# An effective Hybrid Petri net and Tabu Search for Parallel Machines Scheduling with Availability Constraint

**Husam Kaid, Abdulrahman Al-Ahmari, Emad Abouel Nasr, Adel Al-Shayea, Ali K. Kamrani, Mohammed A. El-Meligy, and Haitham A. Mahmoud**
Industrial Engineering Department, College of Engineering, King Saud University, Riyadh 11421, Saudi Arabia
Mechanical Engineering Department, Faculty of Engineering, Helwan University, Cairo 11732, Egypt
Industrial Engineering Department, College of Engineering, University of Houston, Houston, TX 77204 4008, USA
Advanced Manufacturing institute, King Saud University, PO Box 800, Riyadh 11421, Saudi Arabia
yemenhussam@yahoo.com, alahmari@ksu.edu.sa, eabdelghany@ksu.edu.sa, alshayea@ksu.edu.sa, akamrani@uh.edu, melmeligy@ksu.edu.sa, hmahmoud@ksu.edu.sa

## Abstract

Machines can become unavailable during the planning horizon due to unexpected breakdowns. Such periods of unavailability should be properly taken into account when designing the production schedule. This paper proposes an optimal solution to unrelated parallel machine scheduling problems with availability constraints based on timed Petri nets (TPNs) to minimize the maximum completion time of jobs. For this NP-hard problem, a new hybrid Petri net and tabu search (PNTS) approach is proposed to find an optimal solution for this problem. A numerical example is used to assess the performance of the proposed PNTS. The computational results highlight the ability of the proposed PNTS to obtain optimal solutions for this problem.

## Keywords
Scheduling, parallel machines, Petri net and tabu search.

## 1. Introduction

Scheduling has a paramount role in production management. It dictates what will be done and chooses what resources that can be used (Cardon et al. 2000). The production scheduling can be challenging issue at the operational management level to get a schedule that maximizes the criterion. Many scheduling issues exist in modern manufacturing environments. There are several different types of shops in the factory, depending on job flow and the machine design (Moon et al. 2002). Machines, which processes orders on a job sequencing basis, can be categorized as a "flow shop". In a job shop, jobs can be done randomly on the machines in any order and there are no restrictions on machine order enforced. Machine scheduling is one of the most widely researched fields in the literature on operations, and it covers a wide range of issues (Muter 2020). The issue discussed in this paper is that independent jobs should be scheduled on unrelated parallel machines under availability constraint to minimize the maximum completion time (makespan) of jobs. There are *n* separate jobs, each with its own processing time, and can be processed on any *m* parallel machine that is non-identical. There are (n!)m possible schedules choices (Moon et al. 2002). Therefore, the scheduling problem is more difficult to be solved via dispatching rules (shortest processing time (SPT), longest processing time (LPT), ...etc), and these are called non-deterministic polynomial-time (NP)-hard problems (Sethi 1977, Lenstra and Rinnooy Kan 1978). Several studies are proposed to solve unrelated parallel machine scheduling problems.. (Joo et al. 2012) proposed two metaheuristics, these are a GA and a new evolutionary meta-heuristic population-based algorithm—self-evolution—to assess job allocation and unrelated parallel-machine scheduling for makespan minimization. The performance of the meta-heuristic algorithms are assessed using generated examples and the results are compared with optimal solutions. (Fleszar et al. 2012) hybridized mathematical programming elements with a variable neighborhood descent search approach for unrelated parallel machines scheduling problem for makespan minimization. (Joo et al. 2015) hybridized dispatching rules with a GA for unrelated

parallel machines scheduling problem under machine-dependent processing times and sequence and machine-dependent setup times for makespan minimization. (Geyik and Elibal 2017) proposed a fuzzy linguistic method for examining the learning effect on the job scheduling in unrelated parallel-machine with uncertain processing times. (Arroyo et al. 2017) developed a mixed-integer programming model, lower bound, and heuristics based on first- and best-fit earliest job-ready time rules for scheduling non-identical job sizes and unequal ready times in unrelated parallel-batch processing machines for makespan minimization. (Tan et al. 2019) integrated a greedy strategy and GA for green non-identical job sizes scheduling in unrelated parallel machines to minimize total electricity costs of production. Petri nets (PNs) are interesting and useful graphical and mathematical tool to assist in modeling, analysis, control, and scheduling of manufacturing systems (Kaid et al. 2020a, Kaid et al. 2020c, Kaid et al. 2020b). The study of (van der Aalst 1996) presented an approach based on timed Petri net to model scheduling problems. The way for conversion scheduling problems to timed Petri nets is developed. The approach demonstrates that the formalism of the Petri net can be used for modeling jobs, machines, and constraints. By converting a scheduling problem to timed Petri nets we can examine the scheduling problem using Petri net theory. This approach can therefore use Petri net-based analytical methods to find contradictory precedents, define lower and upper limits for minimal makespan, etc. A scheduling problem of large sized flexible manufacturing system (FMS) under availability constraints is considered in the work of (Kammoun et al. 2017). A mathematical model based on timed Petri nets is proposed for optimizing solutions. Based on the decomposed TPN property, the mathematical model solves the scheduling problem. Furthermore, a GA is proposed to find efficient solutions for big scheduling problems. (Rezig et al. 2020) develops a scheduling model with production and maintenance constraints for a discrete event system based on Petri nets. The study demonstrated the performance of the mathematical model, an implementation example of the approach such as hospital bed management. It is known that parallel machines scheduling problems with continuous availability of machines are NP-hard. With availability constraints, unrelated parallel machine scheduling problems are NP-hard for the maximum completion time (makespan) objective (Lee 1996, Sevindik 2006, Al-Harkan and Qamhan 2019). Although scheduling problems under machine availability constraints have become very popular for the last decade, there is still very limited literature on this problem. Therefore, this paper aims to propose a new hybrid Petri net and tabu search approach is proposed to optimize the unrelated parallel machines scheduling problem under availability constraint to minimize makespan. The rest of the paper is organized as follows. Section 2 presents basics of timed Petri nets, the synthesis of unrelated machines based on Petri nets, and scheduling problem. Sections 3 displays the proposed approach. Numerical example is presented in Section 4. Finally, Section 5 presents conclusions and future work.

