International Journal of Industrial Engineering & Production Research (2018)

March 2018, Volume 29, Number 1
pp. 15- 34

DOI: 10.22068/ijiepr.29.1.15 http://IJIEPR.iust.ac.ir/

Minimizing the Number of Tardy Jobs in the Single Machine Scheduling Problem under Bimodal Flexible and Periodic Availability Constraints

Omolbanin Mashkani & Ghasem Moslehi*

Omolbanin Mashkani, *Department of Industrial and Systems Engineering, Isfahan University of Technology,*
Ghasem Moslehi, *Department of Industrial and Systems Engineering, Isfahan University of Technology,*

KEYWORDS

single machine,
flexible periodic availability constraints,
bimodal,
tardy jobs,
branch-and-bound.

ABSTRACT

In single machine scheduling problems with availability constraints, machines are not available for one or more periods of time. In this paper, a single machine scheduling problem with flexible and periodic availability constraints is investigated. In this problem, the maximum continuous working time for each machine can increase in a stepwise manner and can have two different values. In addition, the duration of unavailability for each period depends on the maximum continuous working time of the machine in that same period, again with two different values. The objective is to minimize the number of tardy jobs. In the first stage, the complexity of the problem is investigated; then, a binary integer programming model, a heuristic algorithm, and a branch-and-bound algorithm are proposed in the second stage. Computational results of solving 1680 sample problems indicate that the branch-and-bound algorithm is capable of not only solving problems up to 20 jobs, but also of optimally solving 94.76% of the total number of problems. Based on the computational results, a mean average error of 2% is obtained for the heuristic algorithm.

1. Introduction

When dealing with scheduling problems with availability constraints, at least one machine is unavailable for one or more periods of time. This may result from a number of causes including breakdowns, maintenance, tool change, or overlapped planning horizons. Many researchers have investigated the effects of availability constraints on scheduling problems in various areas such as production planning, cost analysis, and other industrial applications (for example [1], [2]). Depending on how the beginning of unavailability period is determined, scheduling problems with availability constraints are divided into two groups: fixed and flexible. Moreover, if fixed (flexible) unavailability periods occur at predetermined intervals during the planning horizon, then such constraints are called periodic fixed (flexible) availability constraints. In problems with fixed availability constraints, the start time, finish time, and duration of the unavailability periods are all predetermined. In problems with flexible availability constraints, the durations of the unavailability periods are predetermined. However, the start times of these periods are decision variables. In the latter category, in some problems, unavailability occurs

* **Corresponding author:** ***Ghasem Moslehi***

Email: moslehi@cc.iut.ac.ir
Received 26 August 2016; revised 4 December 2017; accepted 2 January 2018

at the predetermined intervals, while, in some others, the maximum continuous working time of the machine is constant, and the
start time of the unavailability periods may be equal to or less than this maximum value. On the other hand, if the duration of unavailability periods is a function of the conditions of the machine or the operator, then the availability constraint is called "variable unavailability".

References [3- 6] may be consulted for more details on scheduling problems with fixed availability constraints. A number of studies have been reported on flexible availability constraints. Yang et al. [7] were the first to introduce the single machine scheduling problem with one flexible unavailability period within a given interval. In this study, makespan minimization with no preemption, symbolized by $1|nr-fa|C_{\max}$, was studied. While the problem was proved to be NP-hard, a heuristic algorithm was proposed for solving the problem. Chen [8] studied the same problem reported in [7] considering flow time. He proposed two binary integer programming models to solve problems up to 10 jobs. In a later study, Chen [9] dealt with the same problem with the objective of total tardiness and developed two binary integer programming models capable of solving problems up to 8 jobs. More recently, Chen [10] studied the single machine scheduling problem with flexible periodic availability constraints to minimize the makespan, which is denoted by $1|nr-fpa|C_{\max}$. He proposed one heuristic algorithm and one mixed integer linear programming model that was capable of solving problems up to 100 jobs. To address the single machine scheduling problem with a flexible availability constraint and with the objective of minimizing total completion time, Yang et al. [11] proposed one heuristic algorithm, one dynamic programming algorithm, and one branch-and-bound algorithm capable of handling problems up to 400 jobs. Ganji et al. [12] investigated the single machine scheduling problem with a flexible unavailability period to minimize the maximum earliness. To find the optimal solution to each of these problems, they proposed a heuristic algorithm and a branch-and-bound algorithm capable of dealing with 14000 jobs.

Scheduling problems with flexible availability constraints, where maximum continuous working time of the machine has a predetermined value, was first introduced by Qi et al. [13]. They assumed predetermined values for both maximum continuous working time and duration of unavailability constraints. To minimize the total completion time, they proposed three heuristic algorithms and one branch-and-bound algorithm capable of solving problems up to 20 jobs. Graves and Lee [14] considered the same problem reported in [13] in which jobs are semi-resumable and the objective functions are maximum lateness and weighted sum of jobs' completion time. For solving these problems, they proposed two dynamic programming algorithms.

Moshiove and Sarig [15] dealt with the single machine scheduling problem with one flexible availability constraint with the objective of total weighted completion time. They developed a dynamic programming algorithm and a heuristic algorithm for solving the problem. Low et al. [16] introduced five heuristic algorithms for solving the single machine scheduling problem with flexible and periodic availability constraints and the objective of minimizing the makespan.

Sbihi and Varnier [17] analyzed the single machine scheduling problem with flexible and periodic availability constraints and the objective of maximum tardiness. They proposed one heuristic and one branch-and-bound algorithm capable of handling 15 jobs.

In the studies performed in the field of scheduling problems with variable availability constraints, duration of the unavailability periods is a linear or exponential function of continuous working time of the machine, while the start time of unavailability constraints and their durations are determined by such factors as deterioration during processing jobs or tool aging. These constraints have been investigated by [18- 20].

Mashkani and Moslehi [21] discussed a new single machine scheduling problem under bimodal flexible periodic availability constraints. The objective function was minimizing the total completion time. They assumed that, in each period, maximum continuous working time of the machine can have two fixed and predetermined values, and the unavailability start time is a decision variable depending on the maximum continuous working time of the machine.

They supposed that the duration of unavailability constraint in each period can have two different values. Hence, in each period, if any increase in the continuous working time of machine is required to improve the objective function, the duration of unavailability at the end of that period will increase to a constant value. In addition, preemption is not allowed.

It should be noted that, in real world, it is possible to maintain and service a machine (such as the milling machine) after a determined continuous working time. If maintenance activities cannot be performed for whatever reason, then the continuous working time of the machine will increase. Hence, it is natural that the machine needs more time for service, oiling, scrubbing, etc., and, thereby, enhancing the unavailability of the machine [21].

Mashkani and Moslehi [21] presented a heuristic and a branch-and-bound algorithm to solve the problem. Their proposed branch-and-bound algorithm was able to cope with problems up to 22 jobs. They were the first ones who investigated the single machine scheduling problem with flexible periodic availability constraints and two values for the continuous working time of the machine in each period.

In this paper, the objective function is minimizing the number of tardy jobs under the same circumstances, supposed by Mashkani and Moslehi [21]. In Section 2, problem definition, symbols and notations are presented; in addition, the problem complexity is investigated. Section 3 presents a developed binary integer programming model, while Section 4 presents relevant theorems and lemmas. In Section 5, a heuristic algorithm for obtaining the near optimal solution is proposed; in Section 6, details of the proposed branch-and-bound algorithm are discussed. In Section 7, a generalized form of the problem with several kinds of unavailability periods is introduced. Sections 8 and 9 are devoted to computational results and concluding remarks, respectively.

2. Problem Definition and Its Complexity

A set of n independent jobs $\{J_1, J_2, ..., J_n\}$ is available at time zero to be processed on a single machine. Setup times are independent of sequences of jobs; they, however, are part of the processing time for each job. In each period, the maximum continuous working time of the machine can have either of two values T_1 and T_2 ($T_2 > T_1$), and the duration of each unavailability period, which depends on continuous working time of the machine, can have either of the two values W_1 and W_2 ($W_2 > W_1$), respectively. Since maximum continuous working time of the machine and duration of the unavailability period can have two different values in each period, such constraints are called "the bimodal flexible and periodic availability constraints" as depicted in Fig. 1. Jobs scheduled between each of two unavailability periods are called a "batch". In Fig. 1, values q_1 to q_4 represent the total processing time of the scheduled jobs for batches 1 to 4, respectively. This problem is symbolized as $1|nr\text{-}fpa,bm|\sum_{i=1}^{n} U_i$.

In this paper, the following notations are used:

n: Number of jobs.

p_i : Processing time of job J_i for $i=1, 2, ..., n$.

d_i : Due date of job J_i for $i=1, 2, ..., n$.

C_i: Completion time of job J_i for $i=1, 2, ..., n$.

K : Number of batches needed for scheduling all jobs.

K^* : Number of batches in the optimal schedule.

B_k : the k^{th} batch for $k= 1, 2, ..., K$.

n_k : Number of scheduled jobs in the kth batch such that $\sum_{k=1}^{K} n_k = n$ for $k = 1, 2, ..., K$

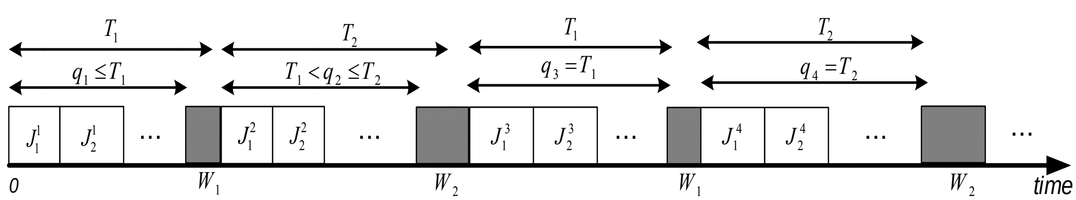

Fig. 1. Single machine scheduling problem under bimodal flexible and periodic availability constraints

J_i^k : Job J_i which is scheduled in the k^{th} batch for $i=1, 2, ..., n$ and $k= 1, 2, ..., K$.

C_i^k : Completion time of job J_i which is scheduled in the kth batch for $i=1, 2, ..., n$ and $k= 1, 2, ..., K$.

q_k : Total processing time of the jobs scheduled in the kth batch for $k= 1, 2, ..., K$.

T_p : Maximum continuous working time of machine for $p=1, 2$.

W_p: Duration of unavailability corresponding to the maximum continuous working time of the machine for $p=1, 2$.

σ : Sequence of scheduled jobs.

σ' : Sequence of unscheduled jobs.

$C(\sigma)$: Completion time for partial sequence σ .

