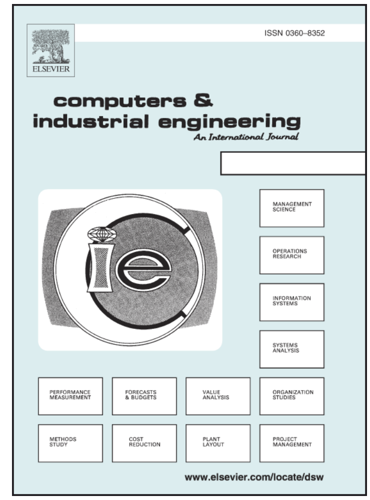
Accepted Manuscript



Parallel Machine Scheduling with Maintenance Activities

Ik Sun Lee

PII: S0360-8352(16)30360-6
DOI: http://dx.doi.org/10.1016/j.cie.2016.09.020
Reference: CAIE 4476

To appear in: *Computers & Industrial Engineering*

Received Date: 14 August 2013
Revised Date: 18 September 2016
Accepted Date: 19 September 2016

Please cite this article as: Lee, I.S., Parallel Machine Scheduling with Maintenance Activities, *Computers & Industrial Engineering* (2016), doi: http://dx.doi.org/10.1016/j.cie.2016.09.020

Parallel Machine Scheduling with Maintenance Activities

Ik Sun Lee

School of Business, Dong-A University, Seo-gu, Busan 602-760, Korea

Abstract

This paper considers a problem of scheduling on parallel machines where each machine requires maintenance activity once over a given time window. The objective is to find a coordinated schedule for jobs and maintenance activities to minimize the scheduling cost represented by either one of several objective measures including makespan, (weighted) sum of completion times, maximum lateness and sum of lateness. The problem is proved to be NP-hard in the strong sense in each case of the objective measures. Some restricted cases of the problem are also characterized for their complexities, for which the associated dynamic programming algorithms are derived.

Keywords: Scheduling; maintenance; parallel machine; problem complexity;

* Corresponding author. Tel.: +82-51-200-7425.
E-mail addresses: lis1007@dau.ac.kr (Ik Sun Lee)

Parallel Machine Scheduling with Maintenance Activities

Abstract

This paper considers a problem of scheduling on parallel machines where each machine requires maintenance activity once over a given time window. The objective is to find a coordinated schedule for jobs and maintenance activities to minimize the scheduling cost represented by either one of several objective measures including makespan, (weighted) sum of completion times, maximum lateness and sum of lateness. The problem is proved to be NP-hard in the strong sense in each case of the objective measures. Some restricted cases of the problem are also characterized for their complexities, for which the associated dynamic programming algorithms are derived.

Keywords: Scheduling; maintenance; parallel machine; problem complexity;

1. Introduction and Problem Description

The majority of machine scheduling models often assume that machines are available all the time for processing jobs over their associated planning horizon. However, this assumption is not realistic in many manufacturing situations, since machines require maintenance activity periodically to prevent malfunctions. During the associated maintenance activity machines are not available for processing jobs. Maintenance encompasses activities including installation, vehicles, equipment, or some physical assets enabling effective work. Preventive maintenance is an activity that a priori prevents potential faults resulting in malfunctions and also prevents critical non-availability of the system. Note that maintenance costs cover a big percentage of the total operating costs, making it very reasonable to include maintenance activities in the production schedule (Ángel-Bello *et al.*, 2011).

In the airline industry, planned maintenance activities can reduce production time by as much as 15% (Laalaoui and M'Hallah, 2016). Moreover, especially in semiconductor manufacturing, it is often observed that machines are idle while waiting for maintenance personnel to do preventive maintenance, even though jobs are waiting. Thus, the operations managers have to create their production schedule carefully so as to minimize their costs while avoiding unexpected resource unavailability. Obviously, careful coordination between maintenance activity and job processing would result in a better schedule, which is the motivation for this study. Here, the authors consider a coordinated scheduling model that takes into account such associated machine maintenance activities.

In the literature, scheduling problems with maintenance activities incorporated can be classified into "*fixed*" and "*coordinated*" models. The first model considers the maintenance activity durations, which are known and fixed in advance, so that the starting and completion times of the maintenance activity are given. The problem of scheduling jobs with this type of maintenance has often been referred to in the literature as "*scheduling with machine availability constraints*". Ángel-Bello *et al.* (2011), Hfaiedh *et al.* (2015), Laalaoui and M'Hallah (2016), Molaee *et al.* (2011), and Sadfi *et al.* (2005) have studied various single machine problems subject to various types of machine availability constraints. Fu *et al.* (2011), Gedik *et al.* (2016), Liao and Sheen (2008), Mellouli *et al.* (2009), and Wang and Cheng (2015) have studied various parallel machine problems allowing various types of unavailable intervals for machines. Cheng and Wang (1999, 2000), Kubiak *et al.* (2002), Kubzin *et al.* (2009) and Lee (1997, 1999) have studied a two machine flow shop problem allowing various types of unavailable intervals for machines.

The second model is concerned with simultaneously determining when to conduct each maintenance activity and when to process each job. Some research has been conducted on scheduling maintenance activities and jobs jointly. For example, Graves and Lee (1999) and Cassady and Kutanoglu (2003) have studied single machine problems allowing maintenance activities to be scheduled jointly with jobs. Aggoune (2004) has studied a flowshop machine problem allowing maintenance activities to be made within any given time window. Costa *et al.* (2016), Lee and Chen (2000), and Sun and Li (2010) have studied some parallel machine problems, subject to the constraint that maintenance activity on each machine should be made within a given time window. Specifically, Lee and Chen (2000) have studied a parallel machine problem to minimize the weighted sum of completion times of jobs. They have proved that the problem is NP-hard and have derived a branch and bound algorithm based on the column generation approach. Sun and Li (2010) have researched two two-machine parallel machines with the makespan or sum of completion times. Costa *et al.* (2016) have developed a genetic algorithm for a parallel machine problem.

This paper considers a coordinated scheduling model on parallel machines where each machine requires maintenance activity once over a given time window, as in Lee and Chen (2000). Moreover, two different maintenance activities are considered. The first one allows more than one machine to be put under maintenance simultaneously if necessary and is called "*independent case*". The second one, called "*dependent case*", allows only one machine to be put under maintenance at any time point due to insufficient maintenance resources (equipment or person); hence, the maintenance activity duration on machines cannot be overlapped onto each other. The model proposed here considers several different objectives of minimizing scheduling costs, each being represented by either one of several objective measures including makespan, (weighted) sum of completion times, maximum lateness and sum of lateness. Each of these scheduling problems is proved to be NP-hard in the strong sense and then some solution properties are characterized. Therewith, solution algorithms are derived using a dynamic programming (DP) approach. A few restricted cases of the problems are also analyzed for their complexities.

The proposed problem is stated in detail as follows: there are n jobs available at time zero to be scheduled on m identical parallel machines without preemption. Maintenance on each machine must be completed exactly once within the given time length T, that is, during the given time window [0, T], where the maintenance activity requires a maintenance time length t, while any job processing is allowed after time T. It is assumed that $T \geq t$ and $T \geq mt$ in the independent and dependent cases, respectively. processing time, weight, due date, and completion time of job j are denoted by p_j, w_j, d_j and C_j , respectively. It is assumed that t, T, p_j's, w_j's and d_j's have integer values. Moreover, this paper does not allow any preemption, so that a job should not be allowed to start until completing its associated maintenance activity if there is not enough time to complete any job processing before starting maintenance activity on the machine, which may incur an occurrence of machine idle time.

The standard classification scheme for scheduling problems (Pinedo (1995)) $\alpha\,|\,\beta\,|\,\gamma$ is adapted in this paper where α indicates the scheduling environment, β describes the job characteristics or restrictive requirements, and γ defines the objective function to be minimized. Accordingly, the proposed problem is represented by an identical parallel machines problem with α = "P". For β, the problem considers "*ind*", "*dep*", "$p_j=p$", "$d_j=d$" and "$m=q$" constraints, where "*ind*", "*dep*", "$p_j=p$", "$d_j=d$" and "$m=q$" indicate the independent case, the dependent case, all identical processing times case, all identical due dates case, and the q identical parallel machines case, respectively. Moreover, for γ, the objective function of the proposed problem may be represented by one of the following:

$C_{max} = \max_{1\leq j\leq n} C_j$ (makespan),

$\sum_{j=1}^{n} C_j$ = sum of completion times,

$\sum_{j=1}^{n} w_j C_j$ = weighted sum of completion times,

$L_{max} = \max_{1\leq j\leq n}\{C_j - d_j,\ 0\}$ (maximum lateness),

$\sum_{j=1}^{n} L_j = \sum_{j=1}^{n} \max(C_j - d_j, 0)$ (sum of lateness).

Table 1 provides the complexities of all of the tested scheduling problems associated with various objective measures. In the table, the complexity orders represent the computational time complexities of the associated DP algorithms, which are derived in Sections 2.1 and 4, where R_1, R_2, R_3 and R_4 are derived in Theorems 10 and 11 in Section 2.

>> Insert Table 1 <<

2. General case analysis

This section will prove that the problems P|ind|γ and P|dep|γ are NP-hard in the strong sense, where $\gamma \in \{C_{max}, \sum_{j=1}^{n} C_j, \sum_{j=1}^{n} w_j C_j, L_{\max}, \sum_{j=1}^{n} L_j\}$.

Theorem 1. The problem P|ind|$C_{\max}$ is NP-hard in the strong sense.

Proof. The proof is done by reduction from the 3-Partition Problem (Garey and Johnson, 1979), which is known to be NP-hard in the strong sense. The 3-Partition Problem is stated as follows:

Given $3q$ elements being integer size $e_1,\ldots, e_{3q}$, where $\sum_{i=1}^{3q} e_i = qB$ and $B/4 < e_i < B/2$ for $i = 1,\ldots, 3q$, does there exist a partition $S_1,\ldots, S_q$ of the index set $\{1,\ldots, 3q\}$ such that $|S_j|=3$ and $\sum_{i \in S_j} e_i = B$ for $j = 1,\ldots, q$?

Now, consider the following instance of the problem P|ind|$C_{\max}$;

$n = 3q$, $m = q$, $t=qB$, $T = (q+1)B$,

$p_j = e_j$, $j = 1,\ldots, 3q$.