## 2. Preliminaries

### 2.1 Basics of Timed Petri Nets

***Definition 1:*** Let $N = (P, T, F, W, K, D, M_o)$ be a finite-capacity timed Petri net, where

1. $P = \{p^0\} \cup P_A \cup P_R$, where $p^0$ is a process idle place, $P_A$ is a finite non-empty set of operation places, $\theta > 0$, and $P_R$ is a finite non-empty set of resource places in the system;
2. $T = \{t_1, t_2, \ldots, t\sigma\}$, $\sigma > 0$, is a finite set of transitions with $P \cap T = \emptyset$ and $P \cup T \neq \emptyset$;
3. $F \subseteq (P \times T) \cup (T \times P)$ is said to be an input and output function of $N$ that connecting places and transitions;
4. $W$: $(P \times T) \cup (T \times P) \rightarrow \mathbf{IN}$ is a mapping function that adds weight for all arcs, where $\mathbf{IN} = \{0, 1, 2, ...\}$ is the set of nonnegative integers;
5. $K$: $P \rightarrow \mathbf{IN}$ is a capacity function that adds the maximal number of tokens to each place $K(p)$;
6. $D$: $T \rightarrow \mathbf{TS}$ is a firing delay function that adds to each transition $t$ the firing delay $D(t)$, where **TS** is the time set, $\mathbf{TS} > 0$;
7. $M_o$: $P \rightarrow \mathbf{IN}$ is an initial marking function that assigns the number of tokens to each place $p_i \in P$. $M_o = (M_o(p_1), M_o(p_2), \ldots, M_o(p_\theta))^T$, $\theta > 0$. $M(p_i)$ is the number of tokens in $p$ at a marking $M$.

***Definition 2:*** Let $(N, M_o)$ be a timed Petri net with $N = (P, T, F, W, K, D, M_o)$ and nodes $p,t \in P \cup T$ are a place and a transition in $N$, respectively. The preset of $p$ ($t$) is the set of all input transitions (places) of $p$ ($t$), i.e., ${}^\bullet p = \{t \in T \mid (t, p) \in F\}$ (${}^\bullet t = \{p \in P \mid (p, t) \in F\}$). The postset of $p$ ($t$) is the set of all output transitions (places) of $p$ ($t$), i.e., $p^\bullet = \{t \in T \mid (p, t) \in F\}$ ($t^\bullet = \{p \in P \mid (t, p) \in F\}$).

***Definition 3:*** Let $(N, M_o)$ be a timed Petri net with $N = (P, T, F, W, K, D, M_o)$. $N$ is called an ordinary net if $p \in P$, $t \in T$, $\forall (p, t) \in F$, and $W(p, t) = 1$, $i = 1, 2, \ldots, \theta$, and $j = 1, 2, \ldots, \sigma$.

***Definition 4:*** Let $(N, M_o)$ be a timed Petri net with $N = (P, T, F, W, K, D, M_o)$. $N$ is called a weighted net if $\exists p \in P$, $\exists t \in T$, $(p, t) \in F$, and $W(p, t) > 1$.

***Definition 5:*** Let $(N, M_o)$ be a timed Petri net with $N = (P, T, F, W, K, D, M_o)$. $N$ is called acceptably marked if (1) $M_o(p^0) \geq 1$, (2 $M_o(p) = 0$, $\forall p \in P_A$, and (3) $M_o(p) = 0$, $\forall p \in P_R$.

***Definition 6:*** Let $(N, M_o)$ be a timed Petri net with $N = (P, T, F, W, K, D, M_o)$. $N$ called a self-loop if for all $p, t \in P \cup T$; $W(p, t) > 0$ implies that $W(t, p) > 0$.

***Definition 7:*** Let $(N, M_o)$ be a timed Petri net with $N = (P, T, F, W, K, D, M_o)$. $N$ is called self-loop free if for all $p, t \in P \cup T$; $W(p, t) > 0$ implies that $W(t, p) = 0$.

***Definition 8:*** Let $(N, M_o)$ be a timed Petri net with $N = (P, T, F, W, K, D, M_o)$. A transition $t \in T$ is called enabled at marking $M$ if

$$M(p) \geq W(p, t), \forall p \in P, \text{ and } \forall p \in {}^\bullet t \quad (1)$$

and

$$K(p) \geq M(p) - W(p, t) + W(t, p), \forall p \in P, \text{ and } \forall p \in t^\bullet \quad (2)$$

***Definition 9:*** Let $(N, M_o)$ be a timed Petri net with $N = (P, T, F, W, K, D, M_o)$. The marking $M'$ resulting from the firing of an enabled transition $t_j$ at marking $M$ is defined by $M[t\rangle M'$ and can be updated from $M$ to $M'$ as follows:

$$M'(p_i) = \begin{bmatrix} M(p_i)+W(p_i, t_j) & \text{if } p_i \in {}^\bullet t_j \setminus t_j^\bullet \\ M(p_i)-W(t_j, p_i) & \text{if } p_i \in t_j^\bullet \setminus {}^\bullet t_j \\ M(p_i)+W(t_j, p_i) - W(p_i, t_j) & \text{if } p_i \in t_j^\bullet \cap {}^\bullet t_j \\ M(p_i) & \text{otherwise} \end{bmatrix} \quad (3)$$

***Definition 10:*** Let $(N, M_o)$ be a timed Petri net with $N = (P, T, F, W, K, D, M_o)$. The set of reachable markings from $M$ is denoted by $R(N, M)$, which is expressed by vertices and arcs; vertices represent markings that are marked with $M_k$ and arcs represent firing of transitions that are marked with $t_j$. If $t_j$ fires, then there is an arc from marking $M_k$ to marking $M_l$ and $M_l$ is reached.

***Definition 11:*** Let $(N, M_o)$ be a timed Petri net with $N = (P, T, F, W, K, D, M_o)$. $[N]$ is called the incidence matrix of net $N$, where $[N]$ defined as $|P| \times |T|$ integer matrix with $[N](p, t) = W(t, p) - W(p, t)$. $[N](p,.)$ denotes the row of $[N]$ that related to the place $p$ and $[N](., t)$ denotes the column of $[N]$ that related to the transition $t$.