T_i : Tardiness of job J_i calculated through Eq. (1).

$$T_i = max\{0, C_i - d_i\} \qquad i = 1, 2, ..., n \tag{1}$$

U_i : A binary variable. It is 1 if job J_i is tardy; otherwise, it is zero for $i=1, 2, ..., n$.

$\sum_{i=1}^{n} U_i$: Number of tardy jobs.

S_k: Starting time of the kth batch for $k= 1, 2, ..., K$.

Qi et al. [13] showed that the single machine scheduling problem with flexible and periodic availability constraints and with the objective function of minimizing maximum completion time of jobs, where the maximum continuous working time of the machine in each period has a predetermined value, is *NP-hard*. In the following, using Theorem 1, it is shown that problem $1|nr\text{-}fpa,bm|\sum_{i=1}^{n} U_i$ is also *NP-hard*.

Theorem 1: The problem $1|nr\text{-}fpa,bm|\sum_{i=1}^{n} U_i$ is *NP-hard*.

Proof: In problem $1|nr\text{-}fpa,bm|\sum_{i=1}^{n} U_i$, assume that T_1 equals T_2 and W_1 equals W_2.

Then, this problem transforms to the problem $1|nr\text{-}fpa|\sum_{i=1}^{n} U_i$ in which maximum continuous working time of the machine has a predetermined value. Therefore, the complexity of the problem $1|nr\text{-}fpa,bm|\sum_{i=1}^{n} U_i$ is at least as equal as the problem $1|nr\text{-}fpa|\sum_{i=1}^{n} U_i$. Since the problem $1|nr-fpa|C_{\max}$ is *NP-hard* [13], the problem $1|nr\text{-}fpa|\sum_{i=1}^{n} U_i$ is also *NP-hard.* As a result, the problem $1|nr\text{-}fpa,bm|\sum_{i=1}^{n} U_i$ has at least the complexity of an *NP-hard* problem.

3. Mathematical Modeling

In this Section, a mathematical model is developed for the problem $1|nr\text{-}fpa,bm|\sum_{i=1}^{n} U_i$. To do so, the following variables are defined:

x_i^k : A binary variable. If job J_i is scheduled in batch k, its value will be 1; otherwise, it will be zero for $i=1, 2, ..., n$ and $k=1, 2, ..., K$.

y_p^k : A binary variable equal to 1 if the maximum continuous working time of the machine in the kth batch equals T_p; otherwise, it is zero for $p=1, 2$ and $k=1, 2, ..., K$.

h_{ij} : This is a parameter. If job J_i is located before job J_i in the EDD sequence (arranging the jobs in the non-decreasing order of their normal due dates), then its value will be 1; otherwise, it will be zero for $i,j=1, 2, ..., n$ and $j \neq i$.

Mathematical model:

$$\textit{minimize} \quad \sum_{i=1}^{n} U_i \tag{2}$$

Subject to:

$$\sum_{k=1}^{K} x_i^k = 1 \qquad i = 1, 2, ..., n \tag{3}$$

$$\sum_{i=1}^{n} p_i . x_i^k \le \sum_{p=1}^{2} T_p . y_p^k \qquad k = 1, 2, ..., K \tag{4}$$

$$S_1 = 0 \tag{5}$$

$$S_k = \sum_{i=1}^{n} \sum_{r=1}^{k-1} p_i . x_i^r + \sum_{r=1}^{k-1} \sum_{p=1}^{2} W_p . y_p^r \qquad k = 2, 3, ..., K \tag{6}$$

$$C_i^k = S_k + p_i + \sum_{j \neq i} p_i . h_{ij} . x_i^k \qquad i = 1, 2, ..., n; k = 2, 3, ..., K \tag{7}$$

$$C_i + M(1 - x_i^k) \ge C_i^k \qquad i = 1, 2, ..., n; \quad k = 2, 3, ..., K \tag{8}$$

$$C_i - d_i \le M . U_i \qquad i = 1, 2, ..., n \tag{9}$$

$$\sum_{p=1}^{2} y_p^k = 1 \qquad k = 1, 2, ..., K \tag{10}$$

$$x_i^k, y_p^k, U_i \in \{0, 1\} \qquad p = 1, 2; i = 1, 2, ..., n; k = 1, 2, ..., K \tag{11}$$

Equation (2) represents the objective function for the number of tardy jobs. According to Eq. (3), each job can be scheduled just in one batch. Equation (4) determines that the total processing times of the scheduled jobs in the kth batch are less than or equal to the maximum continuous working time of the machine. Start time for the first batch is zero, as in Eq. (5). Eq. (6) calculates the start time of the kth batch for $k=2, 3, ..., K$.

In Eq. (7), the completion time of job J_i in the kth batch is represented and the completion time of job J_i is restricted in Eq. (8). Equation (9) determines tardiness of job J_i. In Eqs. (8) and (9), M is a large positive number. According to Eq. (10), the duration of unavailability after the continuous working time of the machine in each batch can have only values W_1 and W_2. Equation (11) deals with the binary characteristics of the decision variables.

In this model, the number of variable x_i^k is $n \times K$, the number of variable y_p^k is $2 \times K$, and the number of variable U_i is n; therefore, the total number of variables in the model will be $n \times K + 2 \times K + n$.

Equation (3) of the model should be feasible for n variables and Eq. (4) and (10) should also be held for K batches. Eventually, since Eqs. (5), (6), (7), (8), and (9) can be merged, then $n \times K$ equations will be feasible by implementing these constraints.

Hence, the number of constraints in the model will be $n+2 \times K+n \times K$. Note that parameter h_{ij} should be determined by EDD arrangement of jobs.

The mathematical model of $1|nr\text{-}fpa,bm|\sum_{i=1}^{n} U_i$ problem is solved by *CPLEX* of the *GAMS* software and has been capable of solving problems up to 12 jobs in less than 3600 seconds.

4. Lemmas and Theorems

In this Section, a number of lemmas and theorems are introduced and proved based on specifications of the problem $1|nr\text{-}fpa,bm|\sum_{i=1}^{n} U_i$. First, the relevant theorems to find a lower bound for the problem $1\left|nr\text{-}fpa,bm\right|\sum_{i=1}^{n}U_i$ are analyzed.

Lee [22] showed that the optimal solution to the problem $1\left|r-a\right|\sum_{i=1}^{n}U_i$ can be calculated by Moore and Hudgson algorithm. In his method, jobs are initially scheduled using Moore and Hudgson algorithm. Then, the duration of the unavailability period is added to the completion time of those jobs, which are scheduled after the unavailability period. Accordingly, in Theorem 2, a lower bound for the problem $1\left|nr\text{-}fpa,bm\right|\sum_{i=1}^{n}U_i$ is proposed.

<u>Theorem 2:</u> The optimal solution to the problem $1\left|\,\right|\sum_{i=1}^{n}U_i$ is a lower bound for the problem $1\left|nr\text{-}fpa,bm\right|\sum_{i=1}^{n}U_i$.

Proof: Since the solution to = problem $1\left|\,\right|\sum_{i=1}^{n}U_i$ is a lower bound for problem $1\left|nr-a\right|\sum_{i=1}^{n}U_i$ as shown in [22], the number of tardy jobs will definitely not decrease when unavailability periods are added. Therefore, it can be concluded that the optimal solution to problem $1\left|\,\right|\sum_{i=1}^{n}U_i$ is a lower bound for problem $1\left|nr\text{-}fpa,bm\right|\sum_{i=1}^{n}U_i$.

<u>Corollary 1</u>: In problem $1\left|nr\text{-}fpa,bm\right|\sum_{i=1}^{n}U_i$, it is assumed that σ is the partial sequence of scheduled jobs and σ' is the set of unscheduled jobs. Using Theorem 2, if the jobs in set σ' are scheduled by Moore and Hudgson algorithm without availability constraints and the number of tardy jobs has consequently become equal to $\sum_{i\in\sigma'}U_i$, then $\sum_{i\in\sigma}U_i+\sum_{i\in\sigma'}U_i$ will be a lower bound for the partial sequence σ .

To find another lower bound for problem $1\left|nr\text{-}fpa,bm\right|\sum_{i=1}^{n}U_i$, first, a lower bound for the single machine scheduling problem with flexible and periodic availability constraints is proposed where the maximum continuous working time of the machine in each period is T_1 and duration of unavailability is W_1. When preemption is allowed, this problem is symbolized by $1\left|r-fpa\right|\sum_{i=1}^{n}U_i$ and when preemption is not allowed, the problem is denoted by $1\left|nr-fpa\right|\sum_{i=1}^{n}U_i$. To solve problem $1\left|r-fpa\right|\sum_{i=1}^{n}U_i$, algorithm H_1 can be used.

Algorithm H_1

Step 0. Arrange the jobs on the basis of EDD rule and name them in the same order. Set $\sigma=\varnothing$, $\pi=\varnothing$, and $\sigma'=\{J_1,J_2,\dots,J_n\}$. Set parameters i,δ,C_i and q_k equal to zero. Set $k=1$ and go to Step 1.

Step 1. Set $i=i+1$. If $i=n+1$, then go to Step 4; otherwise, calculate $q_k=q_k+p_i$. Then, omit job J_i from set σ' and put it in set σ . If $q_k\le T_1$, then set $C_i=C_{i-1}+p_i$ and go to Step 2; otherwise, set $\delta=T_1-q_k$, $k=k+1$, $C_i=C_{i-1}+p_i+W_1$, and $q_k=p_i-\delta$; then, go to Step 1.

Step 2. Calculate the tardiness of job J_i. If it is not tardy, then go to Step 1; otherwise, go to Step 3.

Step 3. If job J_i is the first tardy job in the sequence, then choose a job with the largest processing time from set σ. Omit this job from set σ and put it in set π. Go to Step 1.

Step 4. Schedule the jobs in set π at the end of the sequence in an arbitrary order.

Algorithm H_1 operates similarly to Moore and Hudgson algorithm, yet only with the following slight difference. If the total continuous working time of the machine in each period exceeds a predetermined value (which is T_1), then the operation is cut and an unavailability period is implemented before further processing continues. As a result, the time complexity of algorithm H_1 will be $O(n\log n)$.

Since the optimal solution to problem $1\left|r-a\right|\sum_{i=1}^{n}U_i$ is obtained from Moore and Hudgson algorithm [22], in Theorem 3, it is shown that algorithm H_1 provides the optimal solution to problem $1\left|r-fpa\right|\sum_{i=1}^{n}U_i$. Note that, in problem $1\left|r-fpa\right|\sum_{i=1}^{n}U_i$, the duration of each unavailability period is W_1 and the time distance between two consecutive unavailability periods is exactly T_1.