Moreover, define a threshold value, Q, as

$Q = B$

Then, it will be proved that there exists a feasible schedule for the problem instance satisfying the relation $C_{max} \le Q$ if and only if there exists a solution to the 3-Partition

Problem.

a) For the if-part; suppose that there are q disjoint sets $S_1,\ldots, S_q$ which comprise a solution to the 3-Partition Problem, such as $\{x_{11}, x_{12}, x_{13}\}$, $\{x_{21}, x_{22}, x_{23}\},\ldots, \{x_{q1}, x_{q2}, x_{q3}\}$, where $\{x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_{23},\ldots, x_{q1}, x_{q2}, x_{q3}\} = \{e_1, e_2, e_3, e_4, e_5, e_6,\ldots, e_{3q-2}, e_{3q-1}, e_{3q}\}$ and $\sum_{i=1}^{3} x_{ji} = B$ for $j = 1,\ldots, q$. Then, the associated job sets $S_1'=\{J_1, J_2, J_3\},\ldots, S_q'=\{J_{3q-2}, J_{3q-1}, J_{3q}\}$ have processing times $\{x_{11}, x_{12}, x_{13}\}$, $\{x_{21}, x_{22}, x_{23}\},\ldots, \{x_{q1}, x_{q2}, x_{q3}\}$, respectively. Consider the schedule π such that three jobs in S_k' and the maintenance activity are scheduled on machine k during $[0, B]$ and $[B, (q+1)B]$, respectively, for each $k =1,\ldots, q$, where the three jobs in each job set S_k' can be scheduled in arbitrary order without machine idle time allowed. The structure of the schedule π is depicted as in Figure 1. Then, the schedule π has the makespan $C_{max} =B=Q$.

>> Insert Figure 1 <<

b) For the only if-part; suppose a schedule π is feasible as satisfying the relation $C_{max} \leq Q$. Any job cannot be scheduled after completing maintenance activity on any machine in the schedule π, since the maintenance length t is larger than Q. Then, all the jobs should be completed before starting the maintenance activity on each machine in the schedule π, that is, completed on each machine within the time interval $[0, B]$, since $T - t =B$. Since there are q parallel machines and the total processing times of all the jobs is qB, the schedule π has no machine idle time. This implies that there exists a solution to the 3-Partition Problem. ■

Theorem 2. The problem P|dep|$C_{\max}$ is NP-hard in the strong sense.

Proof. The proof is similar to that of Theorem 1. ■

Theorem 3. The problem P|ind|$\sum_{j=1}^{n} C_j$ is NP-hard in the strong sense.

Proof. The proof is done by reduction from the Numerical 3-Dimensional Matching Problem (Garey and Johnson, 1979) which is known to be NP-hard in the strong sense. The problem is stated as follows;

Given three sets X, Y, Z of q positive integers X={x_1, x_2,…, x_q}, Y={y_1, y_2,…, y_q} and Z={z_1, z_2,…, z_q} such that $\sum_{i=1}^{q}(x_i + y_i + z_i) = qB$, decide if there exist one-to-one functions ϕ and ψ defined on the sets S_1, S_2,…, S_q such that $x_i + y_{\phi(i)} + z_{\psi(i)} = B$, for all $S_i = \{x_i, y_{\phi(i)}, z_{\psi(i)}\}$, i=1,…, q.

Now, consider the following instance of the problem P|ind|$\sum_{j=1}^{n} C_j$;

n = 3q, m = q, t=20qB, T = 4B(5q+2),

$p_j = B + x_j$, j = 1,…, q,

$p_j = 2B + y_{j-q}$, j = q+1,…, 2q,

$p_j = 4B + z_{j-2q}$, j = 2q+1,…, 3q.

Moreover, define a threshold value, Q, as

$$Q = 11qB + \sum_{i=1}^{q}(3x_i + 2y_i + z_i).$$

Then it will be proved that there exists a feasible schedule for the problem instance satisfying the relation $\sum_{j=1}^{n} C_j \le Q$ if and only if there exists a solution to the Numerical 3-Dimensional Matching Problem.

a) For the if-part; suppose that there are q disjoint sets S_1,…, S_q, which comprise a solution to the Numerical 3-Dimensional Matching Problem, such as {x_1, $y_{\phi(1)}$, $z_{\psi(1)}$},…, {x_q, $y_{\phi(q)}$, $z_{\psi(q)}$} where x_j + $y_{\phi(j)}$ + $z_{\psi(j)}$ = B, for j =1,…, q. Then, the associated job sets S'_1={J$_1$, $J_{q+\phi(1)}$, $J_{2q+\psi(1)}$},…, S'_q={J$_q$, $J_{q+\phi(q)}$, $J_{2q+\psi(q)}$} have the processing times {B+x_1, 2B+ $y_{\phi(1)}$, 4B+ $z_{\psi(1)}$},…, {B+x_q, 2B+ $y_{\phi(q)}$, 4B+ $z_{\psi(q)}$}, respectively. Then, the total processing time of the three jobs in S'_k is $p_k + p_{q+\phi(k)} + z_{2q+\psi(k)}$ = 7B+(x_k+ $y_{\phi(k)} + z_{\psi(k)}$) = 8$B$, for k =1,…, q. Consider the schedule π such that three jobs in S'_k and the maintenance activity are scheduled on machine k during [0, 8B] and [8B, 4B(5q+2)], respectively, for each k =1,…, q, where the three jobs in each job set S'_k are scheduled in SPT order without machine idle time allowed. The structure of the schedule π is depicted as in Figure 2. Then, the schedule π has the sum of completion times, $\sum_{j=1}^{3q} C_j$ =11qB + $\sum_{i=1}^{q}(3x_i + 2y_i + z_i)$=$Q$.

>> Insert Figure 2 <<

b) For the only if-part; suppose a schedule π is feasible as satisfying the relation $\sum_{j=1}^{n} C_j \le Q$. Any job cannot be scheduled after completing maintenance activity on any machine in the schedule π since the maintenance length t is larger than Q. Then, all the jobs should be completed before starting the maintenance activity on each machine in the schedule π, that is, completed on each machine within the time interval [0, 8B], since T - t =8B. Since there are q parallel machines and the total processing times of all the jobs is 8qB, the schedule π has no machine idle time.

Now, the following claims can be made for the schedule π;

Claim 1. In the schedule π, q jobs $J_{2q+1}, \ldots, J_{3q}$ are scheduled separately on q machines during [0, 8B]. Suppose, to the contrary, that there are two jobs J_i and J_j among $J_{2q+1}, \ldots, J_{3q}$ which are processed on the same machine during [0, 8B]. Then, the sum of their processing times is $8B + z_{i-2q} + z_{j-2q}$, which is larger than the interval length 8B, since $1 \le z_{i-2q}, z_{j-2q} < B$, which is contradiction.

Claim 2. In the schedule π, q jobs $J_{q+1}, \ldots, J_{2q}$ are scheduled separately on q machines during [0, 8B]. Suppose, to the contrary, that there are two jobs J_i and J_j among $J_{q+1}, \ldots, J_{2q}$ which are processed on the same machine during [0, 8B]. According to Claim 1, J_{2q+k} is already scheduled on the machine, so that the sum of processing times of three jobs J_i, J_j, and J_{2q+k} is $8B + y_{i-q} + y_{j-q} + z_k$, which is larger than the interval length 8B, since $1 \le y_{i-q}, y_{j-q} < B$, which is a contradiction.

Claim 3. In the schedule π, q jobs $J_1, \ldots, J_q$ should be scheduled separately on q machines during [0, 8B]. Suppose, to the contrary, that there are two jobs J_i and J_j among $J_1, \ldots, J_q$ which are processed on the same machine during [0, 8B]. According to Claims 1 and 2, J_{q+l} and J_{2q+k} are already scheduled on the machine, so that the sum of processing times of four jobs J_i, J_j, J_{q+l} and J_{2q+k} is $8B + x_i + x_j + y_l + z_k$, which is larger than the interval length 8B, since $1 \le x_i, x_j < B$, which is a contradiction.

According to Claims 1, 2 and 3, each set of exactly three jobs selected from among $J_1, \ldots, J_{3q}$ should be scheduled (in SPT order) on each machine during [0, 8B]. This implies the existence of a solution to the Numerical 3-Dimensional Matching Problem. ■

Theorem 4. The problem P|dep| $\sum_{j=1}^{n} C_j$ is NP-hard in the strong sense.

Proof. The proof is similar to that of Theorem 3. ■

Corollary 1. The problems P|ind| $\sum_{j=1}^{n} w_j C_j$ and P|dep| $\sum_{j=1}^{n} w_j C_j$ are also NP-hard in the strong sense.

Proof. According to the result of Theorem 3 and 4, it is obvious. ■

Theorem 5. The problems P|ind|$L_{\max}$, P|dep|$L_{\max}$, P|ind| $\sum_{j=1}^{n} L_j$ and P|dep| $\sum_{j=1}^{n} L_j$ are NP-hard in the strong sense even if all the due dates are identical.

Proof. Consider a special situation such that the value 0 is assigned to each due date d_j's, for j=1,…, n. Then, the problems P|ind|$L_{\max}$, P|dep|$L_{\max}$, P|ind| $\sum_{j=1}^{n} L_j$ and P|dep| $\sum_{j=1}^{n} L_j$ will be equivalent to the problems P|ind|$C_{\max}$, P|dep|$C_{\max}$, P|ind| $\sum_{j=1}^{n} C_j$ and P|dep| $\sum_{j=1}^{n} C_j$, respectively. This implies that these problems are NP-hard in the strong sense even if all the due dates are identical, since their special cases are NP-hard in the strong sense. ■

As proved above, the problems P|ind| γ and P|dep| γ are NP-hard in the strong sense, where $\gamma \in \{C_{max}, \sum_{j=1}^{n} C_j, \sum_{j=1}^{n} w_j C_j, L_{\max}, \sum_{j=1}^{n} L_j\}$. Now, this section provides non-dominated sequencing properties for the problems P|ind| γ as in Theorem 6, where $\gamma \in \{C_{max}, \sum_{j=1}^{n} C_j, \sum_{j=1}^{n} w_j C_j, L_{\max}, \sum_{j=1}^{n} L_j\}$.

Theorem 6. For the problems P|ind| γ, where $\gamma \in \{C_{max}, \sum_{j=1}^{n} C_j, \sum_{j=1}^{n} w_j C_j, L_{\max}, \sum_{j=1}^{n} L_j\}$, there exists an optimal schedule such that

(1) for $\gamma \in \{C_{max}, \sum_{j=1}^{n} C_j\}$, all the scheduled jobs on each machine, as depicted in Figure 3-a), are in SPT (Shortest Processing Time) order, and no machine is idle.

(2) for $\gamma \in \{L_{\max}, \sum_{j=1}^{n} L_j\}$, all the scheduled jobs on each machine, as depicted in Figure 3-a), are in EDD (Earliest Due Date) order, and no machine is idle.

(3) for $\gamma = \sum_{j=1}^{n} w_j C_j$, all the scheduled jobs in each of the sets B_k and A_k (k=1,…, m) are in WSPT (Weighted Shortest Processing Time) order, where B_k and A_k denote sets of all the scheduled jobs before and after the maintenance activity, respectively, on machine k. Moreover, no machine is idle, as depicted in Figure 3-b).

Proof. This can be proved using interchange arguments. ■

>> Insert Figure 3 <<

Furthermore, non-dominated sequencing properties for the problems P|dep|γ are provided as in Theorem 7, where $\gamma \in \{C_{max}, \sum_{j=1}^{n} C_j, \sum_{j=1}^{n} w_j C_j, L_{\max}, \sum_{j=1}^{n} L_j\}$.