***Definition 12:*** Let $(N, M_o)$ be a timed Petri net with $N = (P, T, F, W, K, D, M_o)$. At a marking $M$, if there exists a firing sequence of transitions $\delta = t_1\ t_2\ t_3\ \ldots\ t\sigma$ that generates a marking $M'$, then a marking $M'$ is called reachable from $M$, denoted as $M[\delta\rangle M'$ and satisfying the state equation $M' = M + [N]\ \vec{\delta}$, where $\vec{\delta}$: $T \rightarrow \mathbf{IN}$ is called a firing count vector that maps $t$ in $T$ to the number of occurrences of $t$ in $\delta$.

***Definition 13:*** Let $(N, M_o)$ be a timed Petri net with $N = (P, T, F, W, K, D, M_o)$. A place $p$ in $N$ is called $q$ –bounded if there exists $q \in \mathbf{IN}$, $\forall M \in R(N,M_0)$, $\forall p_i \in P$, $M(p_i) \leq q$. $(N, M_o)$ is called $q$ –bounded if $\forall p_i \in P$ is $q$ –bounded.

***Definition 14:*** Let $(N, M_o)$ be a timed Petri net with $N = (P, T, F, W, K, D, M_o)$. $N$ is called safe net system if $\forall M \in R(N,M_0)$, $\forall p_i \in P$, $M(p_i) \leq 1$.

***Definition 15:*** Let $(N, M_o)$ be a timed Petri net with $N = (P, T, F, W, K, D, M_o)$. A marking $M_o$ is called reversible if for each marking $M' \in R(N, M_o)$, $M_o$ is reachable from $M'$.

Consider the example of two unrelated parallel machines presented in Figure 1. The system has two unrelated machines M1 and M2. One job can be processed at a time by each machine. In addition, three jobs are considered to be processed in the system. Figure 2 presents the Petri net model of the two unrelated parallel machines example. It has five places and four transitions. The following sets of places can be used: $P^0 = \{p_1, p_6\}$, $P_R = \{p_4, p_5\}$, and $P_A = \{p_2, p_3\}$. The initial marking is $M_o = (3, 0, 0, 1, 1)^T$. The firing delay of jobs (job type (M1, M2) unit time) are job 1 (5, 8), job 2 (9, 8), and job 3 (5, 7). The tokens in place $p_1$ denote three jobs that need to be produced by one of the two machines. At times 0, all jobs are available and the two machines are free and able to process any job. Let us assume that the machine 1 processes a job 1, i.e. transition $t_1$ (M1 start) fires at time 0 and takes a job 1 from $p_1$ and transfers it into $p_2$ (M1 busy). Thus, place $p_4$ (M1 free) is empty and $p_2$ (M1 busy) has a token. The second machine processes job 2, i.e. transition $t_2$ (M2 start) fires at time 0 and takes a job 2 from $p_1$ and transfers it into $p_3$ (M2 busy). As a result, place $p_5$ (M2 free) is empty and $p_3$ (M2 busy) has a token. In this case, transitions $t_1$ (M1 start) and $t_2$ (M2 start) are not enabled since both machines are busy. At time 5 transition $t_3$ (M1 finish) fires, followed by the firing of transition $t_1$ (M1 start) at time 5. At time 8 $t_4$ (M2 finish) fires, followed by the firing of transition $t_3$ (M1 finish) at time 10. Therefore, the completion time of these jobs are 5, 8 and 10.

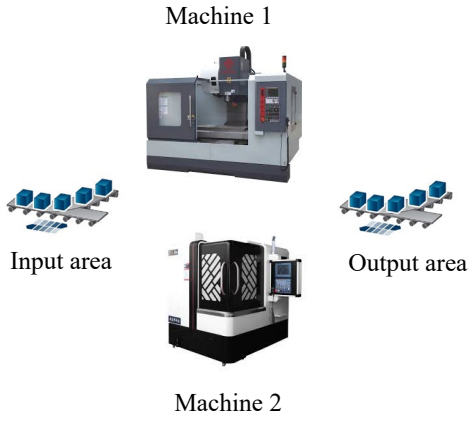


Figure 1. Example of two unrelated parallel machines.

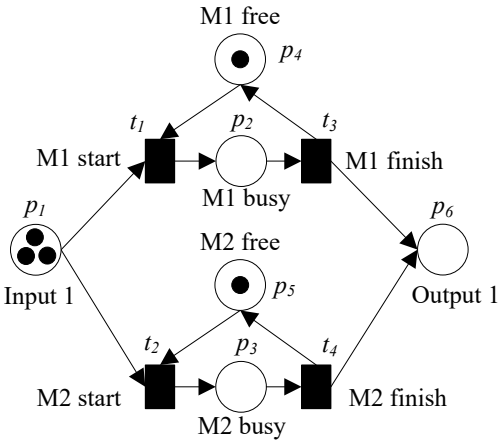


Figure 2. Petri net model of a system presented in Figure 1.

## 2.2 Unrelated Machines Based on Petri Nets

This section presents the modeling approach for recovery subnets to model all the preventive maintenance (PM) actions in unrelated machines.

***Definition 16:*** Let $(N, M_o)$ be a timed Petri net with $N = (P, T, F, W, K, D, M_o)$. Let $r_u \in P_R$ be an unreliable resource (unreliable unrelated machine). A recovery preventive maintenance subnet of $r_u$ is a Petri net with $N_{PMi} = (\{p_i, p_{PMi}\}, \{ t_{Bi}, t_{Ei} \}, F_{PMi})$, where $F_{PMi} = \{( p_i, t_{Bi}), (t_{Bi}, p_{PMi}), (p_{PMi}, t_{Ei}), (t_{Ei}, p_i )\}$, and a preventive maintenance action can be done on an unrelated machine $r_u$ if it is in a busy state (its holders), $p_i \in H(r_u)$, where $H(r_u)$ is a non-empty set of holders of $r_u$, expressed by $H(r_u) = \{p | p \in P_A, p \in {}^{\bullet\bullet}r_u \cap P_A \neq \emptyset\}$. $(N_{PMi}, M_{PMio})$ is called a marked recovery preventive maintenance subnet, where $M_{PMio}(p_i) \geq 0$ and $M_{PMio}(p_{PMi}) = 0$.