<u>Theorem 3:</u> The solution yielded by algorithm H_1 is the optimal solution to problem $1\left|r\text{-}fpa\right|\sum_{i=1}^{n}U_i$.

Proof: Suppose that, in sequence S, job J_i is scheduled in batch B_k and all jobs before job J_i are not tardy. In this sequence which is displayed in Fig. 2, the set of jobs φ is scheduled from the first batch to batch B_{k-1} . If the jobs in set φ are scheduled without unavailability and if the lengths of the entire unavailability periods are summed up and scheduled at the end of this set, then sequence S' is obtained as shown in Fig. 3.

In sequence S' , unlike in sequence S, the jobs in set φ will remain un-tardy whose completion time has not increased. Meanwhile, since the processing time and lengths of unavailability periods are fixed, the completion time of the jobs in both sequences S and S' , placed before job J_i in batch

B_k , will not change; as a result, they will remain un-tardy. Hence, our problem transforms to problem $1|r-a|\sum_{i=1}^{n} U_i$ in which the start time of the unavailability period equals the total processing time of the jobs in set φ and the length of their unavailability period is $(k-1)\times W_1$. Since the optimal solution to problem $1|r-a|\sum_{i=1}^{n} U_i$ is obtained from Moore and Hudgson algorithm [22], it may be concluded that the optimal solution to problem $1|r-fpa|\sum_{i=1}^{n} U_i$ can also be obtained from the same algorithm. Note that, in the optimal solutions to both problems $1|r-a|\sum_{i=1}^{n} U_i$ and $1|r-fpa|\sum_{i=1}^{n} U_i$, tardy jobs are scheduled at the end of the sequence with an arbitrary order.

In Theorem 4, a lower bound is proposed for problem $1|nr-fpa,bm|\sum_{i=1}^{n} U_i$.

<u>Theorem 4:</u> In problem $1|r\text{-}fpa|\sum_{i=1}^{n} U_i$, it is assumed that the maximum continuous working time of machine is T_2 and unavailability duration is W_1 for each period. Hence, the solution to this problem by H_1 algorithm is a lower bound for the problem $1|nr\text{-}fpa,bm|\sum_{i=1}^{n} U_i$.

Proof: According to Theorem 3, the solution to the problem $1|r-fpa|\sum_{i=1}^{n} U_i$ is obtained from algorithm H_1. This value is denoted by F_1^*. Suppose that F_2^* is the optimal solution to the scheduling problem with bimodal flexible and periodic availability constraints where preemption is allowed. This problem is denoted by $1|r-fpa,bm|\sum_{i=1}^{n} U_i$. Since the maximum continuous working time of the machine in problem $1|r-fpa|\sum_{i=1}^{n} U_i$ is always larger than or equal to its corresponding value for problem $1|r-fpa,bm|\sum_{i=1}^{n} U_i$ and as the unavailability period is always shorter than or equal to its corresponding value in problem $1|r-fpa,bm|\sum_{i=1}^{n} U_i$, then relation $F_2^* \geq F_1^*$ is always true. On the other hand, suppose that F^* is the objective function for the problem $1|nr-fpa,bm|\sum_{i=1}^{n} U_i$. It is apparent that F^* is always larger than or equal to the objective function of the same problem when preemption is allowed, meaning $F^* \geq F_2^*$; consequently, $F^* \geq F_1^*$.

<u>Corollary 2:</u> In problem $1|nr-fpa,bm|\sum_{i=1}^{n} U_i$, it is assumed that σ is the partial sequence of scheduled jobs and σ' is the set of unscheduled jobs. Assume that preemption is allowed: the duration of continuous working time of the machine in each period being T_2 and the length of each unavailability period being W_1. So, if the jobs in set σ' are scheduled using H_1 algorithm, the number of tardy jobs is $\sum_{i\in\sigma'} U_i$. As a result, the value of $\sum_{i\in\sigma} U_i + \sum_{i\in\sigma'} U_i$ is a lower bound for partial sequence σ.

<u>Corollary 3:</u> In problem $1||\sum_{i=1}^{n} U_i$, there is no availability constraint and if any availability constraint is added to the problem, then the completion times will not decrease. As a result, the number of tardy jobs in the optimal solution to problem $1||\sum_{i=1}^{n} U_i$ is always smaller than or equal to its corresponding value in the optimal solution of problem $1|r-fpa|\sum_{i=1}^{n} U_i$.

<u>Theorem 5:</u> In the optimal solution of problem $1|nr\text{-}fpa,bm|\sum_{i=1}^{n} U_i$, the order of jobs in each batch follows the procedure of Moore and Hudgson algorithm.

Proof: Scheduling of jobs in each batch is similar to the solution of problem $1||\sum_{i=1}^{n} U_i$ where there are no availability constraints and the optimal solution can be obtained from Moore and Hudgson algorithm.

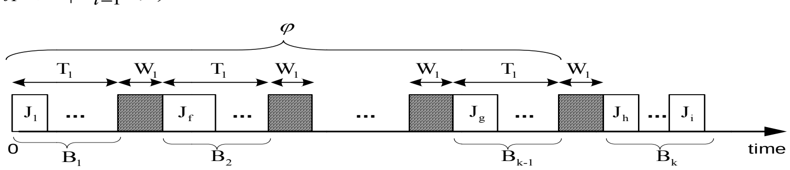

Fig. 2. Sequence S in Theorem 3.

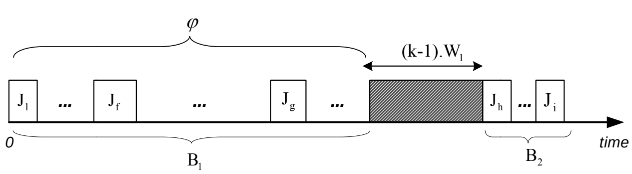

Fig. 3. Sequence S' in Theorem 3.

Using the result of Theorem 5, the following three conclusions may be drawn:

Corollary 4: In problem $1|nr-fpa,bm|\sum_{i=1}^{n} U_i$, the dominant sets consist of those sequences in which the order of jobs in a batch will follow the EDD order if there are no tardy jobs therein.

Corollary 5: In problem $1|nr-fpa,bm|\sum_{i=1}^{n} U_i$, the dominant sets are those sequences in which if a job in one of the batches becomes tardy, then the remaining jobs in that batch will also become tardy.

Corollary 6: In problem $1|nr-fpa,bm|\sum_{i=1}^{n} U_i$, the dominant sets include those sequences in which if job J_i is scheduled in batch B_k and if, further, it has become tardy, then there are no jobs in that batch before J_i whose processing times are larger than that of Ji.

Lemma 1: In problem $1|nr\text{-}fpa,bm|\sum_{i=1}^{n} U_i$, there is one optimal solution in which Eq. (12) holds for job J_i, where job J_i is the first job in batch B_{k+1}.

$$T_p - q_k < p_i \qquad J_i \in B_{k+1}$$
$$k = 1,2,\dots,n-1 \ ; \ p \in \{1,2\} \tag{12}$$

Proof: It is clear that if job J_i can be scheduled in batch B_k, then, by scheduling it in batch B_k, its completion time will definitely be shorter than that in the alternative situation where it is scheduled in batch B_{k+1}. Therefore, neither its tardiness will increase, nor the completion times of other jobs will change.

Lemma 2: In problem $1|nr\text{-}fpa,bm|\sum_{i=1}^{n} U_i$, the dominant sets consist of sequences in which all tardy jobs are located at the end of the sequence.

Proof: Consider sequence S in which job J_i is tardy and there is at least one job without tardiness following it. If job J_i is transferred to the end of sequence S, then sequence S' is obtained. In contrast to S, completion time of the entire jobs except J_i will either remain fixed or decrease in S', while the total number of tardy jobs will not increase.

Corollary 7: If job J_i from set σ' is added to the end of the partial sequence σ and it becomes tardy, then the cardinality of set σ', i.e., $|\sigma'|$, can be calculated and compared to the number of tardy jobs in a feasible solution, i.e., $\sum_{i=1}^{n} U_i$. If Eq. (13) is true, then scheduling the jobs of σ' in the partial sequence σ may be ignored.

$$|\sigma'| \geq \sum_{i=1}^{n} U_i \tag{13}$$

Lemma 3: When job J_i is scheduled in set σ' at the end of partial sequence σ, if J_i becomes tardy and Eq. (14) holds, then by adding set σ' to the end of the partial sequence σ, a complete solution with at most $|\sigma'|$ tardy jobs will be obtained regardless of the job orders and availability length of machines in batches.

$$|\sigma'| < \sum_{i=1}^{n} U_i \tag{14}$$

Proof: According to Lemma 2, if job J_i is tardy, then the whole jobs in set σ' will also become tardy. Therefore, arrangement of these jobs has no impact on improving the objective function. In addition, since their amount of tardiness is of no significance, then they can be scheduled by considering arbitrary maximum availability. In this way, if the number of jobs in set σ' is smaller than $\sum_{i=1}^{n} U_i$, then a complete solution is obtained which has less tardy jobs than the previous solution.

Lemma 4: In partial sequence (σ, W_1, J_i), if job J_i is scheduled in batch B_k and is tardy, then this partial sequence will be dominated by the partial sequence (σ, J_i) where $q_{k-1} + p_i \leq T_2$.

Proof: Consider $C(\sigma)$ to be the completion time for partial sequence σ. Then, tardiness of job J_i in the partial sequence (σ, W_1, J_i) is obtained by Eq. (15).

$$T_i = C(\sigma) + W_1 + p_i - d_i \tag{15}$$

In addition, the tardiness of job J_i in the partial sequence (σ, J_i) can be calculated from Eq. (16).

$$T_i' = \max\{0, C(\sigma) + p_i - d_i\} \tag{16}$$

Comparison of these two equations reveals that it is possible for the tardiness of J_i in (σ, J_i) to become zero and, consequently, the partial sequence (σ, W_1, J_i) to be dominated by partial sequence (σ, J_i).

Lemma 5: Suppose that, in the partial sequence (σ, J_i), job J_i is scheduled in batch B_k and is tardy. If there is a job J_j in set σ' such that it is not tardy in the partial sequence (σ, J_j) and that maximum availability of the machine in B_k does not change, then the partial sequence (σ, J_i) will be dominated by (σ, J_j).

Proof: According to lemma 2, if J_i is tardy, then all the jobs in set σ' will become tardy. Thus, J_i will be definitely tardy in the partial sequence (σ, J_i). Since job J_j in the partial sequence (σ, J_j) is not tardy, then the number of tardy jobs in (σ, J_j) will be less than that in (σ, J_i). Therefore, the partial sequence (σ, J_i) will be dominated by (σ, J_j).