Theorem 7. For the problems P|dep|γ, where $\gamma \in \{C_{max}, \sum_{j=1}^{n} C_j, \sum_{j=1}^{n} w_j C_j, L_{\max}, \sum_{j=1}^{n} L_j\}$, and B_k and A_k are defined as in Theorem 6, there exists an optimal schedule such that

(1) for $\gamma \in \{C_{max}, \sum_{j=1}^{n} C_j\}$, all the scheduled jobs in each of the sets B_k and A_k (k=1,…, m) are in SPT order.

(2) for $\gamma \in \{L_{\max}, \sum_{j=1}^{n} L_j\}$, all the scheduled jobs in each of the sets B_k and A_k (k=1,…, m) are in EDD order.

(3) for $\gamma = \sum_{j=1}^{n} w_j C_j$, all the scheduled jobs in each of the sets B_k and A_k (k=1,…, m) are in WSPT order.

Proof. This can be proved using interchange arguments. ■

Note that for the problems P|dep|γ, where $\gamma \in \{C_{max}, \sum_{j=1}^{n} C_j, \sum_{j=1}^{n} w_j C_j, L_{\max}, \sum_{j=1}^{n} L_j\}$, there may exist idle time on any machine in an optimal schedule. Since it is important to decide when to start a job-processing, when to start the maintenance activity and when to keep the machine idle, this paper provides the following theorem associated with the possible start-times of job-processing, maintenance activity and machine idle duration.

Theorem 8. For the problems P|dep|γ, where $\gamma \in \{C_{max}$, $\sum_{j=1}^{n} C_j$, $\sum_{j=1}^{n} w_j C_j$, $L_{\max}$, $\sum_{j=1}^{n} L_j$ }, there exists an optimal schedule such that for each job j (j=1,…, n) and machine k (k=1,…, m),

(1) the processing of job j starts at the completion time of another job or maintenance activity on a machine.

(2) the maintenance activity on machine k starts at the completion time of any job on machine k or the maintenance activity on another machine.

(3) there exists at most one idle time on each machine only before starting the maintenance activity.

Proof. For (1), if all the machines are under process of other jobs or under maintenance activity, then job j should wait until at least one machine becomes available. For (2), if machine k is under process of any job or if another machine is under maintenance activity, then the maintenance activity on the machine cannot be made due to the dependent case assumption. For (3), suppose that there exists an optimal schedule π^* which has at least one idle time on machine k. Therewith, the following three cases need be discussed.

Firstly, consider the case where machine idle time is inserted between two jobs j_1 and j_2 such that job j_1 precedes job j_2 on machine k. Then, a non-dominated schedule $\tilde{\pi}$ can be found by processing the two jobs consecutively without any idle time inserted, so that $\gamma_{\tilde{\pi}} \le \gamma_{\pi^*}$. This implies that the idle time can be eliminated in this case.

Secondly, consider the case where machine idle time is inserted between maintenance activity and job j such that the maintenance activity precedes job j on machine k. Then, a non-dominated schedule $\tilde{\pi}$ can be found by doing the maintenance activity and processing the job j consecutively without any idle time inserted on machine k, so that $\gamma_{\tilde{\pi}} \le \gamma_{\pi^*}$. This implies that the idle time can be eliminated in this case.

Thirdly, consider the case where machine idle time is inserted between job j and maintenance activity such that job j precedes the maintenance activity on machine k, where s_j^* and C_j^* denote the starting and completion times of job j, respectively, and $s_{M_i}^*$ and $C_{M_i}^*$ denote the starting and completion times of the maintenance activity on machine i, for i=1,…, m, respectively. Then, from the result of (2), a non-dominated schedule $\tilde{\pi}$ can be found by doing the maintenance activity on machine k during [$\tilde{s}_{M_k}$, $\tilde{C}_{M_k}$], where $\tilde{s}_{M_k} = s_{M_k}^* - \Delta$, $\tilde{C}_{M_k} = C_{M_k}^* - \Delta$ and $\Delta = \min\left\{s_{M_k}^* - C_j^*, s_{M_k}^* - \max_{1\le i\le m}\left\{C_{M_i}^* \middle| C_{M_i}^* \le s_{M_k}^*\right\}\right\}$, so that $\gamma_{\tilde{\pi}} \le \gamma_{\pi^*}$. If the relation $\Delta = \left(s_{M_k}^* - C_j^*\right)$ holds, then the machine idle time will be eliminated. However, if the relation $\Delta < \left(s_{M_k}^* - C_j^*\right)$ holds, then the machine idle time occurs at the amount, ($s_{M_k}^*$ - C_j^* - Δ). This implies that the machine idle time can appear only before starting maintenance activity on machine k. Furthermore, there exists at most one idle time on each machine, since there is at most one maintenance activity on each machine during the planning horizon. ■

Without loss of generality, this paper assumes that the maintenance activity on machine k is made at the k-th order among m machines. Then, based on the result of Theorem 8, more specific non-dominated properties associated with the sets B_k and A_k, and the amount of the idle time can be provided for the problems P|dep| γ as in Theorem 9, where $\gamma \in \{C_{max}, \sum_{j=1}^n C_j , \sum_{j=1}^n w_j C_j , L_{\max}, \sum_{j=1}^n L_j \}$.

Theorem 9. For the problems P|dep| γ, where $\gamma \in \{C_{max}, \sum_{j=1}^n C_j , \sum_{j=1}^n w_j C_j , L_{\max}, \sum_{j=1}^n L_j \}$, there exists an optimal schedule such that

(1) $Sum(B_k) \le Sum(B_{k+1})$ and $a_k \ge a_{k+1}$, for k=1,…, m-1, where $Sum(B_k) = \sum_{j\in B_k} p_j$, $a_k \ge 0$, $a_k \in Z$ and a_k denotes the number of scheduled jobs in A_k, for k =1,…, m.

(2) the amount of idle time on machine k cannot be larger than the value $(k-1)\cdot t$, for k=1,…, m.

(3) Considering the relations $y_k = \max\{Sum(B_k)+t,\ y_{k-1}+t\}$, for $k = 2,\ldots, m$, where $y_1 = Sum(B_1) + t$ and y_k denotes the start-time of the first-starting job in A_k (on machine k), that is, the completion time of the maintenance activity on machine k; if the relation $y_k = Sum(B_k) + t$ holds, then there is no idle time on machine k, while if the relation $y_k > Sum(B_k) + t$ holds, then there exists idle time on machine k.

Proof. For (1), this can be proved using interchange arguments. For (2), there is no idle time on machine 1; otherwise, a non-dominated schedule can be found by eliminating the idle time on machine 1. Then, consider the situation where idle time does not appear on machine j ($1 \le j < k$), but appears on machines $j+1,\ldots, k$ consecutively, as depicted in Figure 4. Since $Sum(B_j) \le Sum(B_{j+1}) \le \ldots \le Sum(B_k)$, the amount of the idle time on machine k cannot be larger than the value $(k-j)\cdot t \le (k-1)\cdot t$ (due to $1 \le j < k$). For (3), referring to Theorem 8 and Figure 4, it is obvious. ■

>> Insert Figure 4 <<

For the problems P|ind|γ and P|dep|γ, where $\gamma \in \{C_{max}, \sum_{j=1}^{n} C_j, \sum_{j=1}^{n} w_j C_j, L_{\max}, \sum_{j=1}^{n} L_j\}$, non-dominated properties are provided to be used to restrict the search space, as in Theorems 10 and 11.

Theorem 10. For the problems P|ind|γ, where $\gamma \in \{C_{max}, \sum_{j=1}^{n} C_j, \sum_{j=1}^{n} w_j C_j, L_{\max}, \sum_{j=1}^{n} L_j\}$, there exists an optimal schedule such that on each machine, $P = \sum_{j=1}^{n} p_j$ and $p_{\max} = \max_{1\le j\le n}\{p_j\}$,

(1) the last scheduled job in B_k is completed no later than the time $R_1 = \min\{(T-t),\ [P+(m-1)\cdot t - p_{\max}]/m + p_{\max}\}$, for $k=1,\ldots, m$.

(2) the first scheduled job in A_k starts no earlier than t and no later than (R_1+t), and the last scheduled job in A_k is completed no later than the time $R_2 = [P - p_{\max}]/m + t + p_{\max}$, for $k=1,\ldots, m$.

Proof. For (1), the first term (T-t) implies that the maintenance activity should be completed before time T, so that the last scheduled job in B_k should be completed before (T-t), and the second term $[P+(m-1)\cdot t-p_{\max}]/m+p_{\max}$ can be derived. For (2), since the maintenance activity on each machine requires time duration t, the first scheduled job in A_k starts no earlier than t. Moreover, since there does not exist idle time in an optimal schedule (as in Theorem 6), the first scheduled job in A_k starts no later than (R_1+t), from the result of (1). ■

Theorem 11. For the problems P|dep|γ, where $\gamma \in \{C_{max}$, $\sum_{j=1}^{n} C_j$, $\sum_{j=1}^{n} w_j C_j$, $L_{\max}$, $\sum_{j=1}^{n} L_j$ }, there exists an optimal schedule such that on each machine,

(1) the last scheduled job in B_k is completed no later than the time $R_3 = \min\{(T-(m-k+1)\cdot t), [P+(m-1)\cdot t+(m-2)p_{\max}]/m+p_{\max}\}$, for k=1,…, m.

(2) the first scheduled job in A_k starts no earlier than $k \cdot t$ and no later than $(T-(m-k)\cdot t)$, and the last scheduled job in A_k is completed no later than the time $R_4 = [P+(m-1)p_{\max}]/m+t+p_{\max}$, for k=1,…, m.

Proof. The proof is similar to that of Theorem 10. ■

In summary, for the problems P|ind|γ, the associated optimal schedule needs to satisfy the properties specified as in Theorems 6 and 10, while for the problems P|dep|γ, the associated optimal schedule needs to satisfy the properties specified as in Theorems 7, 8, 9 and 11, where $\gamma \in \{C_{max}$, $\sum_{j=1}^{n} C_j$, $\sum_{j=1}^{n} w_j C_j$, $L_{\max}$, $\sum_{j=1}^{n} L_j$ }.

2.1. DP algorithms

This section derives DP Algorithm A for the problems P|ind|γ based on Theorems 6 and 10, where $\gamma \in \{C_{max}$, $\sum_{j=1}^{n} C_j$, $L_{\max}$, $\sum_{j=1}^{n} L_j$ }.

DP Algorithm A

Indexing: Index all the jobs in SPT order for the problems P|ind|$C_{\max}$ and P|ind|$\sum_{j=1}^{n} C_j$, and index all the jobs in EDD order for the problems P|ind|$L_{\max}$ and P|ind| $\sum_{j=1}^{n} L_j$.

Value Function: $f_j(u_1,..., u_m, v_1,…, v_m)$ = minimum objective value of a partial schedule for jobs 1,…, j such that the total operation time (including job-processing or maintenance times) on machine k is u_k, for k =1,…, m, where if v_k=1, the maintenance activity on machine k is made during the time interval [0, u_k], or if v_k=0, the maintenance activity on machine k is not made during [0, u_k].

Boundary Condition: $f_0(0,…, 0, 0,…, 0) = 0$.