In Definition 16, $p_{PMi}$ represents the recovery preventive maintenance place of $p_i$, $t_{Bi}$ represents the beginning time of PM period on machine and $t_{Ei}$ represents the ending time of PM period on machine.

***Definition 17:*** Let $(N, M_o)$ be a timed Petri net with $N = (P, T, F, W, K, D, M_o)$. For all $r_u \in P_R$, adding a preventive maintenance subnet for each $p_i \in H(r_u)$ results in an unreliable timed Petri net, expressed by $(N_U, M_{Uo}) = (N, M_o) \parallel (N_{PMi}, M_{PMio})$ that is the composition of $(N, M_o)$ and $(N_{PMi}, M_{PMio})$.

***Definition 18:*** Let $(N_U, M_{Uo})$ be an unreliable timed Petri net with $N_U = (P_U, T_U, F_U, W_U, K_U, D_U, M_{Uo})$, and $R(N_U, M_{Uo})$ be its reachable graph, where $P_U = P \cup P_{PM}$, $T_U = T \cup T_B \cup T_E$, and $F_U = F \cup F_{PM}$. $P_{PM}$, $T_B$, and $T_E$ represent the sets of the recovery preventive maintenance places, time to PM action transitions, and time of performing PM

transitions, respectively. Here, $P_{PM} = \cup_{i\in NA}\{p_{PMi}\}$, $T_B = \cup_{i\in NA}\{t_{Bi}\}$, $T_E = \cup_{i\in NA}\{t_{Ei}\}$, $\mathbf{NA} = \{i|p_i \in H(r_u)\}$, $F_{PM} \subseteq (T_E \times P_{PM}) \cup (P_{PM} \times T_B)$, $F_U \subseteq (P_U \times T_U) \cup (T_U \times P_U)$, $W_U \subseteq (P_U \times T_U) \cup (T_U \times P_U) \rightarrow \mathbf{IN}$, $K_U \subseteq P_U \rightarrow \mathbf{IN}$, $D_U \subseteq T_U \rightarrow \mathbf{TS}$, and $M_{Uo}$ is an initial marking of $N_U$.

To illustrate the recovery preventive maintenance subnet of unreliable unrelated machines, consider a timed Petri net model presented in Figure 2, we have two unreliable unrelated machines which are $p_4$ and $p_5$, $H(p_4) = \{p_2\}$, and $H(p_5) = \{p_3\}$. Adding recovery preventive maintenance subnets for $p_4$ and $p_5$ by Definition 16 results in an unreliable timed Petri net, as shown in Figure 3. $\mathbf{NA} = \{2, 3\}$, $P_{PM} = \{p_{PM2}, p_{PM3}\}$, $T_B = \{t_{B1}, t_{B2}\}$, and $T_E = \{t_{E1}, t_{E2}\}$. When an unreliable resource $p_4$ needs preventive maintenance in $p_2$, then the token in $p_2$ moves into $p_{PM2}$ by firing $t_{B1}$ at time of PM action on machine 1 . If $t_{B1}$ fires, it takes a token from $p_2$ and deposits it into place $p_{PM2}$. If the mean time of performing a PM on machine 1 is elapsed, the token in $p_{PM2}$ moves into $p_2$ by firing $t_{E1}$. If transition $t_{E1}$ fires, it takes one token from $p_{PM2}$ and deposits them into $p_2$, denoting that a resource recovery preventive maintenance is finished. In addition, if an unreliable resource $p_5$ needs preventive maintenance in $p_3$, then the token in $p_3$ moves into $p_{PM3}$ by firing $t_{B3}$ at time of PM action on machine 2. If $t_{B2}$ fires, it takes a token from $p_3$ and deposits it into place $p_{PM3}$. If the mean time of performing a PM on machine 2 is elapsed, the token in $p_{PM3}$ moves into $p_3$ by firing $t_{E2}$. If transition $t_{E2}$ fires, it takes one token from $p_{PM3}$ and deposits them into $p_3$, denoting that a resource recovery preventive maintenance is finished.

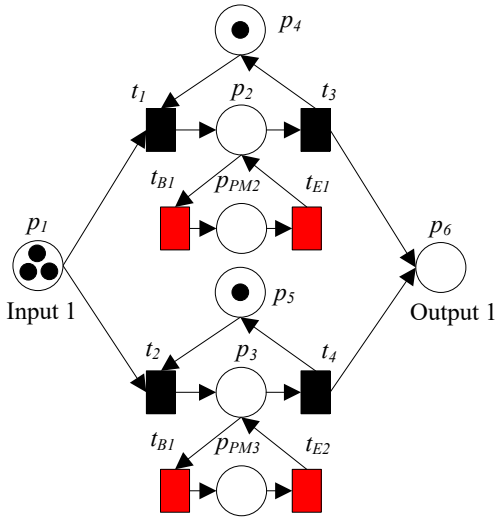


Figure 3. Recovery preventive maintenance model of unreliable unrelated parallel machines.

### 2.3 Scheduling Problem Based on Petri Nets

Scheduling involves allocating resources to tasks over time. Some researchers use 'machines' or 'processors' to refer to resources and tasks are also named ‘steps of a job’ or ‘operations’ in a standard production environment. Unreliable unrelated parallel machines scheduling is defined as assigning jobs to machines. The execution of a job needs one machine, i.e. a job is processed by a machine. In addition, a number of machines may be able to process a certain job. The aim of the scheduling is to find the sequence and processing times for machine jobs to optimize a performance criterion.