Lemma 6: If in the optimal solution to problem $1||\Sigma_{i=1}^{n} U_i$, the start time of the first tardy job is Q and the maximum earliness of jobs is E_{max}, then there is an optimal solution to problem $1|nr\text{-}fpa,bm|\Sigma_{i=1}^{n} U_i$ in which those jobs that begin after $Q + E_{max}$ will be definitely tardy.

Proof: Assume that the jobs in sequence *S*, as shown in Fig. 4 are scheduled using Moore and Hudgson algorithm without availability constraints. It can be seen that job J_j belongs to the set of non-tardy jobs with an earliness of E_j.

If at least one unavailability period is added to this problem and if job J_j is the first to begin after $Q + E_{max}$, then sequence S' will be obtained as shown in Fig. 5. In sequence S', relation $Q + E_{max} + \varepsilon + p_j > E_j$ is true; thus, job J_j will be definitely tardy. Therefore, in problem $1|nr-fpa,bm|\Sigma_{i=1}^{n} U_i$, those jobs which begin after $Q + E_{max}$ will be tardy.

It should be noted that Chen [23] studied the single machine scheduling problem with fixed periodic availability constraints and with the objective function of the number of tardy jobs as shown by $1|nr-pa|\Sigma_{i=1}^{n} U_i$. To solve this problem, Chen proposed five points which also hold true for problem $1|nr-fpa,bm|\Sigma_{i=1}^{n} U_i$. In this paper, attempt is made to benefit from these points. However, our investigations show that only one of these points, introduced below as Notation 1, can be used in the problem $1|nr-fpa,bm|\Sigma_{i=1}^{n} U_i$ and will be introduced in the following as Notation 1.

In order to introduce Notation 1, it is required to consider sequence $S = (\sigma_1, J_i, \sigma_2, J_j)$ in which σ_1 and σ_2 are partial sequences and $\{J_i, \sigma_2, J_j\}$ belong to B_k. In sequence *S*, as presented in Fig. 6, the completion time of partial sequence σ_1 is $C(\sigma_1)$, and the total processing time of the jobs in σ_2 is considered to be P_{σ_2}. Exchanging jobs J_i and J_j in sequence *S* leads to sequence $S' = (\sigma_1, J_j, \sigma_2, J_i)$ shown in Fig. 7. Consequently, Notation 1 can be concluded for these two sequences.

Notation 1: If Eqs. (17) and (18) are true, then sequence *S* will be dominated by S'.

$$p_j - d_j \le p_i - d_i \tag{17}$$

$$d_j \le d_i \tag{18}$$

5. The Heuristic Algorithm H_2

In this section, a heuristic algorithm called H_2 is proposed to obtain a near-optimal solution to the problem $1|nr-fpa,bm|\Sigma_{i=1}^{n} U_i$.

In algorithm H_2, jobs are initially arranged by the EDD rule. Then, an initial order of jobs will be obtained by algorithm H_1. In the following, it is assumed that just one unavailability period exists for scheduling the remaining jobs; therefore, two batches will exist for scheduling the jobs. One of these batches is located before and the other is located after the unavailability period. In order to schedule the jobs in the first batch, the best batch length and, thereby, the best length of unavailability period should be chosen such that the number of tardy jobs becomes minimized. Then, this batch and its unavailability period are considered fixed and the jobs within it will be kept fixed as well.

In the second step, the same procedure is adopted for the remaining jobs and the process continues until the entire jobs are scheduled. Based on what went above in each batch, the impacts of selecting the maximum machine availability of length T_1 or T_2 is calculated, and this will be used to choose the duration of the unavailability period.

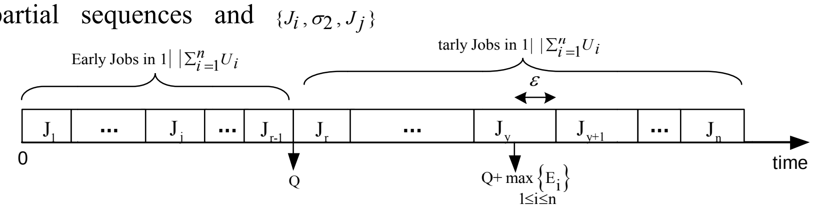

Fig. 4. Sequence S in Lemma 6

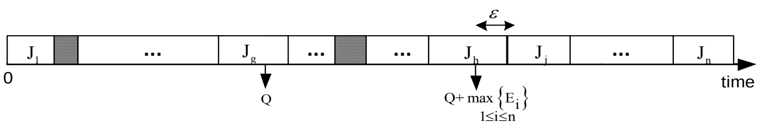

Fig. 5. Sequence S' in Lemma 6

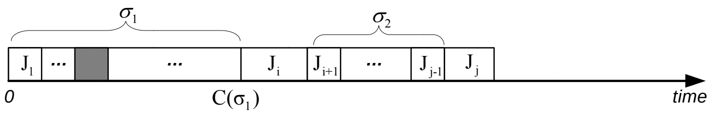

Fig. 6. Sequence S in Notation 1

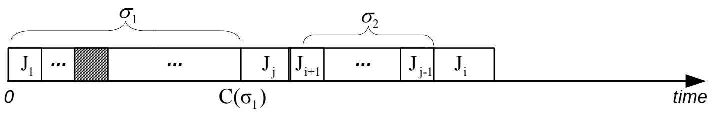

Fig. 7. Sequence S' in Notation 1

Suppose that some jobs in partial sequence σ are scheduled in batches 1 to B_{k-1}. Two scenarios can be imagined for the maximum machine availability in batch B_k. In the first sequence, represented by sequence S in Fig. 8, this maximum value is T_1 and r jobs are scheduled in this batch. Following this batch, an unavailability period of the length W_1 is set. Then, the remaining jobs are scheduled using the order obtained from algorithm H_1 and without availability constraint.

In the second scenario, represented by S' in Fig. 9, it is assumed that the maximum availability of the machine is T_2 and y jobs ($y>r$) can be processed in batch B_k. After this batch, one unavailability period with length W_2 is scheduled and the remaining jobs are scheduled using the order obtained from algorithm H_1.

If the number of tardy jobs in sequence S is less than that in sequence S' , then sequence S is better and the maximum availability of the machine in the current batch should be considered as T_1; otherwise, T_2 should be adopted. In addition to the symbols introduced earlier, the following symbols are also used in algorithm H_2:

i: Job number.

j: Location in the schedule.

$N(T_1)$: Number of tardy jobs resulting from scheduling jobs with maximum availability of T_1.

$N(T_2)$: Number of tardy jobs resulting from scheduling jobs with maximum availability of T_2.

Algorithm H_2 consists of the following steps:

Step 0: Set $\sigma' = \{J_1, J_2, ..., J_n\}$ and parameters i, j, δ, q_k and y equal to zero and k=1.

Step 1: By uing algorithm H_1 within the maximum continuous working time of machine T_2 and unavailability duration W_1, obtain an initial arrangement of jobs and name them based on this order.

Step 2: Select the jobs based on the order assigned in Step 1. Set $i=i+1$ and $j=j+1$. If $i=n+1$, then go to Step 8; otherwise, calculate $q_k = q_k + p_i$. If $q_k > T_1$, then set $y=i$ and go to Step 3; otherwise, schedule job J_i in the location of j. Update the jobs in σ' as $\sigma' = \sigma' - \{J_i\}$ and go to Step 2.

Step 3: Suppose that the maximum availability of the machine for the kth batch is T_1. Schedule job J_i in batch B_{k+1} and use algorithm H_1 within the maximum continuous working time of machine T_2 and the unavailability duration W_1 to schedule the remaining jobs. Put the number of tardy jobs resulting from this sequence into $N(T_1)$. Go to Step 4.

Step 4: Set $\delta = q_k - p_i$. Put the set of jobs σ' into a set φ and create a list M.

Step 4-1: Select the first job from the set of jobs φ and name it J_g.

Step 4-2: Set $\delta = \delta + p_i$.

Step 4-3: If $\delta \le T_2$, then put job J_g in list M and update set φ as $\varphi = \varphi - \{J_g\}$ and set $y = y + 1$. If $\varphi = \varnothing$, then go to Step 4-5; otherwise, go to Step 4-1.

Step 4-4: If $\delta > T_2$, then go to Step 4-5.

Step 4-5: Suppose that the maximum availability of the machine in the kth batch is T_2 and schedule job J_i accompanied with the jobs in list M in this set. Schedule the jobs in $\sigma' = \sigma' - M$ using algorithm H_1 within the maximum continuous working time of the machine T_2 and the unavailability duration W_1 and put the resulting number of tardy jobs into $N(T_2)$. Go to Step 5.

Step 5: If $N(T_2) > N(T_1)$, then go to Step 6. Otherwise, go to Step 7.

Step 6: Schedule one unavailability period with the length of W_1 in location j and schedule job J_i in location $j+1$. Set $k=k+1$, $j=j+2$, and $q_k = p_i$, and then go to Step 2.

Step 7: In locations j to $j+y$, schedule the jobs in list M. Schedule one unavailability period with the length of W_2 in location $j+y+1$. Set $i=i-1+y$, $k=k+1$, $j=j+y+2$, and $q_k=0$. Update set σ' as $\sigma' = \sigma' - M$ and go to Step 2.

Step 8: Calculate the objective function.

Step 1 of algorithm H_2 has a time complexity of $O(n \log n)$. Steps 3 and 4 are of time complexity $O(n)$. Therefore, algorithm H_2 has a time complexity of $O(n \log n)$.

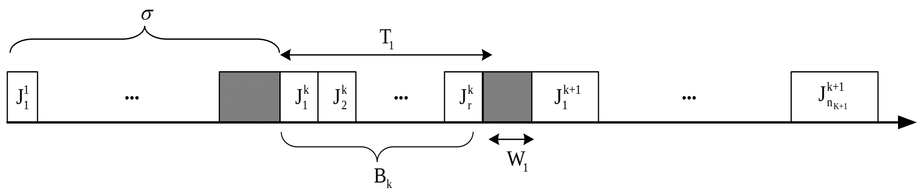

Fig. 8. Sequence S in algorithm H_2

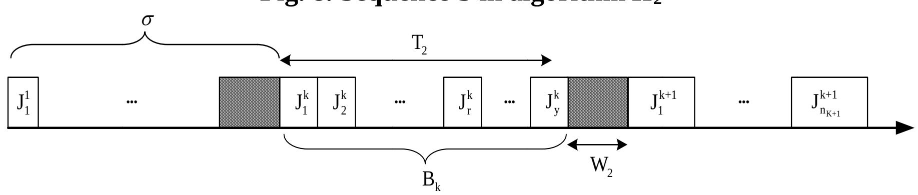

Fig. 9. Sequence S' in the algorithm H_2

6. The Branch-and-Bound Algorithm

In order to solve problem $1|nr\text{-}fpa,bm|\sum_{i=1}^{n} U_i$, a depth first search branch-and-bound algorithm has been used. In this algorithm, jobs are initially arranged based on the EDD order and renumbered in that same order. This arrangement is taken as the input to the branch-and-bound algorithm. In this algorithm, the solution of algorithm H_2 is used as the upper bound, Theorem 4 as the lower bound, and Theorem 5, Lemma 2, Lemma 3, and Notation 1 as dominance rules.