Optimal Solution Value: $\min_{\left\{(u_1,...,u_m) | t \le u_k \le R_2,\ k=1,...,m,\ \sum_{k=1}^{m} u_k = \left(\sum_{j=1}^{n} p_j + m \cdot t\right)\right\}} f_n(u_1,..., u_m, 1,...,1)$

Recurrence Relation: $f_j(u_1,..., u_m, v_1,…, v_m)$ =

a) for the problem P|ind|$C_{\max}$;

$$\begin{cases} \infty,\ if\ \max\{u_k \mid v_k = 0, 1 \le k \le m\} < p_j\ and\ \max\{u_k \mid v_k = 1, 1 \le k \le m\} < t + p_j \\ \infty,\ if\ \max\{u_k \mid v_k = 0, 1 \le k \le m\} > R_1 \\ \min\begin{cases} \min_{\{k | u_k \ge p_j,\ 1 \le k \le m\}} \{\max\{f_{j-1}(u_1,...,u_{k-1}, u_k - p_j, u_{k+1},...,u_m, v_1,...,v_m), u_k\}\} \\ \min_{\{k | T \ge u_k \ge t,\ v_k = 1,\ 1 \le k \le m\}} \{f_j(u_1,...,u_{k-1}, u_k - t, u_{k+1},...,u_m, v_1,...,v_{k-1}, 0, v_{k+1},...,v_m)\} \end{cases} \end{cases},$$

b) for the problem P|ind| $\sum_{j=1}^{n} C_j$;

$$\begin{cases} \infty,\ if\ \max\{u_k \mid v_k = 0, 1 \le k \le m\} < p_j\ and\ \max\{u_k \mid v_k = 1, 1 \le k \le m\} < t + p_j \\ \infty,\ if\ \max\{u_k \mid v_k = 0, 1 \le k \le m\} > R_1 \\ \min\begin{cases} \min_{\{k | u_k \ge p_j,\ 1 \le k \le m\}} \{f_{j-1}(u_1,...,u_{k-1}, u_k - p_j, u_{k+1},...,u_m, v_1,...,v_m) + u_k\} \\ \min_{\{k | T \ge u_k \ge t,\ v_k = 1,\ 1 \le k \le m\}} \{f_j(u_1,...,u_{k-1}, u_k - t, u_{k+1},...,u_m, v_1,...,v_{k-1}, 0, v_{k+1},...,v_m)\} \end{cases} \end{cases},$$

c) for the problem P|ind|$L_{\max}$;

$$\begin{cases} \infty,\ if\ \max\{u_k \mid v_k = 0, 1 \le k \le m\} < p_j\ and\ \max\{u_k \mid v_k = 1, 1 \le k \le m\} < t + p_j \\ \infty,\ if\ \max\{u_k \mid v_k = 0, 1 \le k \le m\} > R_1 \\ \min\begin{cases} \min_{\{k | u_k \ge p_j,\ 1 \le k \le m\}} \{\max\{f_{j-1}(u_1,...,u_{k-1}, u_k - p_j, u_{k+1},...,u_m, v_1,...,v_m), u_k - d_j\}\} \\ \min_{\{k | T \ge u_k \ge t,\ v_k = 1,\ 1 \le k \le m\}} \{f_j(u_1,...,u_{k-1}, u_k - t, u_{k+1},...,u_m, v_1,...,v_{k-1}, 0, v_{k+1},...,v_m)\} \end{cases} \end{cases},$$

d) for the problem P|ind| $\sum_{j=1}^{n} L_j$;

$$\begin{cases} \infty, \ if \ \max\{u_k \mid v_k = 0, 1 \le k \le m\} < p_j \ and \ \max\{u_k \mid v_k = 1, 1 \le k \le m\} < t + p_j \\ \infty, \ if \ \max\{u_k \mid v_k = 0, 1 \le k \le m\} > R_1 \\ \min \begin{cases} \min_{\{k \mid u_k \ge p_j, \ 1 \le k \le m\}} \{ f_{j-1}(u_1,...,u_{k-1}, u_k - p_j, u_{k+1},...,u_m, v_1,...,v_m) + u_k - d_j \} \\ \min_{\{k \mid T \ge u_k \ge t, \ v_k = 1, \ 1 \le k \le m\}} \{ f_j(u_1,...,u_{k-1}, u_k - t, u_{k+1},...,u_m, v_1,...,v_{k-1}, 0, v_{k+1},...,v_m) \} \end{cases} \end{cases} .$$

In each recurrence relation, the first term represents an infeasible case such that there is no space to schedule job j on any machine. The second term represents that the associated partial schedule (u_1,..., u_m, v_1,…, v_m) can be dominated according to Theorem 10. The third term represents that job J_j is scheduled during [u_k-p_j, u_k] on machine k, as depicted in Figure 5-a). The fourth term represents that the maintenance activity on machine k is made during [u_k-t, u_k], as depicted in Figure 5-b).

In DP Algorithm A, there are a total of $O(n(R_1+R_2)^m)$ states and the function value of each state is calculated in $O(m)$ time, so that DP Algorithm A can be applied to the problems P|ind| γ in the complexity order of $O(nm(R_1+R_2)^m)$, where $\gamma \in$ {C_{max}, $\sum_{j=1}^{n} C_j$, $L_{\max}$, $\sum_{j=1}^{n} L_j$ }.

>> Insert Figure 5 <<

Now, DP Algorithm B is also derived for the problem P|ind| $\sum_{j=1}^{n} w_j C_j$ based on Theorems 6 and 10. Denote by x_k the completion time of the last-starting job in B_k, and by y_k the start-time of the first-starting job in A_k, and by z_k the completion time of the last-starting job in A_k, where $0 \le x_k \le R_1$, $t \le y_k \le R_1+t$, $t \le z_k \le R_2$, for k =1,…, m, as in Theorem 10. Then, the maintenance activity on machine k is made during [y_k - t, y_k].

DP Algorithm B

Indexing: Index all the jobs in WSPT order.

Value Function: $f_j(x_1,..., x_m, y_1,…, y_m, z_1,…, z_m)$ = minimum objective value of a partial schedule for jobs 1,…, j such that the scheduled jobs in B_k are processed during [0, x_k] and the scheduled jobs in A_k are processed during [y_k, z_k], for k =1,…, m.

Boundary Condition: $f_0(0,…, 0, y_1,…, y_m, z_1,…, z_m) = 0$, if $t \le y_k = z_k \le R_1+t$ for k =1,…, m,

$f_0(x_1,..., x_m, y_1,…, y_m, z_1,…, z_m) = \infty$, if $x_k > 0$ or $y_k < z_k$ for $k =1,…, m$.

Optimal Solution Value:

$$\min_{\substack{\{(x_1,...,x_m,y_1,...,y_m,z_1,...,z_m) \mid 0\le x_k \le R_1,\ t\le y_k \le R_1+t,\ t\le z_k \le R_2, \\ y_k \le z_k,\ x_k+t=y_k,\ k=1,...,m,\ \sum_{k=1}^{m}(x_k+(z_k-y_k))=\sum_{j=1}^{n} p_j\}}} f_n(x_1,...,x_m,y_1,...,y_m,z_1,...,z_m)$$

Recurrence Relation: $f_j(x_1,..., x_m, y_1,…, y_m, z_1,…, z_m)$ =

$$\begin{cases} \infty, \text{ if } \max_{1\le k\le m}\{x_k\} < p_j \text{ and } \max_{1\le k\le m}\{z_k - y_k\} < p_j \\ \min\begin{cases} \min_{\{k \mid x_k \ge p_j,\ 1\le k\le m\}}\{f_{j-1}(x_1,...,x_{k-1},x_k - p_j, x_{k+1},...,x_m, y_1,...,y_m, z_1,...,z_m) + w_j x_k\} \\ \min_{\{k \mid (z_k - y_k) \ge p_j,\ 1\le k\le m\}}\{f_{j-1}(x_1,...,x_m, y_1,...,y_m, z_1,...,z_{k-1}, z_k - p_j, z_{k+1},...,z_m) + w_j z_k\} \end{cases} \end{cases}.$$

The relation $x_k = 0$ represents the situation where any job cannot be scheduled in B_k, while the relation $x_k > 0$ represents the situation where more jobs should be scheduled in B_k. Furthermore, the relation $y_k = z_k$ represents the situation where any job cannot be scheduled in A_k, while the relation $y_k < z_k$ represents the situation where more jobs should be scheduled in A_k.

In the equation associated with the optimal solution value, the relation $x_k + t = y_k$ represents that there does not exist machine idle time on machine k in an optimal schedule as in Theorem 6.

In the recurrence relation, the first term represents an infeasible case such that there is no space to schedule job j on any machine. The second term represents that job J_j is scheduled in B_k during $[x_k-p_j, x_k]$, as depicted in Figure 6-a). The third term represents that job J_j is scheduled in A_k during $[z_k-p_j, z_k]$, as depicted in Figure 6-b).

In DP Algorithm B, there are a total of $O(nR_1^{2m}R_2^{m})$ states and the function value of each state is calculated in $O(m)$ time, so that DP Algorithm B can be applied to the problem P|ind|$\sum_{j=1}^{n} w_j C_j$ in the complexity order of $O(nmR_1^{2m}R_2^{m})$.

>> Insert Figure 6 <<

Moreover, DP Algorithm C is derived for the problems P|dep|γ based on Theorems 7, 8, 9 and 11, where $\gamma \in \{C_{max}, \sum_{j=1}^{n} C_j, \sum_{j=1}^{n} w_j C_j, L_{\max}, \sum_{j=1}^{n} L_j\}$. Recall that the maintenance activity on machine k is made at the k-th order among m machines. From the result of Theorem 11, $0 \le x_k \le R_3$, $k \cdot t \le y_k \le (T-(m-k)\cdot t)$, $k \cdot t \le z_k \le R_4$, where x_k, y_k and z_k are defined as in DP Algorithm B. Then, the maintenance activity on machine k is made during $[y_k - t, y_k]$.

DP Algorithm C

Indexing: Index all the jobs in SPT order for the problems $P|dep|C_{\max}$ and $P|dep|\sum_{j=1}^{n} C_j$, and index all the jobs in WSPT order for the problem $P|dep|\sum_{j=1}^{n} w_j C_j$, and index all the jobs in EDD order for the problems $P|dep|L_{\max}$ and $P|dep|\sum_{j=1}^{n} L_j$.

Value Function: $f_j(x_1,..., x_m, y_1,\ldots, y_m, z_1,\ldots, z_m)$ = minimum objective value of a partial schedule for jobs $1,\ldots, j$ such that the scheduled jobs in B_k are processed during $[0, x_k]$ and the scheduled jobs in A_k are processed during $[y_k, z_k]$, for $k = 1,\ldots, m$.

Boundary Condition: $f_0(0,\ldots, 0, y_1,\ldots, y_m, z_1,\ldots, z_m) = 0$, if $k \cdot t \le y_k = z_k \le (T-(m-k)\cdot t)$ for $k = 1,\ldots, m$, $f_0(x_1,..., x_m, y_1,\ldots, y_m, z_1,\ldots, z_m) = \infty$, if $x_k > 0$ or $y_k < z_k$ for $k = 1,\ldots, m$.