***Definition 19:*** Let SP = ($J$, $\mathcal{M}$, $PT$, $GN$, $AS$) be a scheduling problem of unreliable unrelated parallel machines, where

1. $J = \{J_1, J_2, \ldots, J_n\}$, $n > 0$, is a finite non-empty set of jobs;
2. $\mathcal{M} = \{m_1, m_2, \ldots, m_m\}$, $m > 0$, is a finite non-empty set of unrelated machines;
3. $PT$: $(J \times \mathcal{M}) \rightarrow \mathbf{TS}$ is a function that defines for each job:
   a. The unrelated machines sets capable of processing job $J_j$, denoted as $\{m_i \in \mathcal{M}, J_j \in J \mid (Jj, mi) \in PT\}$, $j = 1, 2, \ldots, n$ and $i = 1, 2, \ldots, m$;
   b. The required processing time to process job $j$ by a certain unrelated machine $i$, denoted as $PT(Jj, mi)$.
4. $GN$ denotes the Graham scheduling notation and represented by $\alpha \mid \beta \mid \gamma$, where $\alpha$ is the environment of the machine, $\beta$ is job characteristics and constraints, and $\gamma$ is the objective function;
5. $AS$ is a set of assumptions about the structure of an unrelated machines scheduling problem.

***Definition 20:*** Let SP = ($J$, $\mathcal{M}$, $PT$, $GN$, $AS$, $S$) be a scheduling problem of unrelated machines, where $S = \{s_1, s_2, \ldots, s_k\}$ is a set of schedules in the scheduling problem and $s_k$ can be represented by a function $S: J \rightarrow (\mathcal{M} \times \mathbf{TS})$. Let $J_j$ is a job and $(m_i, st_j) \in S$, then $J_j$ is processed by machine $m_i$ starting at time $st_j$. Let $c_j$ is the completion time of job $J_j$ and $m_{i,j}$ is the machine that is used to process job $J_j$.

***Definition 21:*** Let SP = ($J$, $\mathcal{M}$, $PT$, $GN$, $AS$, $S$) be a scheduling problem of unreliable unrelated machines. A schedule $s_j \in S$ is feasible if

1. Each machine cannot be used simultaneously to process multiple jobs: $\forall j,k \in J, (m_{i,j} \cap m_{i,k} \neq \emptyset) \Rightarrow (c_j \leq st_k \vee c_k \leq st_j)$;
2. At most one machine processes each job;
3. Each job is processed in time interval $(0, +\infty)$;
4. If precedence constraints exist for some jobs, , then the precedence are obeyed, i.e., job $j$ has to be processed before job $k$: $c_j \leq st_k$;
5. All jobs are finished;
6. If jobs are non-preemptable, then no job is preempted.
7. If the machine constraints exist, they have to be satisfied;

***Definition 22:*** Let SP = ($J$, $\mathcal{M}$, $PT$, $GN$, $AS$, $S$) be a scheduling problem of unreliable unrelated machines. In any schedule $s_j \in S$, let $c_1, c_2, \ldots, c_n$ be the completion times of jobs. A schedule is called optimal if the optimality criterion value for the maximum completion time ($C_{max}$) is optimal and defined as follows:

$$C_{max} \coloneqq \max_{1 \leq j \geq n} \{c_j\} \quad (4)$$

To convert unreliable unrelated machines scheduling problem onto timed Petri nets, we have to convert terms such as jobs and machines onto places and transitions. Figure 4 presents how to model jobs in terms of an unreliable unrelated machines model.

Assume that a set of n jobs J1, $J_2, \ldots, J_n$ has to be processed by a set of $m$ unrelated machines $m_1, m_2, \ldots, m_m$. Each jobs has three stages: (1) Job $j$ is waiting to be processed, denoted by $sp_j$, (2) Job $j$ is being processed, denoted as $p_{(j,i)}$ and (3) Job $j$ has been processed, denoted by common place $cp$. Thus, two significant milestones are identified: start and job j completion time, which are transitions $st_{(j,i)}$ and $c_{(j,i)}$, respectively. Moreover, each unrelated machine, which can be used to process a job $j$ is modelled by a place $m_i$. Initially, $m_i$ includes one token. Transition $st_{(j,i)}$ needs the unrelated machine if the execution of job $j$ begins, transition $c_{(j,i)}$ releases the unrelated machine if job $j$ finishes.

***Definition 23:*** Let SP = ($J$, $\mathcal{M}$, $PT$, $GN$, $AS$, $S$) be a scheduling problem of unreliable unrelated parallel machines. Let $N_S = (P_S, T_S, F_S, W_S, K_S, D_S, M_{So})$ be the corresponding timed Petri net of the SP if

1. $P_S = \{p_{(j,i)} | (j,i) \in PT\} \cup \{cp_j | j \in J\} \cup \{m_i | i \in \mathcal{M}\} \cup P_{PM} \cup \{sp\}$;
2. $T_S = \{st_{(j,i)} | (j,i) \in PT\} \cup \{c_{(j,i)} | (j,i) \in PT\} \cup T_E \cup T_B$;
3. $F_S \subseteq (P_S \times T_S) \cup (T_S \times P_S)$;
4. $W_S \subseteq (P_S \times T_S) \cup (T_S \times P_S) \rightarrow \mathbf{IN}$;
5. $K_S \subseteq (P_S \times T_S) \cup (T_S \times P_S) \rightarrow \mathbf{IN}$;
6. $D_S(st_{(j,i)}) = PT_{(j,i)}$ and $D_S(c_{(j,i)}) = 0$;
7. $M_{So}$ is an initial marking of $N_S$.

## 3. Tabu Search Algorithm

Tabu search is a popular metaheuristic search approach using local search techniques for optimization problems (Glover 1989). Tabu search is carried out using a neighborhood or local search mechanism to iteratively move from one feasible solution A to an improved solution B in the neighborhood of A until a stop condition is satisfied. In order to prevent solutions previously visited, tabu search uses memory named tabu list to store results about the search process. The tabu list is updated in each TS iteration. The risk of rejecting solutions, which have not yet been created may occur due to the restrictions in the tabu list. Tabu results are then tested for such parameters, which are known as aspiration criteria. If the tabu value is better than the best objective value found earlier, the tabu will accept the solution and delete its move from the tabu list.