After scheduling each job J_i by the branch-and-bound algorithm, there will be two possible branches in the tree, indicating that there are two choices to schedule after scheduling job J_i: job J_{i+1} or an unavailability period. Consequently, the number of branches will be at most $2^n n!$.

As shown in Fig. 10, when scheduling job J_{i+1} in B_k, after job J_i, there will be two branches. In first branch, one availability constraint with value W_1 (or W_2) can be scheduled after J_i. In the second one, J_{i+1} to J_{n_k} will be scheduled after J_i till $q_{k+1} \geq T_1$ *or* T_2. Then, an availability constraint with value W_1 (or W_2) can be considered after J_{n_k}. In each branch, if any of lemmas or theorems is violated, the branch will be fathomed. In addition, if both of branches cannot be fathomed, then the branch with smaller number of tardy jobs should be continued and other one fathomed.

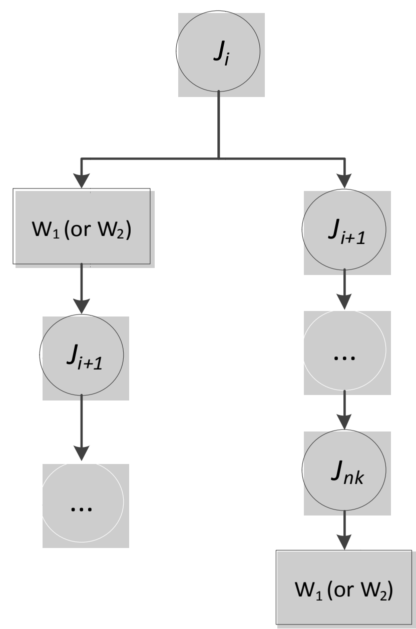

Fig. 10. The branch-and-bound tree

7. Generalization of $1|nr\text{-}fpa,bm|\sum_{i=1}^{n} U_i$

Some of the lemmas and theorems proposed for the problem $1|nr\text{-}fpa,bm|\sum_{i=1}^{n} U_i$ can be generalized to a more general form represented by $1|nr\text{-}fpa,mm|\sum_{i=1}^{n} U_i$ with multiple maximum continuous working time of the machine. In the multi-modal problem, the maximum continuous working time of the machine in each period can adopt P different values consisting of $T_1, T_2, ..., T_P$ and the length of each unavailability period can have P different values consisting of $W_1, W_2, ..., W_P$ depending on the maximum continuous working time of the machine. The mathematical model proposed in Section 2 can be easily generalized to problem $1|nr\text{-}fpa,mm|\Sigma_{i=1}^{n} U_i$. To do so, it will suffice to change the boundaries of p in Eqs. (4), (6), and (10) from 2 to P. Moreover, Theorems 2, 4, and 6, Lemmas 1 to 6, and Notation 1 will not depend on the length of unavailability period or on the length of batches; thus, they are entirely true for problem $1|nr\text{-}fpa,mm|\Sigma_{i=1}^{n} U_i$.

Note that a lower bound for problem $1|nr\text{-}fpa,mm|\Sigma_{i=1}^{n} U_i$ can be obtained using Theorem 4. To do so, it will suffice to calculate the optimal solution to problem $1|r-fpa|\Sigma_{i=1}^{n} U_i$ using H_1 algorithm with two conditions: maximum duration of the continuous working time of the machine is $max_{1\le p\le P}\, T_p$ and the length of the unavailability period is $min_{1\le p\le P}\, W_p$. On the other hand, Lemma 4 is generalized to Lemma 7 for problem $1|nr\text{-}fpa,mm|\Sigma_{i=1}^{n} U_i$.

<u>Lemma 7:</u> Suppose that job J_i in the partial sequence (σ, W_p, J_i) is scheduled in batch B_k and is tardy. Then, this partial sequence will be dominated by the partial sequence (σ, J_i) where $q_{k-1} + p_i \le T_{p'}$ and $p' > p$.

8. Computational Results

To evaluate the performances of heuristic algorithm H_2 and the proposed branch-and-bound algorithm, they were coded in the C# programming language and executed on an Intel (R) Core 2Due 3.16 GHz with 2 GB RAM in WINDOWS 7 environment. If execution time of the problem exceeded 3600 seconds, the branch-and-bound algorithm would be automatically terminated.

In order to generate sample problems, the approach proposed in Chen [23] was used. According to this approach, processing times are randomly generated from a uniform discrete distribution over the interval [1, 10]. Due dates are also randomly generated using the following uniform discrete distribution.

$$d_i = \left[\left(1 - C - \frac{Q}{2}\right).\Sigma_{i=1}^{n} p_i ,\left(1 - C + \frac{Q}{2}\right).\Sigma_{i=1}^{n} p_i\right] \qquad (19)$$

In Eq. (19), C is the tardiness parameter chosen from the set {0.2, 0.6}; Q is the parameter of due dates chosen from the set {0.2, 0.6}; T_1 has the values {10, 15, 20}, and W_1=6.

The number of jobs, i.e., n, is assumed to be any number in the set {8, 10, 12, 14, 16, 18, 20}. Also, T_2 has been selected from {1.4 T_1, 1.8 T_1 } and $W_2 = 1.6\ W_1$. Any possible permutation of C, Q, T_1, T_2, W_1 and W_2 is called a series, yielding 24 (i.e., 2×2×3×2×1×1) series. In each series, for each n, 10 problems are generated (np=10) to create a total number of 1680 (24×7×10) problems.

Computational results of solving the sample problems are presented in Table 1, where specifications of the series, number of problems solved by the proposed branch-and-bound method, and the mean average error of the heuristic algorithm are reported. Since the number of tardy jobs in a sample problem may become zero, an appropriate measure has been utilized to calculate the error of algorithm H_2 in order to evaluate its performance. Moslehi and Jafari [24] used $\frac{Z^{opt}}{Z^{H_2}}$ as a measure for comparing the results obtained from both algorithm H_2 and the branch-and-bound algorithm. In this measure, Z^{opt} is the optimal value and Z^{H_2} is the objective function of algorithm H_2 both of which are calculated from $\Sigma_{i=1}^{n}(1-U_i)$. Therefore, $\frac{Z^{opt}}{Z^{H_2}}$ is always larger than or equal to unity, and the denominator of the measure will never be zero. The closer $\frac{Z^{opt}}{Z^{H_2}}$ gets to 1, the higher the efficiency of the algorithm H_2 will be. The algorithm obtains the optimal value when this measure becomes equal to 1. Under the column $\frac{Z^{opt}}{Z^{H_2}}$, the mean percentage errors of algorithm H_2 are reported, and this value is calculated from optimal solutions which are obtained within the 3600 second boundary. In addition, the average percentages of the nodes fathomed to the total number of crossed nodes as well as the average percentages of nodes fathomed based on the fathoming reason are reported. It is worth mentioning that some of the lemmas and theorems have not been used in the branch-and-bound algorithm due to their computational inefficiency. The last column in the Table presents the average execution times in seconds for problems that were optimally solved using algorithms H_2 and the branch-and-bound method. It is also clear from Table 1 that since the total number of states in problem $1|nr\text{-}fpa,bm|\Sigma_{i=1}^{n} U_i$ is at most $2^n n!$, any increase in problem dimensions in each series results in a corresponding increase in the solution time of the branch-and-bound algorithm.

Fig. 11 presents the number of problems whose optimal solutions were obtained using either algorithms H_2 or the branch-and-bound in less than 3600 seconds. This includes all of the 24 series based on changes in parameters C and Q. Clearly, it can be concluded from the comparison of the number of problems solved in each series that by decreasing C and Q values, in more problems, the optimal solutions were obtained in less than 3600.

Fig. 12 shows that when Q decreases, the number of problems whose optimal solutions are obtained from algorithm H_2 will decrease. Generally, lowering the ranges of due dates leads to reduced optimal solutions obtained in less than 3600 seconds. On the other hand, the solution time for series 1 to 24 reduces when T_1 increases. This is because the number of batches needed to schedule the jobs decreases and, consequently, more optimal solutions are obtained in less than 3600 seconds.

Fig. 13 presents the trend in the changes in the number of problems solved by H_2 and branch-and-bound algorithms with respect to T_1. As observed in Fig. 13, the number of times that H_2 algorithm reaches optimal solutions increases when T_1 increases. The reason may lie in the reduced number of batches needed for scheduling the jobs, which thereby reduces the errors resulting from inappropriate selection of unavailability durations. Based on our computational results, the mean percentage error for algorithm H_2 is 2 percent, which confirms the high capability of the algorithm.

Fig. 14 displays the mean solution times obtained from the branch-and-bound algorithm versus T_1. It can be observed that this parameter decreases by increasing T_1.

It can be seen in Fig. 15 that in series with identical values of T_1, solution time decreases when T_2 - T_1 is larger than T_1. The reason is that the number of batches needed for scheduling jobs reduces in the cases where there is a batch with a maximum availability of T_2 in the optimal solution.