Optimal Solution Value:

$$\min_{\substack{\{(x_1,\ldots,x_m,y_1,\ldots,y_m,z_1,\ldots,z_m) | 0\le x_k \le R_3,\ k\cdot t\le y_k \le (T-(m-k)\cdot t),\ k\cdot t \le z_k \le R_4,\ y_k \le z_k,\ x_k \le x_{k+1}, \\ y_k \le y_{k+1}-t,\ 0\le y_k - x_k - t \le (k-1)\cdot t,\ k=1,\ldots,m,\ \sum_{k=1}^{m}(x_k+(z_k-y_k)) = \sum_{j=1}^{n} p_j\}}} f_n(x_1,\ldots,x_m,y_1,\ldots,y_m,z_1,\ldots,z_m)$$

Recurrence Relation: $f_j(x_1,..., x_m, y_1,\ldots, y_m, z_1,\ldots, z_m) =$

a) for the problem $P|dep|C_{\max}$;

$$\begin{cases} \infty,\ if\ \max_{1\le k\le m}\{x_k\} < p_j\ and\ \max_{1\le k\le m}\{z_k - y_k\} < p_j \\ \min\begin{cases} \min_{\{k | x_k \ge p_j,\ 1\le k\le m\}}\{\max\{f_{j-1}(x_1,\ldots,x_{k-1},x_k-p_j,x_{k+1},\ldots,x_m,y_1,\ldots,y_m,z_1,\ldots,z_m),\ x_k\}\} \\ \min_{\{k | (z_k-y_k)\ge p_j,\ 1\le k\le m\}}\{\max\{f_{j-1}(x_1,\ldots,x_m,y_1,\ldots,y_m,z_1,\ldots,z_{k-1},z_k-p_j,z_{k+1},\ldots,z_m),\ z_k\}\} \end{cases} \end{cases},$$

b) for the problem $P|dep|\sum_{j=1}^{n} C_j$;

$$\begin{cases} \infty,\ if\ \max_{1\le k\le m}\{x_k\} < p_j\ and\ \max_{1\le k\le m}\{z_k - y_k\} < p_j \\ \min\begin{cases} \min_{\{k | x_k \ge p_j,\ 1\le k\le m\}}\{f_{j-1}(x_1,\ldots,x_{k-1},x_k-p_j,x_{k+1},\ldots,x_m,y_1,\ldots,y_m,z_1,\ldots,z_m) + x_k\} \\ \min_{\{k | (z_k-y_k)\ge p_j,\ 1\le k\le m\}}\{f_{j-1}(x_1,\ldots,x_m,y_1,\ldots,y_m,z_1,\ldots,z_{k-1},z_k-p_j,z_{k+1},\ldots,z_m) + z_k\} \end{cases} \end{cases},$$

c) for the problem $P|dep|\sum_{j=1}^{n} w_j C_j$;

$$\begin{cases} \infty, \; if \;\; \max_{1\le k\le m}\{x_k\} < p_j \;\; and \; \max_{1\le k\le m}\{z_k - y_k\} < p_j \\ \min\begin{cases} \min_{\{k \mid x_k \ge p_j, \; 1\le k\le m\}}\{f_{j-1}(x_1,\ldots,x_{k-1},x_k - p_j, x_{k+1},\ldots,x_m,y_1,\ldots,y_m,z_1,\ldots,z_m) + w_j x_k\} \\ \min_{\{k \mid (z_k - y_k)\ge p_j, \; 1\le k\le m\}}\{f_{j-1}(x_1,\ldots,x_m,y_1,\ldots,y_m,z_1,\ldots,z_{k-1},z_k - p_j, z_{k+1},\ldots,z_m) + w_j z_k\} \end{cases} \end{cases}.$$

d) for the problem P|dep|$L_{\max}$;

$$\begin{cases} \infty, \; if \;\; \max_{1\le k\le m}\{x_k\} < p_j \;\; and \; \max_{1\le k\le m}\{z_k - y_k\} < p_j \\ \min\begin{cases} \min_{\{k \mid x_k \ge p_j, \; 1\le k\le m\}}\{\max\{f_{j-1}(x_1,\ldots,x_{k-1},x_k - p_j, x_{k+1},\ldots,x_m,y_1,\ldots,y_m,z_1,\ldots,z_m), \; x_k - d_j\}\} \\ \min_{\{k \mid (z_k - y_k)\ge p_j, \; 1\le k\le m\}}\{\max\{f_{j-1}(x_1,\ldots,x_m,y_1,\ldots,y_m,z_1,\ldots,z_{k-1},z_k - p_j, z_{k+1},\ldots,z_m), \; z_k - d_j\}\} \end{cases} \end{cases}$$

,

e) for the problem P|dep| $\sum_{j=1}^{n} L_j$;

$$\begin{cases} \infty, \; if \;\; \max_{1\le k\le m}\{x_k\} < p_j \;\; and \; \max_{1\le k\le m}\{z_k - y_k\} < p_j \\ \min\begin{cases} \min_{\{k \mid x_k \ge p_j, \; 1\le k\le m\}}\{f_{j-1}(x_1,\ldots,x_{k-1},x_k - p_j, x_{k+1},\ldots,x_m,y_1,\ldots,y_m,z_1,\ldots,z_m) + x_k - d_j\} \\ \min_{\{k \mid (z_k - y_k)\ge p_j, \; 1\le k\le m\}}\{f_{j-1}(x_1,\ldots,x_m,y_1,\ldots,y_m,z_1,\ldots,z_{k-1},z_k - p_j, z_{k+1},\ldots,z_m) + z_k - d_j\} \end{cases} \end{cases}.$$

In the equation associated with the optimal solution value, the relation $x_k \le x_{k+1}$ implies the relation $Sum(B_k) \le Sum(B_{k+1})$ in Theorem 9, and the relation $0 \le y_k - x_k - t \le (k-1)\cdot t$ represents that the amount of idle time on machine k cannot be larger than the value $(k-1)\cdot t$ as in Theorem 9, and the relation $y_k \le y_{k+1} - t$ represents that the maintenance activity on machine k precedes that on machine (k+1).

In the recurrence relation, the first term represents an infeasible case such that there is no space to schedule job j on any machine. The second term represents that job J_j is scheduled in B_k during $[x_k - p_j, x_k]$, as depicted in Figure 6-a). The third term represents that job J_j is scheduled in A_k during $[z_k - p_j, z_k]$, as depicted in Figure 6-b).

In DP Algorithm C, there are a total of $O(nR_3^m R_4^m T^m)$ states and the function value of each state is calculated in $O(m)$ time, so that DP Algorithm C can be applied to the problems P|dep|γ in the complexity order of $O(nmR_3^m R_4^m T^m)$, where $\gamma \in \{C_{max}$, $\sum_{j=1}^{n} C_j$, $\sum_{j=1}^{n} w_j C_j$, $L_{\max}$, $\sum_{j=1}^{n} L_j$ $\}$.

This section has derived three DP Algorithms A, B and C for the problems P|ind|γ and P|dep|γ, where $\gamma \in \{C_{max}$, $\sum_{j=1}^{n} C_j$, $\sum_{j=1}^{n} w_j C_j$, $L_{\max}$, $\sum_{j=1}^{n} L_j$ $\}$. Some of their restricted cases will be characterized in the following sections.

3. Fixed number of machines case

This section shows first that the proposed scheduling problems with only two machines, P|ind, m=2|γ and P|dep, m=2|γ are NP-hard, where $\gamma \in \{C_{max}, \sum_{j=1}^{n} C_j, \sum_{j=1}^{n} w_j C_j, L_{\max}, \sum_{j=1}^{n} L_j\}$.

Theorem 12. The problems P|ind, m=2|$\sum_{j=1}^{n} C_j$ and P|dep, m=2|$\sum_{j=1}^{n} C_j$ are NP-hard.

Proof. Lee and Chen (2000) have proved that the problems P|ind, m=2|$\sum_{j=1}^{n} C_j$ and P|dep, m=2|$\sum_{j=1}^{n} C_j$ are NP-hard by reduction from the Partition Problem.

Corollary 2. The problems P|ind, m=2|$\sum_{j=1}^{n} w_j C_j$ and P|dep, m=2|$\sum_{j=1}^{n} w_j C_j$ are also NP-hard.

Proof. According to the result of Theorem 12, it is obvious. ■

Theorem 13. The problems P|ind, m=2|$C_{\max}$ and P|dep, m=2|$C_{\max}$ are NP-hard.

Proof. The proof is similar to that of Lee and Chen (2000) associated with Theorem 12. ■

Theorem 14. The problems P|ind, m=2|$L_{\max}$, P|dep, m=2|$L_{\max}$, P|ind, m=2|$\sum_{j=1}^{n} L_j$ and P|dep, m=2|$\sum_{j=1}^{n} L_j$ are NP-hard even if all the due dates are identical.

Proof. The proof is similar to that of Theorem 5. ■

The results of Theorems 12, 13 and 14 imply that the problems P|ind, m=q($\geq$2)|γ and P|dep, m=q($\geq$ 2)|γ are NP-hard, where $\gamma \in \{C_{max}, \sum_{j=1}^{n} C_j, \sum_{j=1}^{n} w_j C_j, L_{\max}, \sum_{j=1}^{n} L_j\}$. Now, it will be proved that the problems P|ind, m=q($\geq$2)|γ and P|dep,

m=q(≥ 2)|γ are NP-hard in the ordinary sense, where $\gamma \in \{C_{max}$, $\sum_{j=1}^{n} C_j$, $\sum_{j=1}^{n} w_j C_j$, $L_{\max}$, $\sum_{j=1}^{n} L_j$ }.

Theorem 15. The problems P|ind, m=q(≥ 2)|γ and P|dep, m=q(≥ 2)|γ are NP-hard in the ordinary sense, where $\gamma \in \{C_{max}$, $\sum_{j=1}^{n} C_j$, $\sum_{j=1}^{n} w_j C_j$, $L_{\max}$, $\sum_{j=1}^{n} L_j$ }.

Proof. It has been shown that those problems are NP-hard as shown in Theorems 12, 13, 14 and Corollary 2. DP Algorithm A can be applied to the problems P|ind, m=q(≥ 2)|γ in the pseudo-polynomial complexity order of $O(nq(R_1+R_2)^q)$, where $\gamma \in \{C_{max}$, $\sum_{j=1}^{n} C_j$, $L_{\max}$, $\sum_{j=1}^{n} L_j$ }. DP Algorithm B can be applied to the problem P|ind, m=q(≥ 2)|$\sum_{j=1}^{n} w_j C_j$ in the pseudo-polynomial complexity order of $O(nqR_1^{2q}R_2^q)$. DP Algorithm C can be applied to the problems P|dep, m=q(≥ 2)|γ in the pseudo-polynomial complexity order of $O(nqR_3^qR_4^qT^q)$, where $\gamma \in \{C_{max}$, $\sum_{j=1}^{n} C_j$, $\sum_{j=1}^{n} w_j C_j$, $L_{\max}$, $\sum_{j=1}^{n} L_j$ }. Thus, those problems are NP-hard in the ordinary sense. ■

4. Identical processing time case

This section considers the case when all the processing times are identical. Firstly, for the problems P|ind, p_j=p|γ non-dominated properties are provided as in Theorem 16, where $\gamma \in \{C_{max}$, $\sum_{j=1}^{n} C_j$, $\sum_{j=1}^{n} w_j C_j$, $L_{\max}$, $\sum_{j=1}^{n} L_j$ }.

