | * | 0.59 | * | 0.53 | * | 0.42 | * | 0.52 |
| 80 | * | 0.72 | * | 1.28 | * | 1.43 | * | 0.91 | * | 0.98 | * | 0.95 |
| 90 | * | 2.13 | * | 1.76 | * | 1.55 | * | 1.81 | * | 2.05 | * | 1.53 |
| 100 | * | 3.75 | * | 3.75 | * | 4.30 | * | 2.81 | * | 3.20 | * | 2.97 |
| 200 | * | 92.30 | * | 88.49 | * | 89.61 | * | 73.26 | * | 98.59 | * | 83.86 |
| 300 | * | 501.52 | * | 648.41 | * | 410.31 | * | 493.53 | * | 643.40 | * | 660.80 |
| 400 | * | 2773.67 | * | 2707.07 | * | 1706.22 | * | 2265.59 | * | 2574.25 | * | 2697.20 |
| 500 | * | 8433 | * | 5274 | * | 5593 | * | 6072 | * | 7625.00 | * | 7983.00 |
| 600 | * | * | * | * | * | * | * | * | * | * | * | * |
| 700 | * | * | * | * | * | * | * | * | * | * | * | |
| 800 | * | * | * | * | * | * | * | * | * | * | * | |
| 900 | * | * | * | * | * | * | * | * | * | * | * | |
| 1000 | * | * | * | * | * | * | * | * | * | * | * | |
| 2000 | * | * | * | * | * | * | * | * | * | * | * | |

Table 2: the time of solving problems with GAMS and WSFTA(continuing)

| | Series7 | | Series8 | | Series9 | | Series10 | | Series11 | | Series12 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| n | time of GAMS | time of WSFTA | time of GAMS | time of WSFTA | time of GAMS | time of WSFTA | time of GAMS | time of WSFTA | time of GAMS | time of WSFTA | time of GAMS | time of WSFTA |
| 1 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 2 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 3 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 4 | 0.01 | 0.00 | 0.02 | 0.00 | 0.02 | 0.00 | 0.02 | 0.00 | 0.01 | 0.00 | 0.02 | 0.00 |
| 5 | 0.18 | 0.00 | 0.22 | 0.00 | 0.13 | 0.00 | 0.14 | 0.00 | 0.19 | 0.00 | 0.23 | 0.00 |
| 6 | 1.46 | 0.00 | 2.81 | 0.00 | 2.86 | 0.00 | 3.71 | 0.00 | 2.29 | 0.00 | 3.45 | 0.00 |
| 7 | 12.33 | 0.00 | 22.66 | 0.00 | 15.33 | 0.00 | 21.26 | 0.00 | 16.89 | 0.00 | 34.15 | 0.00 |
| 8 | 35.62 | 0.00 | 89.03 | 0.00 | 85.40 | 0.00 | 76.03 | 0.00 | 58.92 | 0.00 | 83.50 | 0.00 |
| 9 | 89.89 | 0.00 | 247.12 | 0.00 | 168.98 | 0.00 | 273.88 | 0.00 | 198.80 | 0.00 | 495.96 | 0.00 |
| 10 | 148.80 | 0.00 | 1841.79 | 0.00 | 545.25 | 0.00 | 1396.85 | 0.00 | 342.28 | 0.00 | 2330.34 | 0.00 |
| 20 | * | 0.00 | * | 0.00 | * | 0.00 | * | 0.00 | * | 0.00 | * | 0.00 |
| 30 | * | 0.01 | * | 0.01 | * | 0.01 | * | 0.01 | * | 0.01 | * | 0.02 |
| 40 | * | 0.02 | * | 0.02 | * | 0.05 | * | 0.03 | * | 0.03 | * | 0.05 |
| 50 | * | 0.08 | * | 0.06 | * | 0.14 | * | 0.15 | * | 0.09 | * | 0.13 |
| 60 | * | 0.18 | * | 0.17 | * | 0.38 | * | 0.38 | * | 0.26 | * | 0.24 |
| 70 | * | 0.48 | * | 0.42 | * | 0.69 | * | 0.43 | * | 0.58 | * | 0.73 |
| 80 | * | 0.71 | * | 0.69 | * | 0.84 | * | 1.42 | * | 0.96 | * | 1.29 |
| 90 | * | 1.62 | * | 1.14 | * | 2.49 | * | 2.23 | * | 1.71 | * | 1.41 |
| 100 | * | 1.73 | * | 2.93 | * | 3.36 | * | 3.15 | * | 3.31 | * | 2.71 |
| 200 | * | 64.20 | * | 59.09 | * | 104.51 | * | 99.06 | * | 76.15 | * | 75.86 |
| 300 | * | 435.12 | * | 385.17 | * | 567.27 | * | 397.28 | * | 366.76 | * | 520.89 |
| 400 | * | 1787.97 | * | 1891.50 | * | 2449.35 | * | 2778.96 | * | 2168.08 | * | 1758.10 |
| 500 | * | 5937.00 | * | 6530.00 | * | 6308.00 | * | 8263.00 | * | 6910.00 | * | 5756.00 |
| 600 | * | * | * | * | * | * | * | * | * | * | * | * |
| 700 | * | * | * | * | * | * | * | * | * | * | * | |
| 800 | * | * | * | * | * | * | * | * | * | * | * | |
| 900 | * | * | * | * | * | * | * | * | * | * | * | |
| 1000 | * | * | * | * | * | * | * | * | * | * | * | |
| 2000 | * | * | * | * | * | * | * | * | * | * | * | |