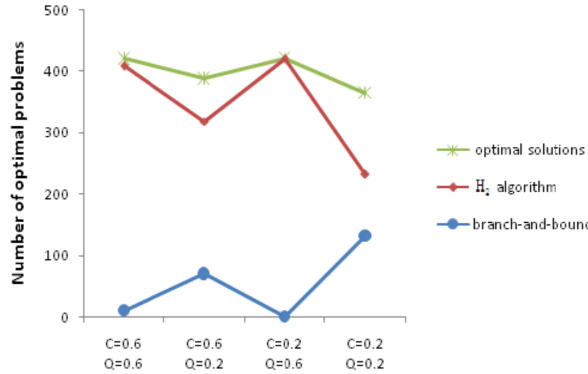

Fig. 11. Performances of H_2 and branch-and-bound algorithms versus changes in C and Q

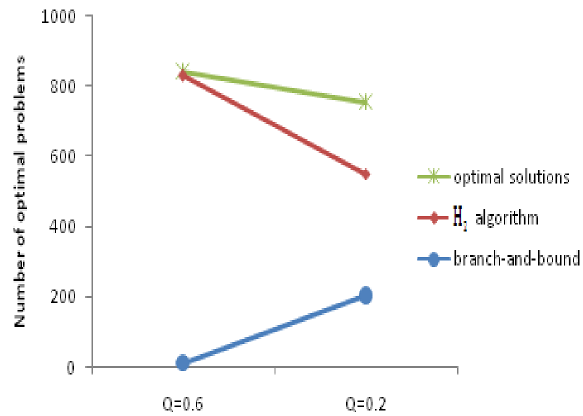

Fig. 12. Performances of H2 and branch-and-bound algorithms versus changes in Q

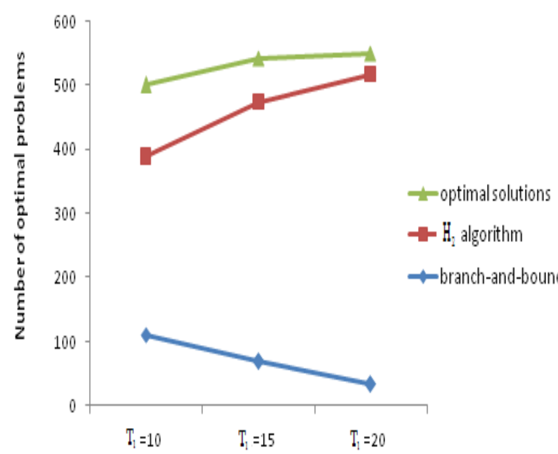

Fig. 13. Performances of the H_2 and branch-and-bound algorithms versus changes in T_1

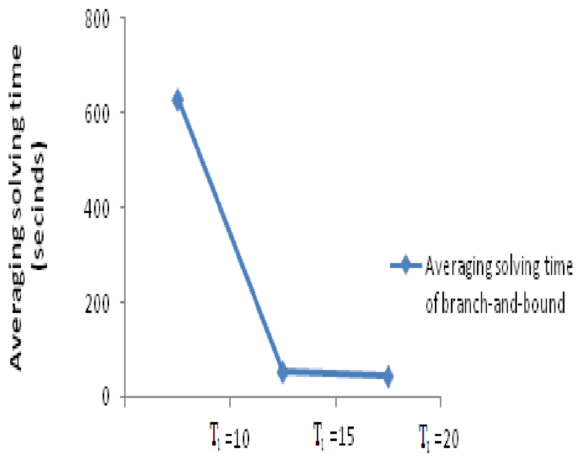

Fig. 14. Mean solution time recorded for the branch-and-bound algorithm versus T_1

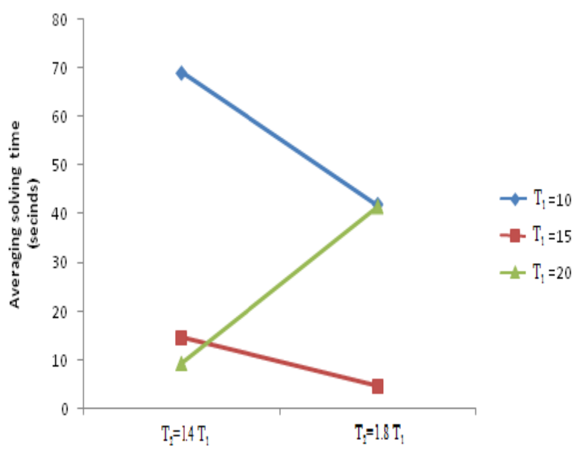

Fig. 15. Mean solution time obtained by branch-and-bound algorithm in series with identical values of T_2

9. Conclusions and Suggestions for Future Studies

In this paper, the single machine scheduling problem with bimodal flexible and periodic availability constraints was investigated. According to this definition, it is assumed that, in each period, the maximum continuous working time of the machine may adopt either of two different values, and the duration of each unavailability period depends on the maximum availability of the machine in that period which can also adopt either of two different values. Our objective was to minimize the number of tardy jobs in a problem denoted by $1|nr\text{-}fpa,bm|\sum_{i=1}^{n} U_i$.
In order to obtain the optimal solution, a binary integer programming model was initially proposed. It was shown that optimal solutions to problems of larger scales are not possible due to the complexity of the model. Then, several lemmas and theorems were introduced and proved. Finally, a heuristic algorithm and an efficient branch-and-bound algorithm were proposed to solve the problem. Dominance rules, lower bounds, and upper bounds were implemented in both algorithms. The proposed branch-and-bound algorithm was found to be capable of solving problems up to 20 jobs. In addition, in section 7, the problem was scaled up to investigate situations where there are multiple-modal availability constraints. It was demonstrated that most of lemmas and theorems, which are proposed for bimodal problem, also hold for the multiple-modal problem.
Further studies were suggested to enhance the efficiency of the branch-and-bound algorithm through improving heuristic algorithm H_2 as well as the lower bound of the problem. In addition, it will be desirable to investigate the single machine scheduling problem with bimodal flexible and periodic availability constraints in which two predetermined intervals are considered in each period for the occurrence of unavailability and in which the length of each unavailability period depends on the interval changes and can be selected in a stepwise manner.

Tab. 1. Computational results for sample problems.

Series	n	Number of Optimal instances	$\frac{Z^{opt}}{Z^{H_2}}$	Average percentage of entire fathomed nodes	Average percentage of fathomed nodes by					Avg. solving time	
					Lower bound	Theorem 5	Lemma 2	Notation 1	Lemma 3	H_2	*B&B*
Series 1 C = 0.2 Q = 0.2	8	10	1.02	7.29	76.74	2.33	18.60	0.00	2.33	0.06	0.11
	10	10	1.02	13.81	70.81	23.89	4.43	0.80	0.07	0.05	1.31
	12	10	1.09	53.31	77.49	17.02	2.76	2.65	0.08	0.03	3.99

$T_1 = 10$	14	10	1.16	59.70	85.90	8.73	4.77	0.59	0.00	0.07	325.9
$W_1 = 6$	16	10	1.09	52.59	76.29	22.87	0.30	0.53	0.00	0.06	124.4
$T_2 = 1.4\,T_1$	18	10	1.13	61.01	81.91	14.19	3.64	0.26	0.00	0.14	1226.
$W_2 = 1.6\,W_1$	20	10	1.36	87.50	79.98	18.54	0.19	1.29	0.01	0.16	223.8
Series 2	8	10	1.00	17.09	74.49	20.56	1.09	3.87	0.00	0.02	0.50
$C = 0.2$	10	10	1.10	50.11	70.22	20.42	4.71	4.53	0.12	0.05	4.39
$Q = 0.2$	12	10	1.01	17.28	54.60	26.58	16.65	2.17	0.00	0.02	38.42
$T_1 = 10$	14	10	1.12	79.47	69.44	19.07	10.44	1.05	0.00	0.05	253.4
$W_1 = 6$	16	10	1.10	56.05	64.24	23.53	11.14	1.09	0.00	0.04	545.0
$T_2 = 1.8\,T_1$	18	10	1.08	44.87	74.97	20.99	3.71	0.33	0.00	0.10	756.5
$W_2 = 1.6\,W_1$	20	10	--	--	--	--	--	--	--	--	--
Series 3	8	10	1.00	0.00	++	++	++	++	++	0.01	0.02
$C = 0.2$	10	10	1.11	49.43	76.34	1.73	19.52	1.21	1.21	0.02	0.56
$Q = 0.2$	12	10	1.00	34.25	63.13	34.44	0.00	2.43	0.00	0.01	5.61
$T_1 = 15$	14	10	1.07	41.45	72.46	20.51	6.03	1.00	0.01	0.03	95.04
$W_1 = 6$	16	10	1.09	43.01	70.39	20.56	8.15	0.84	0.06	0.03	8.55
$T_2 = 1.4\,T_1$	18	10	1.09	31.79	68.21	25.51	5.27	1.00	0.00	0.07	249.9
$W_2 = 1.6\,W_1$	20	10	1.10	61.29	71.36	26.82	0.56	1.26	0.00	0.06	255.2

--: No optimal solution achieved by branch-and-bound and H_2 algorithms.

++: Optimal solution achieved by H_2 algorithm and branch-and-bound didn't use.

Tab. 1. continued.

Series	n	Number of Optimal instances	$\frac{Z^{opt}}{Z^{H_2}}$	Average percentage of entire fathomed nodes	Average percentage of fathomed nodes by					Avg. solving time	
					Lower bound	Theorem 5	Lemma 2	Notation 1	Lemma 3	H_2	*B&B*
Series 4	8	10	1.00	0.00	++	++	++	++	++	0.02	0.02
$C = 0.2$	10	10	1.01	17.44	69.89	28.40	0.41	1.23	0.06	0.02	0.73
$Q = 0.2$	12	10	1.02	34.72	54.65	37.87	0.17	7.30	0.01	0.01	4.67
$T_1 = 15$	14	10	1.06	26.96	58.16	40.98	0.67	0.17	0.02	0.02	6.00
$W_1 = 6$	16	10	1.04	20.37	56.31	42.74	0.19	0.74	0.01	0.02	6.31
$T_2 = 1.8\,T_1$	18	10	1.11	67.69	58.21	29.15	12.22	0.36	0.07	0.05	23.84
$W_2 = 1.6\,W_1$	20	10	1.05	30.92	87.99	0.56	10.89	0.00	0.56	0.05	1.06
Series 5	8	10	1.00	0.00	++	++	++	++	++	0.01	0.01
$C = 0.2$	10	10	1.01	16.79	71.35	27.01	0.38	1.22	0.05	0.02	0.82
$Q = 0.2$	12	10	1.01	8.53	87.57	7.57	3.24	1.08	0.54	0.01	0.12
$T_1 = 20$	14	10	1.01	26.80	58.84	38.84	0.02	2.30	0.00	0.02	13.97
$W_1 = 6$	16	10	1.03	27.24	58.45	39.71	0.02	1.81	0.00	0.02	24.28
$T_2 = 1.4\,T_1$	18	10	1.01	18.33	73.79	26.16	0.03	0.02	0.00	0.05	19.35
$W_2 = 1.6\,W_1$	20	10	1.08	38.77	62.66	33.63	2.93	0.77	0.01	0.04	92.00
Series 6	8	10	1.00	0.00	++	++	++	++	++	0.01	0.01
$C = 0.2$	10	10	1.00	0.00	++	++	++	++	++	0.02	0.02
$Q = 0.2$	12	10	1.00	8.86	49.20	50.22	0.33	0.25	0.00	0.01	3.06

$T_1 = 20$	14	10	1.10	53.74	53.86	44.42	0.03	1.68	0.01	0.02	37.44
$W_1 = 6$	16	10	1.03	54.16	47.03	51.85	0.01	1.11	0.00	0.02	249.45
$T_2 = 1.8\,T_1$	18	10	1.03	22.99	48.17	50.09	0.00	1.74	0.00	0.03	414.67
$W_2 = 1.6\,W_1$	20	10	1.03	18.30	85.57	1.72	12.03	0.00	0.69	0.04	0.48

Tab. 1. continued.