TS has many factors or parameters, which must be tuned before its searching starts. Initial solution, the next element, the tabu list, the aspiration condition, and the stopping condition are main stopping of the TS. In this study, the initial solution for TS has been set by the dispatching rule longest processing time (LPT) policy, which sequences tasks or jobs in a descending order of processing time. In addition, to generate new candidate solutions for TS algorithm, we used the pairwise interchange operator. According to preliminary experiments, the number of candidates generated was 50. The size of the tabu list is set to ten elements and the first-in-first-out (FIFO) technique is used to update the tabu list based

on the same preliminary experiments. As previously reported, the aspiration criteria were applied (when a move leads to a better solution than the previously found best solution value). Finally, the stopping criteria is then defined such that computational times (in seconds) are similar in each algorithm or if the lower bound is reached. The general TS algorithm can be constructed as follows if a minimization problem occurs (Kaid et al. 2015, Alharkan et al. 2020):

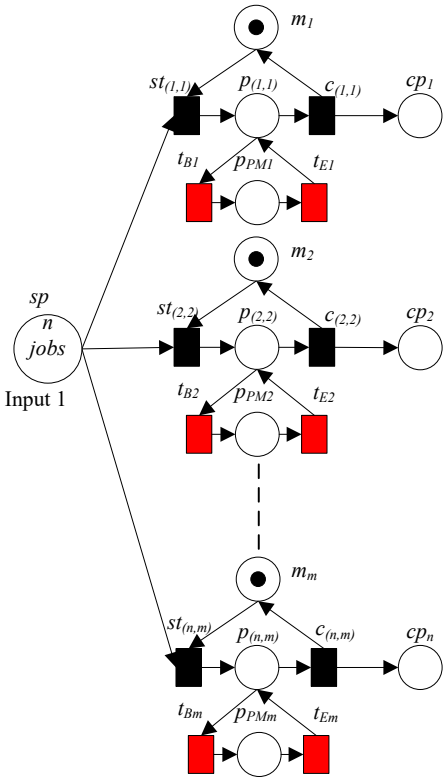


Figure 4. Unreliable unrelated parallel machines scheduling model based on timed Petri nets.

**Algorithm 1: Construction of PNTS algorithm.**

**Input:** unreliable unrelated parallel machines problem SP = ($J$, $\mathcal{M}$, $PT$, $GN$, $AS$, $S$) and corresponding timed Petri net of the SP $N_S$ = ($P_S$, $T_S$, $F_S$, $W_S$, $K_S$, $D_S$, $M_{So}$)

**Initialization:** Let $s$ to be the initial solution obtained by timed Petri net, let the best solution is to be $s_0$, $s_0 = s$ set the tabu list, medium-term and long-term memories.

**Step 1**: **while** No. of iterations < maximum iterations **do**

1. Create a set "$K$" of solutions;
2. Compute $K$ solutions by timed Petri net;
3. Find best neighbor $s'$ of $K$;
4. $s = s'$;
5. Update aspiration conditions and tabu list;
6. **If** $f(s) < f(s_0)$, **Then** $s_0 = s$;

**end while**

**Output:** The best solution $s_0$.

## 4. Numerical Example

Consider an unrelated parallel machine scheduling problem shown in Figure 5. It consists of three machines $i$ = 1, 2, 3 operating nine original jobs $j$ = 1, 2, …, 9. The job processing times on machines $p_{ji}$ and the periods of preventive maintenance ([$t_{Bi}$, $t_{Ei}$]) are presented in Tables 1 and 2, respectively.

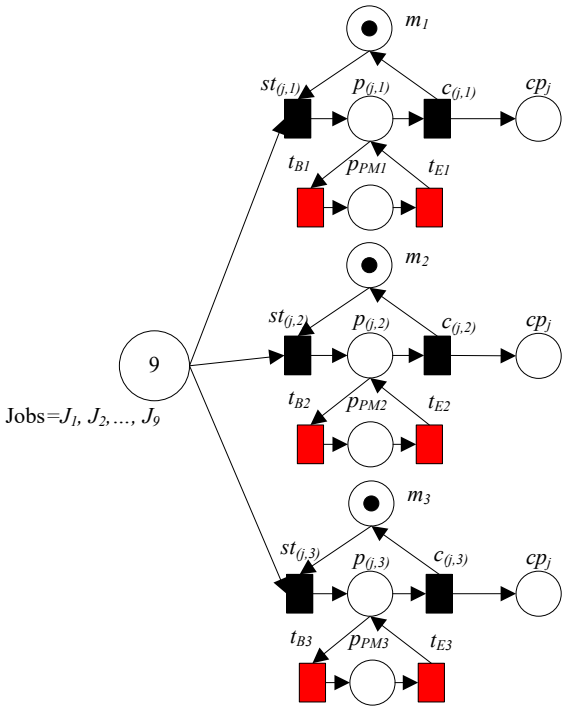


Figure 5. Unrelated parallel machines scheduling model based on timed Petri nets for numerical example.

Table 1. The job processing times (hr).

| Job | Machine | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| 1 | 18 | 9 | 4 |
| 2 | 14 | 7 | 3 |
| 3 | 24 | 12 | 6 |
| 4 | 31 | 15 | 7 |
| 5 | 16 | 8 | 4 |
| 6 | 20 | 10 | 5 |
| 7 | 22 | 11 | 6 |
| 8 | 26 | 13 | 6 |
| 9 | 14 | 7 | 4 |

Table 2. The preventive maintenance periods.

| Machine | $t_{Bi}$ (hr) | $t_{Ei}$ (hr) |
|---|---|---|
| 1 | 14 | 17 |
| 2 | 15 | 19 |
| 3 | 10 | 15 |

To solve the example presented in Figure 5 by using the developed PNTS model, we have been coded and implemented the proposed approach and the example using MATLAB R2015a on the computer with Intel(R) Core (TM) i7-4702MQ CPU @ 2.20 GHz, 16 GB RAM.