Table 3: the time of solving problems with GAMS and WSFTA(continuing)

| | Series13 | | Series14 | | Series15 | | Series16 | | Series17 | | Series18 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| n | time of GAMS | time of WSFTA | time of GAMS | time of WSFTA | time of GAMS | time of WSFTA | time of GAMS | time of WSFTA | time of GAMS | time of WSFTA | time of GAMS | time of WSFTA |
| 1 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 2 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 3 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 4 | 0.01 | 0.00 | 0.02 | 0.00 | 0.01 | 0.00 | 0.01 | 0.00 | 0.01 | 0.00 | 0.01 | 0.00 |
| 5 | 0.25 | 0.00 | 0.21 | 0.00 | 0.13 | 0.00 | 0.14 | 0.00 | 0.12 | 0.00 | 0.08 | 0.00 |
| 6 | 2.20 | 0.00 | 1.71 | 0.00 | 2.24 | 0.00 | 1.37 | 0.00 | 1.55 | 0.00 | 1.91 | 0.00 |
| 7 | 14.68 | 0.00 | 19.25 | 0.00 | 19.67 | 0.00 | 20.98 | 0.00 | 11.93 | 0.00 | 12.10 | 0.00 |
| 8 | 48.78 | 0.00 | 76.35 | 0.00 | 39.86 | 0.00 | 69.57 | 0.00 | 61.18 | 0.00 | 65.13 | 0.00 |
| 9 | 110.50 | 0.00 | 326.66 | 0.00 | 75.50 | 0.00 | 162.98 | 0.00 | 145.63 | 0.00 | 277.26 | 0.00 |
| 10 | 233.82 | 0.00 | 919.09 | 0.00 | 125.79 | 0.00 | 624.30 | 0.00 | 241.07 | 0.00 | 717.57 | 0.00 |
| 20 | * | 0.00 | * | 0.00 | * | 0.00 | * | 0.00 | * | 0.00 | * | 0.00 |
| 30 | * | 0.01 | * | 0.01 | * | 0.01 | * | 0.01 | * | 0.00 | * | 0.01 |
| 40 | * | 0.04 | * | 0.04 | * | 0.02 | * | 0.03 | * | 0.01 | * | 0.01 |
| 50 | * | 0.16 | * | 0.11 | * | 0.10 | * | 0.06 | * | 0.02 | * | 0.02 |
| 60 | * | 0.21 | * | 0.24 | * | 0.18 | * | 0.24 | * | 0.04 | * | 0.03 |
| 70 | * | 0.55 | * | 0.50 | * | 0.33 | * | 0.46 | * | 0.06 | * | 0.08 |
| 80 | * | 1.06 | * | 0.90 | * | 0.84 | * | 0.81 | * | 0.12 | * | 0.09 |
| 90 | * | 1.57 | * | 1.68 | * | 1.35 | * | 1.41 | * | 0.23 | * | 0.17 |
| 100 | * | 2.65 | * | 2.60 | * | 2.34 | * | 2.22 | * | 0.22 | * | 0.22 |
| 200 | * | 89.82 | * | 87.98 | * | 73.56 | * | 62.68 | * | 3.55 | * | 3.36 |
| 300 | * | 599.94 | * | 624.02 | * | 384.16 | * | 424.17 | * | 15.15 | * | 15.80 |
| 400 | * | 2606.63 | * | 2661.48 | * | 1836.96 | * | 1795.74 | * | 43.74 | * | 47.63 |
| 500 | * | 7423.00 | * | 7015.00 | * | 5733.00 | * | 8002.00 | * | 39.81 | * | 47.79 |
| 600 | * | * | * | * | * | * | * | * | * | 120.63 | * | 96.08 |
| 700 | * | * | * | * | * | * | * | * | 427.00 | * | 414.32 | |
| 800 | * | * | * | * | * | * | * | * | 788.62 | * | 600.26 | |
| 900 | * | * | * | * | * | * | * | * | 1094.58 | * | 1185.51 | |
| 1000 | * | * | * | * | * | * | * | * | 1452.62 | * | 1833.05 | |
| 2000 | * | * | * | * | * | * | * | * | * | * | * | |