Theorem 16. For the problems P|ind, p_j=p|γ, where $\gamma \in \{C_{max}$, $\sum_{j=1}^{n} C_j$, $\sum_{j=1}^{n} w_j C_j$, $L_{\max}$, $\sum_{j=1}^{n} L_j$ }, there exists an optimal schedule such that

(1) for $\gamma \in \{C_{max}$, $\sum_{j=1}^{n} C_j$ }, all the scheduled jobs on each machine are in arbitrary order without machine idle time allowed.

(2) for $\gamma = \sum_{j=1}^{n} w_j C_j$, all the scheduled jobs on each machine are in non-increasing weight order without machine idle time allowed.

(3) For $\gamma \in \{L_{\max}, \sum_{j=1}^{n} L_j\}$, all the scheduled jobs on each machine are in EDD order without machine idle time allowed.

Proof. This can be proved using interchange arguments. ■

Now, this paper derives an optimal Algorithm D for the problems P|ind, $p_j=p|\gamma$ based on Theorems 16, where $\gamma \in \{C_{max}, \sum_{j=1}^{n} C_j, \sum_{j=1}^{n} w_j C_j, L_{\max}, \sum_{j=1}^{n} L_j\}$.

Algorithm D

Indexing: index all the jobs in arbitrary order for the problems P|ind, $p_j=p|C_{\max}$ and P|ind, $p_j=p|\sum_{j=1}^{n} C_j$, and index all the jobs in non-increasing weight order for the problem P|ind, $p_j=p|\sum_{j=1}^{n} w_j C_j$, and index all the jobs in EDD order for the problems P|ind, $p_j=p|L_{\max}$ and P|ind, $p_j=p|\sum_{j=1}^{n} L_j$.

Schedule Construction

Step 1. Set k=1. If the relation $k \le \lfloor (T-t)/p \rfloor$ holds, then go to Step 2. Otherwise, go to Step 4.

Step 2. If $k \cdot m \le n$, then m jobs are scheduled on m machines separately and go to Step 3. Otherwise, go to Step 7.

Step 3. Set $k = k + 1$. If the relation $k \le \lfloor (T-t)/p \rfloor$ holds, then go to Step 2. Otherwise, go to Step 4.

Step 4. The maintenance activities on all the machines are made simultaneously during $[\lfloor (T-t)/p \rfloor p, \lfloor (T-t)/p \rfloor p + t]$.

Step 5. If $k \cdot m \le n$, then m jobs are scheduled on m machines separately and go to Step 6. Otherwise, go to Step 7.

Step 6. Set $k = k + 1$ and go to Step 5.

Step 7. The remaining $(n-(k-1)\cdot m)$ jobs are scheduled on m machines separately.

For the problems P|ind, p_j=$p|\gamma$, where $\gamma \in \{C_{max}, \sum_{j=1}^{n} C_j\}$, the complexity of the optimal Algorithm D is in the order of $O(n)$. For the problems P|ind, p_j=$p|\gamma$, where $\gamma \in \{\sum_{j=1}^{n} w_j C_j, L_{\max}, \sum_{j=1}^{n} L_j\}$, the complexity of the optimal Algorithm D is in the order of $O(n \log n)$.

Now, for the problems P|dep, p_j=$p|\gamma$, where $\gamma \in \{C_{max}, \sum_{j=1}^{n} C_j, \sum_{j=1}^{n} w_j C_j, L_{\max}, \sum_{j=1}^{n} L_j\}$, non-dominated sequencing properties are provided as in Theorem 17.

Theorem 17. For the problems P|dep, p_j=$p|\gamma$, where $\gamma \in \{C_{max}, \sum_{j=1}^{n} C_j, \sum_{j=1}^{n} w_j C_j, L_{\max}, \sum_{j=1}^{n} L_j\}$, there exists an optimal schedule such that

(1) for $\gamma \in \{C_{max}, \sum_{j=1}^{n} C_j\}$, all the scheduled jobs on each machine are in arbitrary order.

(2) for $\gamma = \sum_{j=1}^{n} w_j C_j$, all the scheduled jobs on each machine are in non-increasing weight order.

(3) For $\gamma \in \{L_{\max}, \sum_{j=1}^{n} L_j\}$, all the scheduled jobs on each machine are in EDD order.

Proof. This can be proved using interchange arguments. ■

Note that for the problems P|dep, p_j=$p|\gamma$, where $\gamma \in \{C_{max}, \sum_{j=1}^{n} C_j, \sum_{j=1}^{n} w_j C_j, L_{\max}, \sum_{j=1}^{n} L_j\}$, there may exist idle time in an optimal schedule, so that Theorem 8 in Section 2 still holds. Furthermore, non-dominated properties are provided as in Theorem 18, which is similar to Theorem 9 in Section 2.

Theorem 18. For the problems P|dep|γ, where $\gamma \in \{C_{max}, \sum_{j=1}^{n} C_j, \sum_{j=1}^{n} w_j C_j, L_{\max}, \sum_{j=1}^{n} L_j\}$, there exists an optimal schedule such that

(1) $b_k \le b_{k+1}$ and $a_k \ge a_{k+1}$, for k=1,…, m-1, where b_k and a_k denote the number of

scheduled jobs in B_k and A_k, respectively, and b_k, $a_k \geq 0$, and a_k, $b_k \in Z$, for k =1,…, m.

(2) the amount of idle time on machine k cannot be larger than the value $(k-1)\cdot t$, for k=1,…, m.

(3) Considering the relations $y_k = \max\{p\cdot b_k + t,\ y_{k-1}+t\}$, for k = 2,…, m, where $y_1 = p\cdot b_1 + t$, $k\cdot t \leq y_k \leq (T-(m-k)\cdot t)$ and y_k is defined as in Theorem 9; if the relation $y_k = p\cdot b_k + t$ holds, then there is no idle time on machine k, while if the relation $y_k > p\cdot b_k + t$ holds, then there exists idle time on machine k.

(4) the total number of all the combinations (y_1,…, y_m) cannot be larger than the value $(n+1)^m$.

Proof. For (1), (2) and (3), the proof is similar to that of Theorem 9. For (4), the following relations hold;

$$y_1 = p\cdot b_1 + t,$$

$$y_2 = \max\{p\cdot b_2 + t,\ y_1 + t\} = \max\{p\cdot b_2 + t,\ p\cdot b_1 + 2t\}$$

$$y_3 = \max\{p\cdot b_3 + t,\ y_2 + t\} = \max\{p\cdot b_3 + t,\ \max\{p\cdot b_2 + t, p\cdot b_1 + 2t\} + t\}$$

$$\cdots$$

$$y_k = \max\{p\cdot b_k + t,\ y_{k-1}+t\} = \max\{p\cdot b_k + t, \max\{p\cdot b_{k-1} + t, \max\{p\cdot b_{k-2} + t, \max\{....\} + t\} + t\} + t\}.$$

Then, the total number of values of y_1 is (n+1), since $0 \leq b_1 \leq n$. Given any value of y_1, the total number of values of y_2 cannot be larger than the value (n+1), since $0 \leq b_2 \leq n$, so that the total number of all the possible (y_1, y_2) pairs cannot be larger than the value $(n+1)^2$. Furthermore, if the values of y_1,…, y_{k-1} are given at any values, then the total number of values of y_k cannot be larger than the value (n+1), since $0 \leq b_k \leq n$, for k=2,…, m, so that the total number of all the possible (y_1,…, y_k) tuples cannot be larger than the value $(n+1)^k$. Therefore, the total number of all the combinations (y_1,…, y_m) cannot be larger than the value $(n+1)^m$. ■

In summary, the associated optimal schedule needs to satisfy the properties specified as in Theorems 17 and 18. Now, this paper derives DP Algorithm E for the problems

P|dep, $p_j=p|\gamma$ based on Theorems 8, 11, 17 and 18, where $\gamma \in \{C_{max}, \sum_{j=1}^{n} C_j, \sum_{j=1}^{n} w_j C_j, L_{\max}, \sum_{j=1}^{n} L_j\}$. b_k, a_k and y_k are defined as in Theorem 18, where $y_1 = p \cdot b_1 + t$, $y_r = \max\{p \cdot b_r + t, y_{r-1} + t\}$, for r =2,…, m, $k \cdot t \le y_k \le (T - (m-k) \cdot t)$, and b_k, $a_k \ge 0$, and a_k, $b_k \in Z$, for k =1,…, m, and $\sum_{k=1}^{m} (a_k + b_k) = n$. Then, the maintenance activity on machine k is made during $[y_k - t, y_k]$.

DP Algorithm E

Indexing: index all the jobs in arbitrary order for the problems P|dep, $p_j=p|C_{\max}$ and P|dep, $p_j=p|\sum_{j=1}^{n} C_j$, and index all the jobs in non-increasing weight order for the problem P|dep, $p_j=p|\sum_{j=1}^{n} w_j C_j$, and index all the jobs in EDD order for the problems P|dep, $p_j=p|L_{\max}$ and P|dep, $p_j=p|\sum_{j=1}^{n} L_j$.

Value Function: $f_j(b_1,..., b_m, y_1,…, y_m, a_1,…, a_m)$ = minimum objective value of a partial schedule for jobs 1,…, j such that b_k jobs in B_k are processed during $[0, p \cdot b_k]$ and a_k jobs in A_k are processed during $[y_k, y_k + p \cdot a_k]$, for k =1,…, m.

Boundary Condition: $f_0(0,…, 0, y_1,…, y_m, 0,…, 0) = 0$, if $y_1 = p \cdot b_1 + t$, $y_r = \min\{p \cdot b_r + t, y_{r-1} + t\}$ for r =2,…, m, $k \cdot t \le y_k \le (T - (m-k) \cdot t)$ and $0 \le b_k \le n$, for k =1,…, m.