Table 3. The optimal PNTS model solution and decision variables.

| Machines | Scheduled jobs | Max ($C_{ji}$) (hr) |
|---|---|---|
| M 1 | J 9, PM(J 11), J 2 | 31 |
| M 2 | J 4, PM(J 10), J6 | 29 |
| M 3 | J 8, J 1, PM(J 12), J 5, J 7, J 3 | 31 |

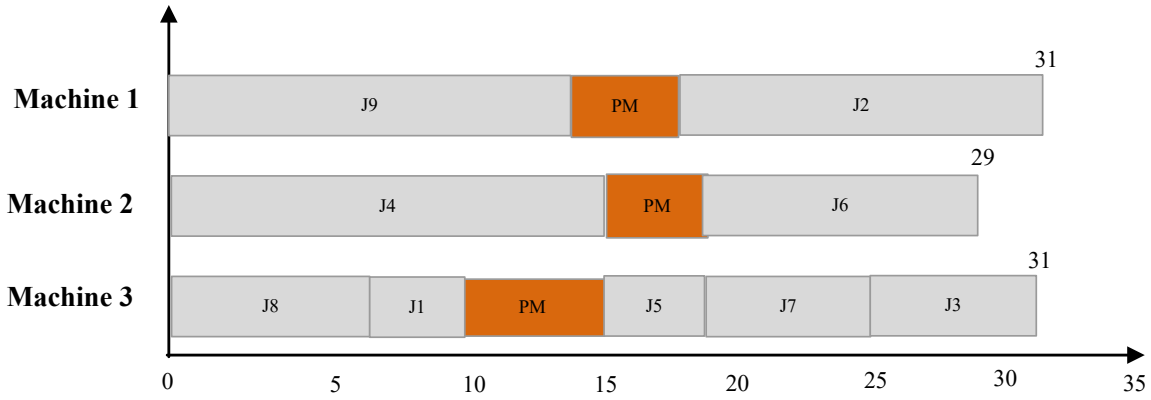


Figure 6. Gantt chart of the optimal sequence for jobs using PNTS.

## 5. Conclusion

This paper proposes an optimal solution to unrelated parallel machines scheduling problem subject to preventive maintenance to minimize the maximum completion time of jobs. To do this, a new hybrid Petri net and tabu search (PNTS) approach is proposed to find an optimal solution for this problem. The computational results show that the proposed PNTS are very effective and efficient to obtain optimal solutions. Further study should be considered for the assessment and implementation of some useful information in metaheuristics to evaluate how PNTS works with other optimization techniques. Moreover, the proposed method can be extended for the considered scheduling problem with limited buffer capacity constraint.

## Acknowledgments

This Project was funded by the National Plan for Science, Technology and Innovation (MAARIFAH), King Abdulaziz City for Science and Technology, Kingdom of Saudi Arabia, Award Number (14-ELE69-02).

## References

Al-Harkan, I. M. & Qamhan, A. A., Optimize Unrelated Parallel Machines Scheduling Problems With Multiple Limited Additional Resources, Sequence-Dependent Setup Times and Release Date Constraints, IEEE Access, vol. 7, pp. 171533-171547, 2019.

Alharkan, I., Saleh, M., Ghaleb, M. A., Kaid, H., Farhan, A. & Almarfadi, A., Tabu search and particle swarm optimization algorithms for two identical parallel machines scheduling problem with a single server, Journal of King Saud University-Engineering Sciences, vol. 32, no. 5, pp. 330-338, 2020.

Arroyo, J. E. C., Leung, J. Y.-T. J. C. & Research, O., Scheduling unrelated parallel batch processing machines with non-identical job sizes and unequal ready times, vol. 78, pp. 117-128, 2017.

Cardon, A., Galinho, T., Vacher, J.-P. J. R. & Systems, A., Genetic algorithms using multi-objectives in a multi-agent system, vol. 33, no. 2-3, pp. 179-190, 2000.

Fleszar, K., Charalambous, C. & Hindi, K. S. J. J. o. I. M., A variable neighborhood descent heuristic for the problem of makespan minimisation on unrelated parallel machines with setup times, vol. 23, no. 5, pp. 1949-1958, 2012.

Geyik, F. & Elibal, K. J. A. S. C., A linguistic approach to non-identical parallel processor scheduling with fuzzy processing times, vol. 55, pp. 63-71, 2017.

Glover, F., Tabu search—part I, ORSA Journal on computing, vol. 1, no. 3, pp. 190-206, 1989.

Joo, C. M., Kim, B. S. J. C. & Engineering, I., Hybrid genetic algorithms with dispatching rules for unrelated parallel machine scheduling with setup time and production availability, vol. 85, pp. 102-109, 2015.
Joo, C. M., Kim, B. S. J. I. E. & Systems, M., Non-identical parallel machine scheduling with sequence and machine dependent setup times using meta-heuristic algorithms, vol. 11, no. 1, pp. 114-122, 2012.
Kaid, H., Al-Ahmari, A. & Li, Z., Colored resource-oriented Petri net based ladder diagrams for PLC implementation in reconfigurable manufacturing systems, IEEE Access, vol. 8, pp. 217573-217591, 2020a.
Kaid, H., Al-Ahmari, A., Li, Z. & Davidrajuh, R., Intelligent colored token Petri nets for modeling, control, and validation of dynamic changes in reconfigurable manufacturing systems, Processes, vol. 8, no. 3, p. 358, 2020b.
Kaid, H., Al-Ahmari, A., Li, Z. & Davidrajuh, R., Single controller-based colored Petri nets for deadlock control in automated manufacturing systems, Processes, vol. 8, no. 1, p. 21, 2020c.
Kaid, H., Alharkan, I., Ghaleb, A. & Ghaleb, M. A., Metaheuristics for identical parallel machines scheduling to minimize mean tardiness, 2015 International Conference on Industrial Engineering and Operations Management (IEOM), pp. 1-6, 2015.
Kammoun, M. A., Ezzeddine, W., Rezg, N. & Achour, Z. J. A. S., FMS scheduling under availability constraint with supervisor based on timed Petri nets, vol. 7, no. 4, p. 399, 2017.
Lee, C.-Y., Machine scheduling with an availability constraint, Journal of global optimization, vol. 9, no. 3-4, pp. 395-416, 1996.
Lenstra, J. K. & Rinnooy Kan, A. J. O. R., Complexity of scheduling under precedence constraints, vol. 26, no. 1, pp. 22-35, 1978.
Moon, C., Lee, M., Seo, Y., Lee, Y. H. J. C. & engineering, i., Integrated machine tool selection and operation sequencing with capacity and precedence constraints using genetic algorithm, vol. 43, no. 3, pp. 605-621, 2002.
Muter, İ. J. E. J. o. O. R., Exact algorithms to minimize makespan on single and parallel batch processing machines, 2020.
Rezig, S., Ezzeddine, W., Turki, S. & Rezg, N. J. M., Mathematical Model for Production Plan Optimization—A Case Study of Discrete Event Systems, vol. 8, no. 6, p. 955, 2020.
Sethi, R. J. M. o. O. R., On the complexity of mean flow time scheduling, vol. 2, no. 4, pp. 320-330, 1977.
Sevindik, K. 2006. Parallel machine scheduling subject to machine availability constraints. Bilkent University.
Tan, M., Yang, H.-L. & Su, Y.-X. J. M. C., Genetic algorithms with greedy strategy for green batch scheduling on non-identical parallel machines, vol. 11, no. 4, pp. 439-452, 2019.
van der Aalst, W. M. J. O.-R.-S., Petri net based scheduling, vol. 18, no. 4, pp. 219-229, 1996.