Table 4: the time of solving problems with GAMS and WSFTA(continuing)

| | Series19 | | Series20 | | Series21 | | Series22 | | Series23 | | Series24 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| n | time of GAMS | time of WSFTA | time of GAMS | time of WSFTA | time of GAMS | time of WSFTA | time of GAMS | time of WSFTA | time of GAMS | time of WSFTA | time of GAMS | time of WSFTA |
| 1 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 2 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 3 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 4 | 0.01 | 0.00 | 0.00 | 0.00 | 0.01 | 0.00 | 0.03 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 5 | 0.11 | 0.00 | 0.09 | 0.00 | 0.10 | 0.00 | 0.26 | 0.00 | 0.02 | 0.00 | 0.01 | 0.00 |
| 6 | 1.42 | 0.00 | 1.04 | 0.00 | 1.28 | 0.00 | 3.21 | 0.00 | 1.28 | 0.00 | 3.34 | 0.00 |
| 7 | 10.70 | 0.00 | 11.80 | 0.00 | 12.30 | 0.00 | 16.20 | 0.00 | 11.20 | 0.00 | 10.26 | 0.00 |
| 8 | 63.71 | 0.00 | 76.29 | 0.00 | 73.52 | 0.00 | 83.91 | 0.00 | 62.90 | 0.00 | 72.87 | 0.00 |
| 9 | 109.36 | 0.00 | 402.29 | 0.00 | 163.32 | 0.00 | 486.36 | 0.00 | 109.85 | 0.00 | 113.68 | 0.00 |
| 10 | 3.6.98 | 0.00 | 1276.17 | 0.00 | 284.79 | 0.00 | 2162.56 | 0.00 | 263.21 | 0.00 | 443.80 | 0.00 |
| 20 | * | 0.00 | * | 0.00 | * | 0.00 | * | 0.00 | * | 0.00 | * | 0.00 |
| 30 | * | 0.01 | * | 0.01 | * | 0.01 | * | 0.01 | * | 0.00 | * | 0.00 |
| 40 | * | 0.04 | * | 0.04 | * | 0.01 | * | 0.01 | * | 0.00 | * | 0.00 |
| 50 | * | 0.02 | * | 0.01 | * | 0.02 | * | 0.02 | * | 0.01 | * | 0.02 |
| 60 | * | 0.04 | * | 0.02 | * | 0.04 | * | 0.04 | * | 0.03 | * | 0.03 |
| 70 | * | 0.07 | * | 0.05 | * | 0.08 | * | 0.06 | * | 0.05 | * | 0.06 |
| 80 | * | 0.13 | * | 0.07 | * | 0.13 | * | 0.14 | * | 0.07 | * | 0.10 |
| 90 | * | 0.16 | * | 0.13 | * | 0.16 | * | 0.20 | * | 0.12 | * | 0.12 |
| 100 | * | 0.18 | * | 0.19 | * | 0.24 | * | 0.26 | * | 0.18 | * | 0.20 |
| 200 | * | 3.57 | * | 1.99 | * | 4.13 | * | 4.20 | * | 2.83 | * | 2.36 |
| 300 | * | 13.52 | * | 13.60 | * | 20.80 | * | 21.36 | * | 11.84 | * | 10.18 |
| 400 | * | 31.72 | * | 34.11 | * | 68.50 | * | 78.29 | * | 49.66 | * | 39.59 |
| 500 | * | 27.28 | * | 37.07 | * | 68.24 | * | 63.13 | * | 37.24 | * | 39.18 |
| 600 | * | 96.66 | * | 88.51 | * | 178.38 | * | 174.04 | * | 102.34 | * | 89.66 |
| 700 | * | 327.97 | * | 372.44 | * | 627.21 | * | 674.30 | * | 373.92 | * | 429.14 |
| 800 | * | 530.96 | * | 526.68 | * | 1029.14 | * | 1011.01 | * | 659.40 | * | 618.93 |
| 900 | * | 785.45 | * | 897.99 | * | 1736.85 | * | 1815.68 | * | 902.63 | * | 971.87 |
| 1000 | * | 1331.43 | * | 1022.05 | * | 3345.23 | * | 2648.06 | * | 1566.21 | * | 1433.76 |
| 2000 | * | * | * | * | * | * | * | * | * | * | * | |