Optimal Solution Value:

$$\min_{\substack{\{(b_1,\dots,b_m,y_1,\dots,y_m,a_1,\dots,a_m) | y_1=p\cdot b_1+t,\ y_r=\max\{p\cdot b_r+t, y_{r-1}+t\},\ r=2,\dots,m, \\ k\cdot t\le y_k \le (T-(m-k)\cdot t), y_k - pb_k \le (k-1)\cdot t, b_k \le b_{k+1}, a_k \ge a_{k+1}, a_k, b_k \ge 0, a_k, b_k \in Z, k=1,\dots,m, \sum_{k=1}^{m}(b_k+a_k)=n\}}} f_n(b_1,\dots,b_m,y_1,\dots,y_m,a_1,\dots,a_m)$$

Recurrence Relation: $f_j(b_1,..., b_m, y_1,…, y_m, a_1,…, a_m)$ =

a) for the problem P|dep, $p_j=p|C_{\max}$;

$$\min\begin{cases} \min_{\{k|b_k>0,\ 1\le k\le m\}}\{\max\{f_{j-1}(b_1,\dots,b_{k-1},b_k-1,b_{k+1},\dots,b_m,y_1,\dots,y_m,a_1,\dots,a_m),\ p\cdot b_k\}\} \\ \min_{\{k|a_k>0,\ 1\le k\le m\}}\{\max\{f_{j-1}(b_1,\dots,b_m,y_1,\dots,y_m,a_1,\dots,a_{k-1},a_k-1,a_{k+1},\dots,a_m),\ y_k+p\cdot a_k\}\} \end{cases}$$

b) for the problem P|dep, $p_j=p|\sum_{j=1}^{n} C_j$;

$$\min\begin{cases} \min_{\{k|b_k>0,\ 1\le k\le m\}}\{f_{j-1}(b_1,\dots,b_{k-1},b_k-1,b_{k+1},\dots,b_m,y_1,\dots,y_m,a_1,\dots,a_m)+p\cdot b_k\} \\ \min_{\{k|a_k>0,\ 1\le k\le m\}}\{f_{j-1}(b_1,\dots,b_m,y_1,\dots,y_m,a_1,\dots,a_{k-1},a_k-1,a_{k+1},\dots,a_m)+y_k+p\cdot a_k\} \end{cases}$$

c) for the problem P|dep, $p_j=p|\sum_{j=1}^{n} w_j C_j$;

$$\min\begin{cases}\min_{\{k|b_k>0,\ 1\le k\le m\}}\left\{f_{j-1}(b_1,\ldots,b_{k-1},b_k-1,b_{k+1},\ldots,b_m,y_1,\ldots,y_m,a_1,\ldots,a_m)+w_j\cdot p\cdot b_k\right\}\\ \min_{\{k|a_k>0,\ 1\le k\le m\}}\left\{f_{j-1}(b_1,\ldots,b_m,y_1,\ldots,y_m,a_1,\ldots,a_{k-1},a_k-1,a_{k+1},\ldots,a_m)+w_j(y_k+p\cdot a_k)\right\}\end{cases}$$

d) for the problem P|dep, $p_j=p|L_{\max}$;

$$\min\begin{cases}\min_{\{k|b_k>0,\ 1\le k\le m\}}\left\{\max\left\{f_{j-1}(b_1,\ldots,b_{k-1},b_k-1,b_{k+1},\ldots,b_m,y_1,\ldots,y_m,a_1,\ldots,a_m),\ p\cdot b_k-d_j\right\}\right\}\\ \min_{\{k|a_k>0,\ 1\le k\le m\}}\left\{\max\left\{f_{j-1}(b_1,\ldots,b_m,y_1,\ldots,y_m,a_1,\ldots,a_{k-1},a_k-1,a_{k+1},\ldots,a_m),\ y_k+p\cdot a_k-d_j\right\}\right\}\end{cases}$$

e) for the problem P|dep, $p_j=p|\sum_{j=1}^{n} L_j$;

$$\min\begin{cases}\min_{\{k|b_k>0,\ 1\le k\le m\}}\left\{f_{j-1}(b_1,\ldots,b_{k-1},b_k-1,b_{k+1},\ldots,b_m,y_1,\ldots,y_m,z_1,\ldots,z_m)+p\cdot b_k-d_j\right\}\\ \min_{\{k|a_k>0,\ 1\le k\le m\}}\left\{f_{j-1}(b_1,\ldots,b_m,y_1,\ldots,y_m,a_1,\ldots,a_{k-1},a_k-1,a_{k+1},\ldots,a_m)+y_k+p\cdot a_k-d_j\right\}\end{cases}$$

In the equation associated with the optimal solution value, the relation y_k - $pb_k \le (k-1)\cdot t$ represents that the amount of idle time on machine k cannot be larger than the value $(k-1)\cdot t$ as in Theorem 18.

In the recurrence relation, the first term represents that job J_j is scheduled in B_k during $[p(b_k-1), pb_k]$. The second term represents that job J_j is scheduled in A_k during $[y_k+p(a_k-1), y_k+pa_k]$.

In DP Algorithm E, since $0\le b_k\le n$, $0\le a_k\le n$, and the total number of all the combinations $(y_1,\ldots, y_m)$ cannot be larger than the value $(n+1)^m$, there are a total of $O((n+1)^{3m})$ states and the function value of each state is calculated in $O(m)$ time, so that DP Algorithm E can be applied to the problems P|dep, $p_j=p|\gamma$, where $\gamma\in\{C_{max}$, $\sum_{j=1}^{n} C_j$, $\sum_{j=1}^{n} w_j C_j$, $L_{\max}$, $\sum_{j=1}^{n} L_j\}$, in the complexity order of $O(m(n+1)^{3m})$, which is a polynomial complexity order for any fixed m. However, their problem complexities remain open to prove for arbitrary value of m.

5. Identical due date case

As proved in Theorem 5 in Section 2, the problems P|ind, $d_j=d|\gamma$ and P|dep, $d_j=d|\gamma$

are NP-hard in the strong sense, where $\gamma \in \{L_{\max}, \sum_{j=1}^{n} L_j\}$. However, DP Algorithm A can be applied to the problems P|ind, d_j=d|γ in the complexity order of $O(nm(R_1+R_2)^m)$, where $\gamma \in \{L_{\max}, \sum_{j=1}^{n} L_j\}$. DP Algorithm C can be applied to the problems P|dep, d_j=d|γ in the complexity order of $O(nmR_3{}^mR_4{}^mT^m)$, where $\gamma \in \{L_{\max}, \sum_{j=1}^{n} L_j\}$.

6. Conclusion

This paper considers the problem of scheduling on parallel machines, where each machine requires maintenance activity once during a given time window. In the problem presented here, two different maintenance activities are considered, including the independent case and the dependent case. The first one allows more than one machine to be under maintenance activity simultaneously, if necessary The second one allows only one machine to be under maintenance activity at any time point. Thus, the complexities of various parallel machine problems are characterized with each of the two maintenance activities considered for various scheduling measures. Moreover, some restricted cases of the proposed problem are also characterized for their complexities, for which the associated DP algorithms are derived.

References

Wang, X. and Cheng, T.C.E., A heuristic for scheduling jobs on two identical parallel machines with a machine availability constraint, International Journal of Production Economics 161 (2015), 74-82.

Gedik, R., Rainwater, C., Nachtmann, H., and Pohl, Ed A., Analysis of a parallel machine scheduling problem with sequence dependent setup times and job availability intervals, European Journal of Operational Research 251 (2016), 640-650.

Fu, B., Huo, Y., and Zhao, H., Approximation schemes for parallel machine scheduling with availability constraints, Discrete Applied Mathematics 159 (2011) 1555-1565.

Mellouli, R., Sadfi, C., Chu, C. and Kacem, I., Identical parallel-machine scheduling under availability constraints to minimize the sum of completion times, European Journal of Operational Research 197 (2009) 1150–1165

Liao, L.W. and Sheen, G.J., Parallel machine scheduling with machine availability and eligibility constraints, European Journal of Operational Research 184 (2008) 458-467.

Kubzin, M.A., Potts, C.N., and Strusevich, V.A., Approximation results for flow shop scheduling problems with machine availability constraints, Computers & Operations Research 36 (2009) 379-390.

Costa, A., Cappadonna, F.A., and Fichera, S., Minimizing the total completion time on a parallel machine system with tool changes, Computers & Industrial Engineering 91 (2016), 290-301.

Sun, K. and Li, H., Scheduling problems with multiple maintenance activities and non-preemptive jobs on two identical parallel machines, International Journal of Production Economics 124 (2010), 151-158.

Laalaoui, Y. and M'Hallah, R., A binary multiple knapsack model for single machine scheduling with machine unavailability, Computers & Operations Research 72 (2016), 71-82.

Hfaiedh, W., Sadfi, C., Kacem, I., and Hadj-Alouane A., A branch-and-bound method for the single-machine scheduling problem under a non-availability constraint for maximum delivery time minimization, Applied Mathematics and Computation 252 (2015) 496–502.

Ángel-Bello, F., Álvarez, A., Pacheco, J., Martínez, I., A single machine scheduling problem with availability constraints and sequence-dependent setup costs, Applied

Mathematical Modelling 35 (2011), 2041-2050.

Molaee, E., Moslehi, G., and Reisi, M., Minimizing maximum earliness and number of tardy jobs in the single machine scheduling problem with availability constraint, Computers and Mathematics with Applications 62 (2011) 3622–3641.

Aggoune, R., Minimizing the makespan for the flow shop scheduling problem with availability constraints, European Journal of Operational Research 153 (2004), 534-543

Cassady, C.R. and Kutanoglu, E., Minimizing job tardiness using integrated preventive maintenance planning and production scheduling, IIE transactions 35 (2003), 503-513

Cheng, T.C.E and Wang, G., Two-machine flowshop scheduling with consecutive availability constraints, Information Processing Letters 71 (1999), 49-54

Cheng, T.C.E and Wang, G., An improved heuristic for two-machine flowshop scheduling with an availability constraint, Operations Research Letters 26 (2000), 223-229

Garey, M.R. and Johnson, D.S., Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, New York, 1979.

Graves, G.H. and Lee, C.-Y., Scheduling maintenance and semiresumable jobs on a single machine, Naval Research Logistics 46 (1999), 845-863.

Kubiak, W., Blazewicz, J., Formanowicz, P., Breit, J. and Schmidt, G., Two-machine flow shops with limited machine availability, European Journal of Operational Research, 136 (2002), 528-540

Lee, C.Y., Minimizing the makespan in the two-machine flowshop scheduling problem with an availability constraint, Operations Research Letters 20 (1997), 129-139.

Lee, C.Y., Two-machine flowshop scheduling with availability constraints, European Journal of Operational Research 114 (1999), 420-429.

Lee, C.Y. and Chen, Z.L., Scheduling jobs and maintenance activities on parallel machines. Naval Research Logistics 47 (2000), 145-165.

Pinedo, M., Scheduling: Theory, algorithms, and systems, Prentice Hall, Englewood Cliffs, NJ, 1995.