## Biographies

**Husam Kaid** is a researcher in the Industrial Engineering Department, College of Engineering, King Saud University, Saudi Arabia. He received his BS in Industrial Engineering from the University of Taiz, Taiz, Yemen, in 2010. He received his PhD in Industrial Engineering from the King Saud University, Saudi Arabia, in 2021. He received his MS in Industrial Engineering from the King Saud University, Saudi Arabia, in 2015. His research areas and specialties are design and analysis of manufacturing systems, deadlock control in manufacturing systems, supply chain, simulation, operations research, optimization techniques, and bibliometric network analysis.

**Abdulrahman Al-Ahmari** a Professor of industrial engineering with King Saud University, Riyadh, Saudi Arabia. He received the Ph.D. degree in manufacturing systems engineering from the University of Sheffield, Sheffield, U.K., in 1998. He worked as Dean of the Advanced Manufacturing Institute, Chairman of Industrial Engineering Department and He led a number of funded projects from different organizations in Saudi Arabia. He has published papers in leading Journal of Industrial and Manufacturing Engineering. His current research interests include advanced manufacturing technologies, Petri nets, analysis and design of manufacturing systems, computer integrated manufacturing, optimization of manufacturing operations, flexible manufacturing systems and cellular manufacturing systems, and applications of decision support systems in manufacturing.

**Emad Abouel Nasr** is a Professor in Industrial Engineering Department, College of Engineering, King Saud University, Saudi Arabia, and Mechanical Engineering Department, Faculty of Engineering, Helwan University, Egypt. He received his PhD in Industrial Engineering from University of Houston, TX, USA, in 2005. His current research focuses on CAD, CAM, rapid prototyping, advanced manufacturing systems, and collaborative engineering.

**Emad Abouel Nasr** received the Ph.D. degree in industrial engineering from the University of Houston, TX, USA, in 2005. He is currently a Professor with the Industrial Engineering Department, College of Engineering, King Saud University, Saudi Arabia, and an Associate Professor with Mechanical Engineering Department, Faculty of

Engineering, Helwan University, Egypt. His current research interests include CAD, CAM, rapid prototyping, advanced manufacturing systems, supply chain management, and collaborative engineering.

**Adel Al-Shayea** was a consultant at King Saud University Rector's Of_ce and worked for SABIC Marketing Ltd., Riyadh, and at the Institute of Public Administration (IPA). In addition, he is a consultant in the King Abdullah Institute of Research and Consulting Studies. He is a consultant industrial engineer (CE-SCE). He is a member of the Saudi Council of Engineers (SCE), Saudi Arabia, as well as a member of several committees such as the national committee for the codification and standardization of operation and maintenance works. He is currently the Assistant Vice President for academic and educational affairs at King Saud University. He participated and conducted several consultative works for governmental and private organizations. He also refereed several engineering works in Saudi Arabia.

**Ali K. Kamrani** received the B.S. degree in electrical engineering, the M.Eng. degree in electrical engineering, the M.Eng. degree in computer science and engineering mathematics, and the Ph.D. degree in industrial engineering from the University of Louisville, Louisville, Kentucky. He is currently an Associate Professor of industrial engineering with the University of Houston and the departments Ph.D. Program Coordinator and Advisor. He is the Founder and the Director of the Free Form Fabrication and Rapid Prototyping (FFF&RP) Laboratory. He is also developing the Design and Manufacturing Automation (D&MA) Laboratory, College of Engineering. Prior to joining the University of Houston, he was an Associate Professor of industrial and manufacturing systems engineering with the University of Michigan-Dearborn, an Adjunct Professor Faculty with Wayne State University, and a Faculty Member for Program in Manufacturing (PIM), the University of Michigan at Ann Arbor Interdisciplinary Program. He is the Founder and previous Coordinator of the Rapid Prototyping Laboratory, College of Engineering and Computer Science, University of Michigan-Dearborn.

**Mohammed A. El-Meligy** received the B.Sc. degree in information technology from Menoufiya University, Shibin Al Kawm, Egypt, in 2005. Since 2009, he has been a Software Engineer with King Saud University, Riyadh, Saudi Arabia. His research interests include Petri nets, supervisory control of discrete event systems, database software, and network administration.

**Haitham A. Mahmoud** received the Ph.D. degree in industrial engineering from the University of Helwan, Egypt, in 2012. He is currently an Assistant Professor in the Department of Industrial Engineering, College of Engineering, King Saud University, Riyadh, Saudi Arabia, and the Mechanical Engineering Department, Faculty of Engineering, Helwan University, Egypt. He worked as an engineering consultant for several industrial organizations in Egypt. His current research interests include optimization modeling, theory, and algorithm design with applications in waste management and energy management, financial engineering, and big data.