Table 5: Specification of series and error bound of WSFTA

| *Series* | $d$ | $C$ | $Q$ | $P_n$ | Average error of optimal solution between exact & heuristic algorithms (%) |
|---|---|---|---|---|---|
| 1 | $1/4\sum P_j$ | 0.2 | 0.2 | $[1,15]$ | 9.61 |
| 2 | $1/4\sum P_j$ | 0.2 | 0.2 | $[16,30]$ | 20.24 |
| 3 | $1/4\sum P_j$ | 0.2 | 0.6 | $[1,15]$ | 5.51 |
| 4 | $1/4\sum P_j$ | 0.2 | 0.2 | $[16,30]$ | 5.30 |
| 5 | $1/4\sum P_j$ | 0.6 | 0.2 | $[1,15]$ | 4.38 |
| 6 | $1/4\sum P_j$ | 0.6 | 0.2 | $[16,30]$ | 6.43 |
| 7 | $1/4\sum P_j$ | 0.6 | 0.6 | $[1,15]$ | 5.20 |
| 8 | $2/4\sum P_j$ | 0.2 | 0.2 | $[16,30]$ | 5.79 |
| 9 | $2/4\sum P_j$ | 0.2 | 0.2 | $[1,15]$ | 23.35 |
| 10 | $2/4\sum P_j$ | 0.2 | 0.2 | $[16,30]$ | 15.47 |
| 11 | $2/4\sum P_j$ | 0.2 | 0.6 | $[1,15]$ | 8.82 |
| 12 | $2/4\sum P_j$ | 0.2 | 0.2 | $[16,30]$ | 6.80 |
| 13 | $2/4\sum P_j$ | 0.6 | 0.2 | $[1,15]$ | 21.17 |
| 14 | $2/4\sum P_j$ | 0.6 | 0.2 | $[16,30]$ | 8.46 |
| 15 | $2/4\sum P_j$ | 0.6 | 0.6 | $[1,15]$ | 5.92 |
| 16 | $2/4\sum P_j$ | 0.2 | 0.2 | $[16,30]$ | 8.42 |
| 17 | $3/4\sum P_j$ | 0.2 | 0.2 | $[1,15]$ | 9.23 |
| 18 | $3/4\sum P_j$ | 0.2 | 0.2 | $[16,30]$ | 24.09 |
| 19 | $3/4\sum P_j$ | 0.2 | 0.6 | $[1,15]$ | 9.01 |
| 20 | $3/4\sum P_j$ | 0.2 | 0.2 | $[16,30]$ | 16.49 |
| 21 | $3/4\sum P_j$ | 0.6 | 0.2 | $[1,15]$ | 18.32 |
| 22 | $3/4\sum P_j$ | 0.6 | 0.2 | $[16,30]$ | 11.82 |
| 23 | $3/4\sum P_j$ | 0.6 | 0.6 | $[1,15]$ | 5.05 |
| 24 | $3/4\sum P_j$ | 0.2 | 0.2 | $[16,30]$ | 7.53 |

## 7 Conclusion

In this study, single machine scheduling with flexible maintenance is considered to minimize the weighted number of tardy jobs. The starting time of maintenance is the decision variable and, initially, it is proved that the problem is NP-hard. Then, a mathematical model is proposed and solved by the GAMS software. For improving the time of solving the problems, a heuristic algorithm (WSFTA) is developed. To evaluate the efficiency of the proposed algorithms, 696 test problems with different sizes of the problem in the range from 1 to 2000 jobs, are generated. The computational results demonstrate that the number of problems that are solved optimally using GAMS isn't up to 10 jobs but heuristic algorithm can solve the problems up to 1000 jobs with maximum error bound 10.93% in optimality.