Series	n	Number of Optimal instances	$\frac{Z^{opt}}{Z^{H_2}}$	Average percentage of entire fathomed nodes	Average percentage of fathomed nodes by					Avg. solving time	
					Lower bound	Theorem 5	Lemma 2	Notation 1	Lemma 3	H_2	*B&B*
Series 7	8	10	1.00	0.00	++	++	++	++	++	0.02	0.02
$C = 0.2$	10	10	1.00	0.00	++	++	++	++	++	0.03	0.03
$Q = 0.6$	12	10	1.00	0.00	++	++	++	++	++	0.02	0.02
$T_1 = 10$	14	10	1.00	0.00	++	++	++	++	++	0.02	0.02
$W_1 = 6$	16	10	1.00	0.00	++	++	++	++	++	0.03	0.03
$T_2 = 1.4\,T_1$	18	10	1.00	0.00	++	++	++	++	++	0.05	0.05
$W_2 = 1.6\,W_1$	20	10	1.00	0.00	++	++	++	++	++	0.05	0.05
Series 8	8	10	1.00	0.00	++	++	++	++	++	0.02	0.02
$C = 0.2$	10	10	1.00	0.00	++	++	++	++	++	0.02	0.02
$Q = 0.6$	12	10	1.00	0.00	++	++	++	++	++	0.01	0.01
$T_1 = 10$	14	10	1.00	0.00	++	++	++	++	++	0.01	0.02
$W_1 = 6$	16	10	1.00	0.00	++	++	++	++	++	0.02	0.03
$T_2 = 1.8\,T_1$	18	10	1.00	0.00	++	++	++	++	++	0.04	0.04
$W_2 = 1.6\,W_1$	20	10	1.00	0.00	++	++	++	++	++	0.04	0.04
Series 9	8	10	1.00	0.00	++	++	++	++	++	0.01	0.02
$C = 0.2$	10	10	1.00	0.00	++	++	++	++	++	0.02	0.02
$Q = 0.6$	12	10	1.00	0.00	++	++	++	++	++	0.01	0.02
$T_1 = 15$	14	10	1.00	0.00	++	++	++	++	++	0.01	0.01
$W_1 = 6$	16	10	1.00	0.00	++	++	++	++	++	0.02	0.02
$T_2 = 1.4\,T_1$	18	10	1.00	0.00	++	++	++	++	++	0.03	0.03
$W_2 = 1.6\,W_1$	20	10	1.00	0.00	++	++	++	++	++	0.03	0.03

Tab. 1. continued.

Series	n	Number of Optimal instances	$\frac{Z^{opt}}{Z^{H_2}}$	Average percentage of entire fathomed nodes	Average percentage of fathomed nodes by					Avg. solving time	
					Lower bound	Theorem 5	Lemma 2	Notation 1	Lemma 3	H_2	*B&B*
Series 10	8	10	1.00	0.00	++	++	++	++	++	0.01	0.02
$C = 0.2$	10	10	1.00	0.00	++	++	++	++	++	0.02	0.02
$Q = 0.6$	12	10	1.00	0.00	++	++	++	++	++	0.01	0.01
$T_1 = 15$	14	10	1.00	0.00	++	++	++	++	++	0.01	0.01
$W_1 = 6$	16	10	1.00	0.00	++	++	++	++	++	0.02	0.02
$T_2 = 1.8\,T_1$	18	10	1.00	0.00	++	++	++	++	++	0.02	0.02

$W_2 = 1.6\, W_1$	20	10	1.00	0.00	++	++	++	++	++	0.03	0.03
Series 11	8	10	1.00	0.00	++	++	++	++	++	0.01	0.01
$C = 0.2$	10	10	1.00	0.00	++	++	++	++	++	0.02	0.02
$Q = 0.6$	12	10	1.00	0.00	++	++	++	++	++	0.01	0.01
$T_1 = 20$	14	10	1.00	0.00	++	++	++	++	++	0.01	0.01
$W_1 = 6$	16	10	1.00	0.00	++	++	++	++	++	0.01	0.02
$T_2 = 1.4\, T_1$	18	10	1.00	0.00	++	++	++	++	++	0.02	0.03
$W_2 = 1.6\, W_1$	20	10	1.00	0.00	++	++	++	++	++	0.02	0.02
Series 12	8	10	1.00	0.00	++	++	++	++	++	0.01	0.01
$C = 0.2$	10	10	1.00	0.00	++	++	++	++	++	0.01	0.01
$Q = 0.6$	12	10	1.00	0.00	++	++	++	++	++	0.01	0.01
$T_1 = 20$	14	10	1.00	0.00	++	++	++	++	++	0.01	0.01
$W_1 = 6$	16	10	1.00	0.00	++	++	++	++	++	0.01	0.01
$T_2 = 1.8\, T_1$	18	10	1.00	0.00	++	++	++	++	++	0.02	0.02
$W_2 = 1.6\, W_1$	20	10	1.00	0.00	++	++	++	++	++	0.02	0.02

Tab. 1. continued.

Series	n	Number of Optimal instances	$\frac{Z^{opt}}{Z^{H_2}}$	Average percentage of entire fathomed nodes	Average percentage of fathomed nodes by					Avg. solving time	
					Lower bound	Theorem 5	Lemma 2	Notation 1	Lemma 3	H_2	*B&B*
Series 13	8	10	1.03	34.11	77.55	16.78	3.40	2.27	0.00	0.02	2.85
$C = 0.6$	10	10	1.06	15.95	85.85	13.87	0.16	0.03	0.08	0.03	2.07
$Q = 0.2$	12	10	1.08	36.51	81.20	15.76	2.41	0.63	0.00	0.02	283.3
$T_1 = 10$	14	10	1.10	37.69	76.10	19.11	3.68	1.10	0.00	0.10	324.5
$W_1 = 6$	16	10	1.04	30.89	71.54	22.03	4.58	1.84	0.00	0.11	73.80
$T_2 = 1.4\, T_1$	18	10	1.08	75.92	81.82	14.36	2.79	1.03	0.00	0.07	663.5
$W_2 = 1.6\, W_1$	20	10	1.15	71.82	76.40	21.21	1.21	1.18	0.00	0.05	886.7
Series 14	8	10	1.01	32.00	75.01	20.29	3.28	1.42	0.00	0.02	3.73
$C = 0.6$	10	10	1.07	34.43	78.97	19.27	1.52	0.22	0.01	0.02	9.07
$Q = 0.2$	12	10	1.03	42.69	73.26	24.83	1.84	0.07	0.00	0.01	184.5
$T_1 = 10$	14	10	1.05	45.21	75.33	22.40	1.06	1.21	0.00	0.07	69.01
$W_1 = 6$	16	10	1.04	10.23	65.96	17.60	14.35	2.04	0.05	0.04	6.54
$T_2 = 1.8\, T_1$	18	10	1.01	20.05	67.77	28.97	2.85	0.42	0.00	0.05	262.7
$W_2 = 1.6\, W_1$	20	10	1.00	0.00	++	++	++	++	++	0.05	0.05
Series 15	8	10	1.01	15.78	74.22	21.49	3.37	0.92	0.00	0.01	4.89
$C = 0.6$	10	10	1.03	24.74	77.80	21.85	0.35	0.00	0.00	0.02	0.68
$Q = 0.2$	12	10	1.04	31.34	88.09	9.62	1.98	0.31	0.00	0.01	0.27
$T_1 = 15$	14	10	1.04	25.68	63.59	27.23	6.25	2.89	0.04	0.05	2.10
$W_1 = 6$	16	10	1.00	0.00	++	++	++	++	++	0.03	0.04
$T_2 = 1.4\, T_1$	18	10	1.03	27.73	66.98	29.20	2.45	1.36	0.00	0.04	58.12
$W_2 = 1.6\, W_1$	20	10	1.02	20.52	59.62	39.41	0.83	0.14	0.00	0.03	5.12

Tab. 1. continued.

Series	n	Number of Optimal instances	$\frac{Z^{opt}}{Z^{H_2}}$	Average percentage of entire fathomed nodes	Average percentage of fathomed nodes by					Avg. solving time	
					Lower bound	Theorem 5	Lemma 2	Notation 1	Lemma 3	H_2	*B&B*
Series 16	8	10	1.03	7.73	94.12	0.00	5.88	0.00	0.00	0.02	0.05
$C = 0.6$	10	10	1.00	8.47	58.26	40.28	1.46	0.00	0.00	0.02	8.28
$Q = 0.2$	12	10	1.00	8.58	58.98	39.30	1.07	0.65	0.00	0.01	35.67
$T_1 = 15$	14	10	1.00	0.00	++	++	++	++	++	0.05	0.05
$W_1 = 6$	16	10	1.00	0.00	++	++	++	++	++	0.03	0.03
$T_2 = 1.8\,T_1$	18	10	1.00	0.00	++	++	++	++	++	0.02	0.02
$W_2 = 1.6\,W_1$	20	10	1.01	8.51	63.78	32.69	1.39	2.14	0.00	0.03	14.19
Series 17	8	10	1.01	7.75	93.55	0.00	6.45	0.00	0.00	0.01	0.03
$C = 0.6$	10	10	1.00	0.00	++	++	++	++	++	0.01	0.01
$Q = 0.2$	12	10	1.00	0.00	++	++	++	++	++	0.01	0.01
$T_1 = 20$	14	10	1.00	0.00	++	++	++	++	++	0.03	0.04
$W_1 = 6$	16	10	1.00	0.00	++	++	++	++	++	0.03	0.03
$T_2 = 1.4\,T_1$	18	10	1.00	0.00	++	++	++	++	++	0.03	0.03
$W_2 = 1.6\,W_1$	20	10	1.01	9.19	100.00	0.00	0.00	0.00	0.00	0.02	0.16
Series 18	8	10	1.00	8.36	58.21	40.85	0.00	0.93	0.00	0.02	0.28
$C = 0.6$	10	10	1.00	0.00	++	++	++	++	++	0.01	0.01
$Q = 0.2$	12	10	1.00	0.00	++	++	++	++	++	0.01	0.01
$T_1 = 20$	14	10	1.00	0.00	++	++	++	++	++	0.03	0.03
$W_1 = 6$	16	10	1.00	0.00	++	++	++	++	++	0.02	0.02
$T_2 = 1.8\,T_1$	18	10	1.00	0.00	++	++	++	++	++	0.02	0.02
$W_2 = 1.6\,W_1$	20	10	1.00	0.00	++	++	++	++	++	0.03	0.03

Tab. 1. continued.