Sadfi, C., Penz, B., Rapine, C., Blazewicz, J. and Formanowicz, P., An improved approximation algorithm for the single machine total completion time scheduling problem with availabiltiy constraints, European Journal of Operational Research 161 (2005), 3-10

Objective	Additional problem characteristics	Independent case	Dependent case
C_{max}		Unary NP-hard, $O(nm(R_1+R_2)^m)$	Unary NP-hard, $O(nmR_3{}^mR_4{}^mT^m)$
	$m=q\ (\geq 2)$	Binary NP-hard, $O(nq(R_1+R_2)^q)$	Binary NP-hard, $O(nqR_3{}^qR_4{}^qT^q)$
	$p_j=p$	$O(n)$	*Unknown*, $O(m(n+1)^{3m})$
$\sum_{j=1}^{n} C_j$		Unary NP-hard, $O(nm(R_1+R_2)^m)$	Unary NP-hard, $O(nmR_3{}^mR_4{}^mT^m)$
	$m=q\ (\geq 2)$	Binary NP-hard, $O(nq(R_1+R_2)^q)$	Binary NP-hard, $O(nqR_3{}^qR_4{}^qT^q)$
	$p_j=p$	$O(n)$	*Unknown*, $O(m(n+1)^{3m})$
$\sum_{j=1}^{n} w_jC_j$		Unary NP-hard, $O(nmR_1{}^{2m}R_2{}^m)$	Unary NP-hard, $O(nmR_3{}^mR_4{}^mT^m)$
	$m=q\ (\geq 2)$	Binary NP-hard, $O(nqR_1{}^{2q}R_2{}^q)$	Binary NP-hard, $O(nqR_3{}^qR_4{}^qT^q)$
	$p_j=p$	$O(n\log n)$	*Unknown*, $O(m(n+1)^{3m})$
L_{max}		Unary NP-hard, $O(nm(R_1+R_2)^m)$	Unary NP-hard, $O(nmR_3{}^mR_4{}^mT^m)$
	$m=q\ (\geq 2)$	Binary NP-hard, $O(nq(R_1+R_2)^q)$	Binary NP-hard, $O(nqR_3{}^qR_4{}^qT^q)$
	$p_j=p$	$O(n\log n)$	*Unknown*, $O(m(n+1)^{3m})$
	$d_j=d$	Unary NP-hard, $O(nm(R_1+R_2)^m)$	Unary NP-hard, $O(nmR_3{}^mR_4{}^mT^m)$
$\sum_{j=1}^{n} L_j$		Unary NP-hard, $O(nm(R_1+R_2)^m)$	Unary NP-hard, $O(nmR_3{}^mR_4{}^mT^m)$
	$m=q\ (\geq 2)$	Binary NP-hard, $O(nq(R_1+R_2)^q)$	Binary NP-hard, $O(nqR_3{}^qR_4{}^qT^q)$
	$p_j=p$	$O(n\log n)$	*Unknown*, $O(m(n+1)^{3m})$
	$d_j=d$	Unary NP-hard, $O(nm(R_1+R_2)^m)$	Unary NP-hard, $O(nmR_3{}^mR_4{}^mT^m)$

Table 1. Complexities of the scheduling problems.

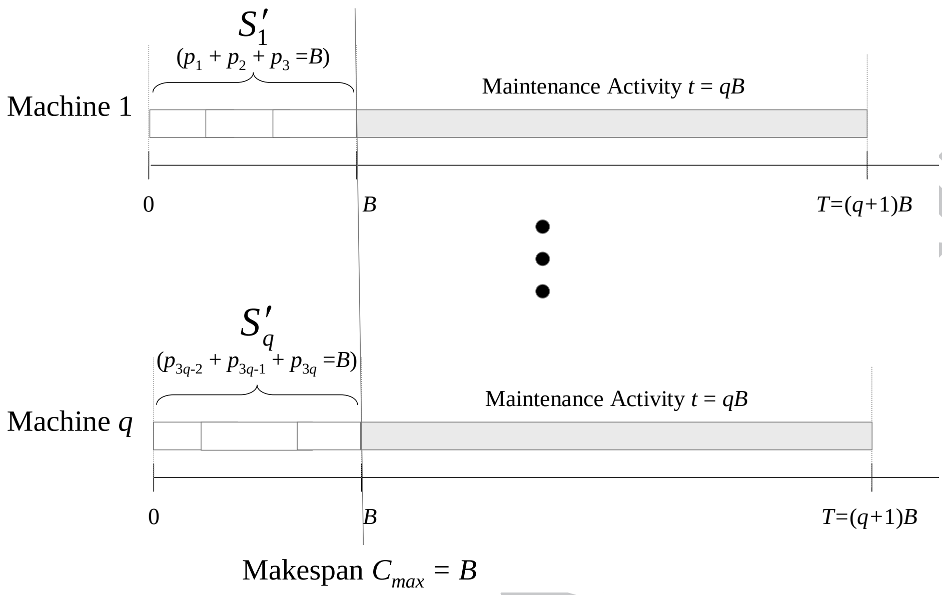


Figure 1. The structure of the schedule π.

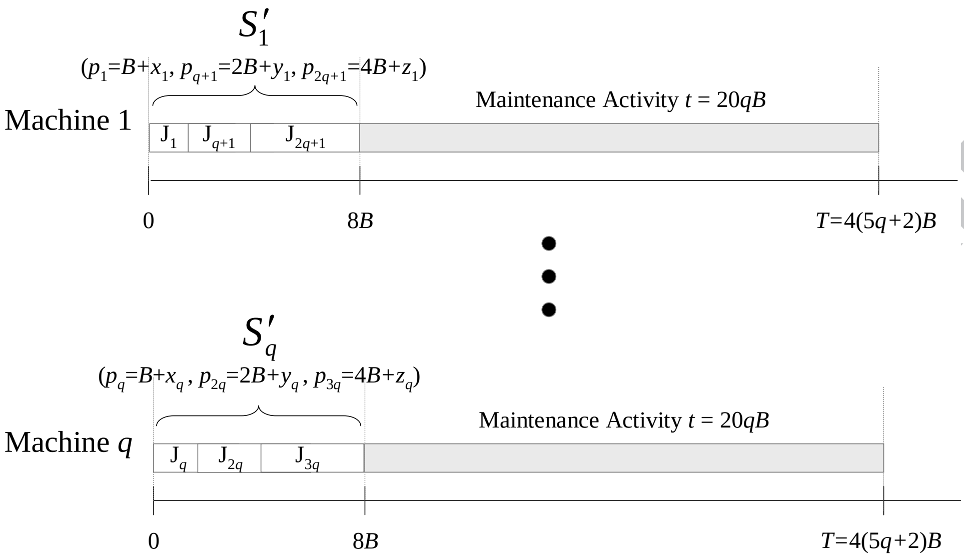


Figure 2. The structure of the schedule π .

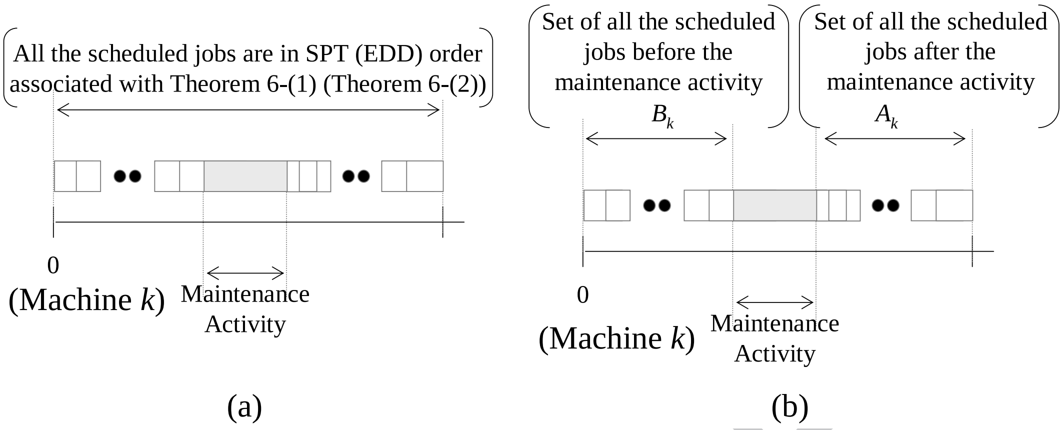


Figure 3. Illustrative examples associated with Theorem 6.

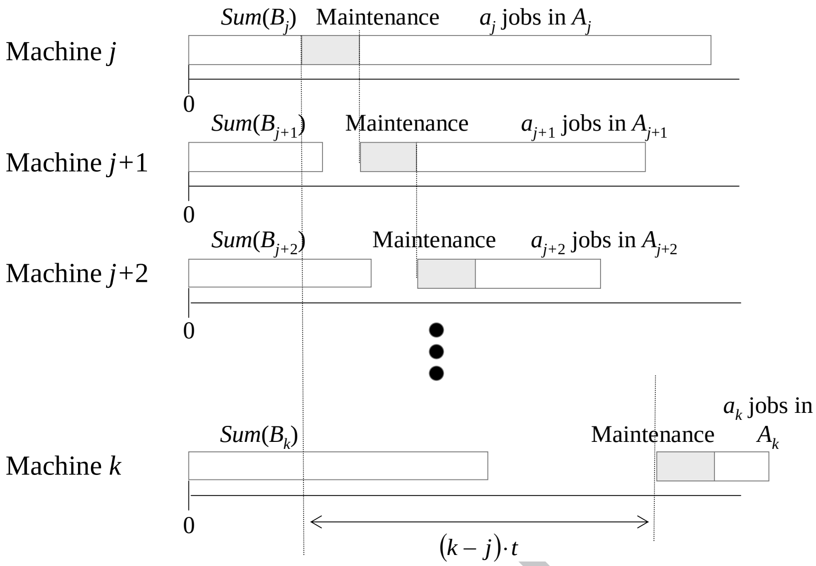


Figure 4. Illustrative example associated with Theorem 9.

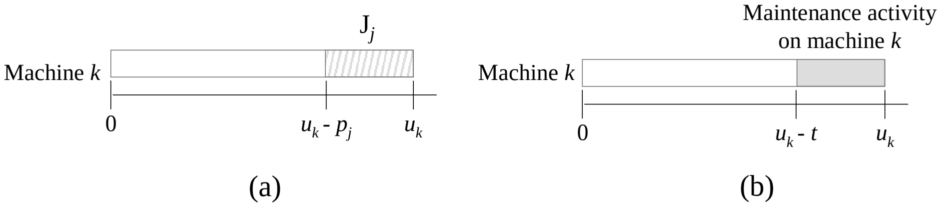


Figure 5. The possible cases in DP Algorithm A.

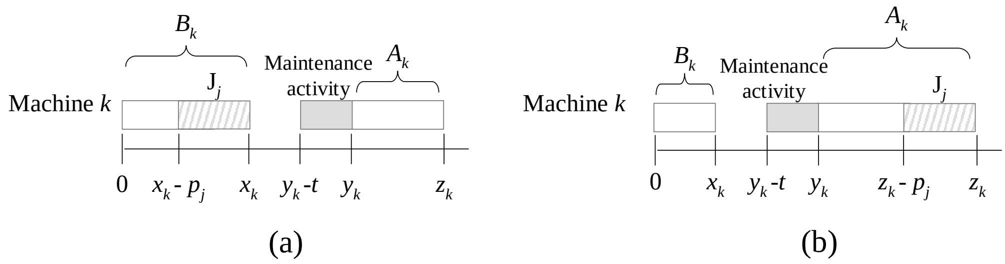


Figure 6. The possible cases in DP Algorithms B and C.

Highlights of this paper

- Parallel machine scheduling with maintenance activity.
- makespan, sum of completion times, maximum lateness and sum of lateness.
- For each scheduling measure, the problem is proved to be strongly NP-hard.
- Some restricted cases are also characterized for their complexities.