## Acknowledgments

The author is supported by the Department of Industrial Engineering at the Golpayegan University of Technology and her special thanks go to the Department for providing all necessary facilities available to her for successfully conducting this research.

## References

[1] Adiri, I., Bruno, J., Frostig, E., Rinnoy Kan, A.H.G., Single machine flow-time scheduling with a single breakdown, Acta Information, **26** (1989), 679–696.

[2] Baker, K.R., Introduction to sequencing and scheduling, John Wily, New York, (1974).

[3] Breit, J., Improved approximation for non-preemtive single machine flow-time scheduling with an availability constraint, European Journal of Operational Research, **183**, (2007), 516–524.

[4] Chen, J.S., Optimization models for the machine scheduling problem with a single flexible maintenance activity. Engineering Optimization **38** (2006), 53–71.

[5] Chen W.J, minimizing number of tardy jobs on a single machine subject to periodic maintenance, Omega, **37** (2009), 591–599.

[6] Chen, J.S., Using integer programming to solve the machine scheduling problem with a flexible maintenance activity, Journal of Statistics & Management Systems, **9**, no. 1 (2006), 87-104.

[7] Chen J.S, Scheduling of nonresumable jobs and flexible maintenance activities on a single machine to minimize makespan, European Journal of Operational Research **190**, no. 1 (2008), 90–102.

[8] Ganji, F., Moslehi, G, Ghalebsaz Jeddi, B., Minimizing maximum earliness in single-machine scheduling with flexible maintenance time, Scientia Iranica **24**, no. 4 (2017), 2082–2094.

[9] Graves, G.H., and Lee, C.Y.:Scheduling maintenance and semiresumable jobs on a single machine, Naval Research Logistics (NRL), **46** no. 7 (1999), 845–863.

[10] Kacem, I., Approximation algorithm for the weighted flow-time minimization on a single machine with a fixed non-availability interval, Computers & Industrial Engineering **54** (2008), 401–410.

[11] Kacem, I., Chu, C., Souissi, A., Single-machine scheduling with an availability constraint to minimize the weighted sum of the completion times, Computers & Operations Research, **35** (2008), 827–844.

[12] Kacem, I., Chu, C., Efficient branch-and-bound algorithm for minimizing the weighted sum of completion times on a single machine with one availability constraint, Int. J. Production Economics, **112** (2008), 138–150.

[13] Lee, C.Y., Machine scheduling with an availability constraint, Journal of Global Optimization, **9**, no. 3 (1996), 395–416.

[14] Liao, C.J, Chen, W.J, Single-machine scheduling with periodic maintenance and nonresumable jobs, Computers & Operations Research, **30** (2003), 1335–1347.

[15] Low, C., Ji, M., Hsu, C.J., and Su, C.T., Minimizing the makespan in a single machine scheduling problems with flexible and periodic maintenance, Applied Mathematical Modeling, **34**, no. 2 (2009), 334–342.

[16] Mashkani, O., Moslehi, Gh., Minimizing the Number of Tardy Jobs in the Single Machine Scheduling Problem under Bimodal Flexible and Periodic Availability Constraints, International Journal of Industrial Engineering & Production Research, **29**, no. 1, (2018), 15–34.

[17] Molaee, E., Minimizing maximum earliness and number of tardy jobs in the single machine scheduling problem, Computers & Mathematics with Applications, **60** (2010), 2909–2919.

[18] Moore, J.M., An n job, one machine sequencing algorithm for minimizing the number of late jobs, Management Science, **15**, no. 1 (1968), 102–109.

[19] Pinedo, M.L., Scheduling: theory, algorithms, and systems, Springer Verlag, (2008).

[20] Qi, X., A note on worst-case performance of heuristics for maintenance scheduling problems, Discrete Applied Mathematics, **155**, no. 3 (2007), 416–422.

[21] Sbihi, M., and Varnier, C., Single-machine scheduling with periodic and flexible periodic maintenance to minimize maximum tardiness, Computers & Industrial Engineering, **55**, no. 4 (2008), 830–840.

[22] Yang, D.L., Hung, C.L., Hsu, C.J., Chen, M.S., Minimizing the makespan in a single machine scheduling problem with a flexible maintenance, J. Chinese Inst. Insyst. Eng. **19** (2002), 63–66.