Series	n	Number of Optimal instances	$\frac{Z^{opt}}{Z^{H_2}}$	Average percentage of entire fathomed nodes	Average percentage of fathomed nodes by					Avg. solving time	
					Lower bound	Theorem 5	Lemma 2	Notation 1	Lemma 3	H_2	*B&B*
Series 19	8	10	1.02	7.40	85.06	11.04	3.90	0.00	0.00	0.02	0.14
$C = 0.6$	10	10	1.03	15.77	74.71	15.12	9.52	0.52	0.13	0.02	0.76
$Q = 0.6$	12	10	1.02	7.67	79.12	15.38	5.27	0.00	0.22	0.02	0.20
$T_1 = 10$	14	10	1.00	0.00	++	++	++	++	++	0.02	0.02
$W_1 = 6$	16	10	1.04	8.16	88.68	9.93	1.33	0.00	0.07	0.02	2.36
$T_2 = 1.4\,T_1$	18	10	1.00	0.00	++	++	++	++	++	0.06	0.06
$W_2 = 1.6\,W_1$	20	10	1.00	0.00	++	++	++	++	++	0.05	0.06
Series 20	8	10	1.00	8.45	77.27	20.95	1.78	0.00	0.00	0.02	0.38
$C = 0.6$	10	10	1.04	17.05	72.72	20.95	5.65	0.61	0.07	0.02	1.08
$Q = 0.6$	12	10	1.01	8.98	81.08	17.33	1.59	0.00	0.00	0.02	2.53
$T_1 = 10$	14	10	1.00	0.00	++	++	++	++	++	0.02	0.02

$W_1 = 6$	16	10	1.00	0.00	++	++	++	++	++	0.02	0.02
$T_2 = 1.8\,T_1$	18	10	1.00	0.00	++	++	++	++	++	0.04	0.05
$W_2 = 1.6\,W_1$	20	10	1.00	0.00	++	++	++	++	++	0.04	0.05
Series 21	8	10	1.00	0.00	++	++	++	++	++	0.01	0.02
$C = 0.6$	10	10	1.01	8.11	96.67	0.00	3.33	0.00	0.00	0.02	0.05
$Q = 0.6$	12	10	1.01	8.33	97.78	0.00	2.22	0.00	0.00	0.01	0.06
$T_1 = 15$	14	10	1.00	0.00	++	++	++	++	++	0.02	0.02
$W_1 = 6$	16	10	1.00	0.00	++	++	++	++	++	0.02	0.02
$T_2 = 1.4\,T_1$	18	10	1.00	0.00	++	++	++	++	++	0.03	0.03
$W_2 = 1.6\,W_1$	20	10	1.00	0.00	++	++	++	++	++	0.04	0.04

Tab. 1. continued.

Series	n	Number of Optimal instances	$\frac{Z^{opt}}{Z^{H_2}}$	Average percentage of entire fathomed nodes	Average percentage of fathomed nodes by					Avg. solving time	
					Lower bound	Theorem 5	Lemma 2	Notation 1	Lemma 3	H_2	*B&B*
Series 22	8	10	10	1.00	0.00	++	++	++	++	++	0.02
$C = 0.6$	10	10	10	1.00	0.00	++	++	++	++	++	0.01
$Q = 0.6$	12	10	10	1.00	0.00	++	++	++	++	++	0.01
$T_1 = 15$	14	10	10	1.00	0.00	++	++	++	++	++	0.01
$W_1 = 6$	16	10	10	1.00	0.00	++	++	++	++	++	0.02
$T_2 = 1.8\,T_1$	18	10	10	1.00	0.00	++	++	++	++	++	0.03
$W_2 = 1.6\,W_1$	20	10	10	1.00	0.00	++	++	++	++	++	0.03
Series 23	8	10	10	1.00	0.00	++	++	++	++	++	0.02
$C = 0.6$	10	10	10	1.00	0.00	++	++	++	++	++	0.02
$Q = 0.6$	12	10	10	1.00	0.00	++	++	++	++	++	0.01
$T_1 = 20$	14	10	10	1.00	0.00	++	++	++	++	++	0.01
$W_1 = 6$	16	10	10	1.00	0.00	++	++	++	++	++	0.01
$T_2 = 1.4\,T_1$	18	10	10	1.00	0.00	++	++	++	++	++	0.02
$W_2 = 1.6\,W_1$	20	10	10	1.00	0.00	++	++	++	++	++	0.02
Series 24	8	10	10	1.00	0.00	++	++	++	++	++	0.01
$C = 0.6$	10	10	10	1.00	0.00	++	++	++	++	++	0.02
$Q = 0.6$	12	10	10	1.00	0.00	++	++	++	++	++	0.01
$T_1 = 20$	14	10	10	1.00	0.00	++	++	++	++	++	0.01
$W_1 = 6$	16	10	10	1.00	0.00	++	++	++	++	++	0.01
$T_2 = 1.8\,T_1$	18	10	10	1.00	0.00	++	++	++	++	++	0.02
$W_2 = 1.6\,W_1$	20	10	10	1.00	0.00	++	++	++	++	++	[illegible]

References

[1] Jalali Naini, S.G., Aryanezhad, M.B., Jabbarzadeh, A., & Babaei H., *"Condition based maintenance for two-component systems with reliability and cost considerations,"* International Journal of Industrial Engineering & Production Research, Vol. 20, No. 3, (2009), pp. 107-116.

[2] Riahi, M. and Ansarifard, M., *"Maintenance improvement of ball bearings for industrial applications,"* International Journal of Industrial Engineering & Production

Research, Vol. 19, No. 7, (2008), pp. 123-126.

[3] Ma, Y., Chu, C., & Zuo, C., "*A survey of scheduling with deterministic machine availability constraints*," Computers & Industrial Engineering, Vol. 58, (2009), PP. 199-211.

[4] Molaee, E., & Moslehi, G., *"Minimizing the Number of Tardy Jobs on a Single Machine with an Availability Constraint,"* Journal of Industrial Engineering, Vol. 2014, (2014), PP. 1-13.

[5] Molaee, E., Moslehi, G., & Reisi, M., *"Minimizing maximum earliness and number of tardy jobs in the single machine scheduling problem with availability constraint,"* Computers & Mathematics with Applications, Vol. 62, (2011), PP. 3622-3641.

[6] Noroozi, A., Molla-Alizadeh-Zavardehi, S., & Mokhtari, H., *"A fixed and flexible maintenance operations planning optimization in a parallel batch machines manufacturing system,"* International Journal of Industrial Engineering & Production Research, Vol. 27, (2016), PP. 167-178.

[7] Yang, D.L., Hung, C.L., & Hsu, J., *"Minimizing the makespan in a single machine scheduling problem with a flexible maintenance activity,"* Journal of the Chinese Institute of Engineers, Vol. 19, (2002), PP. 63-66.

[8] Chen, J.S., *"Using integer programming to solve the machine scheduling with a flexible maintenance activity,"* Journal of Statistics and Management Systems, Vol. 9, (2006), PP. 87-104.

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

[10] Chen, J.S., *"Scheduling of nonresumable jobs and flexible maintenance activities on a single machine to minimize makespan,"* European Journal of Operational Research, Vol. 190, (2008), PP. 90-102.1

[11] Yang, S.L., Ma, Y., Xu, D.L., & Yang, J.B., *"Minimizing total completion time on a single machine with a flexible maintenance activity,"* Computers & Operations Research, Vol. 38, (2011), PP. 755-770.

[12] Ganji, F., Moslehi, G., & Ghalebsaz-Jeddi, B., *"Minimizing maximum earliness in single machine scheduling with flexible maintenance time,"* Scientia Iranica, Vol. 24, (2017), PP. 2082-2094.

[13] Qi, X., Chen, T., & Tu, F., *"Scheduling the maintenance on a single machine,"* Journal of the Operational Research Society, Vol. 50, (1999), PP. 1071-1078.

[14] Graves, G.H., & Lee. C.Y., *"Scheduling maintenance and semiresumable jobs on a single machine,"* Naval Research Logistics, Vol. 46, (1999), PP. 845-863.

[15] Mosheiov, G., & Sarig, A., *"Scheduling a maintenance activity to minimize total weighted completion times,"* Computers & Mathematics with Applications, Vol. 57, (2009), PP. 619-623.

[16] Low, C., Ji, M., Hsu, C.J., & Su, C.T., *"Minimizing the makespan in a single machine scheduling problems with flexible and periodic maintenance,"* Applied Mathematical Modelling, Vol. 34, (2010), PP. 334–342.

[17] Sbihi, M., & Varnier C., *"Single machine scheduling with periodic and flexible periodic maintenance to minimize maximum tardiness,"* Computers & Industrial Engineering, Vol. 55, (2008), PP. 830-840.

[18] Yang, S.J., & Yang, D.L., *"Minimizing the total completion time in single machine scheduling with aging/deteriorating effects and deteriorating maintenance activities,"* Computers & Mathematics with Applications, Vol. 60, (2010), PP. 2161-2169.

[19] Xu, D., & Yin, Y., *"Scheduling jobs under increasing linear machine maintenance time,"* Journal of Scheduling, Vol. 13, (2010), PP. 443–449.

[20] Yang, S.J., Yang, D.L., & Cheng, T.C.E., *"Single machine due-window assignment and scheduling with job-dependent aging effects and deteriorating maintenance,"* Computers & Operations Research, Vol. 37, (2010), PP. 1510–1514.

[21] Mashkani, O., & Moslehi, G., *"Minimizing the total completion time in a single machine scheduling problem under bimodal flexible periodic availability constraints,"* International Journal of Computer Integrated Manufacturing, Vol. 29, (2016), PP. 323-341.

[22] Lee, C.Y. *"Machine scheduling with an availability constraint,"* Journal of Global Optimization, Vol. 9,(1996), PP. 363–382.

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

[24] Moslehi, G., & Jafari, A., *"Minimizing the number of tardy jobs under piecewise-linear deterioration,"* Computers & Industrial Engineering, Vol. 59, (2010), PP. 573-584.

Follow This Article at The Following Site

Moslehi G, Mashkani O. Minimizing the Number of Tardy Jobs in the Single Machine Scheduling Problem under Bimodal Flexible and Periodic Availability Constraints. IJIEPR. 2018; 29 (1) :15-34
URL: http://ijiepr.iust.ac.ir/article-1-689-en.html