# Scheduling Robotic Cellular Manufacturing Systems With Timed Petri Net, A* Search, and Admissible Heuristic Function

Bo Huang, *Senior Member, IEEE*, MengChu Zhou, *Fellow, IEEE*, Abdullah Abusorrah, *Senior Member, IEEE*, and Khaled Sedraoui

***Abstract*—System scheduling is a decision-making process that plays an important role in improving the performance of robotic cellular manufacturing (RCM) systems. Timed Petri nets (PNs) are a formalism suitable for graphically and concisely modeling such systems and obtaining their reachable state graphs. Within their reachability graphs, timed PNs' evolution and intelligent search algorithms can be combined to find an efficient operation sequence from an initial state to a goal one for the underlying systems of the nets. To schedule RCM systems, this work proposes an A* search with a new heuristic function based on timed PNs. When compared with related approaches, the proposed one can deal with token remaining time, weighted arcs, and multiple resource copies commonly seen in the PN models of RCM systems. The admissibility of the proposed heuristic function is proved. Finally, experimental results are given to show the effectiveness and efficiency of the proposed method and heuristic function.**

***Note to Practitioners*—Robotic cellular manufacturing (RCM) systems are among the most common and complicated discrete-event dynamic systems, which provide a great number of choices of resources and processing routes to allow high system productivity. Timed Petri nets (PNs) and intelligent search algorithms on their reachability graphs are ideal tools to handle the RCM scheduling problem. This work proposes an A* search method based on the evolutions of timed PNs to optimally schedule RCM systems. The proposed method can deal with token remaining time, weighted arcs, and multiple resource copies often encountered in the PN models of RCM systems.**

***Index Terms*—Heuristic search, Petri nets (PNs), robotic cellular manufacturing (RCM) systems, system scheduling.**

Manuscript received July 15, 2020; revised September 2, 2020; accepted September 21, 2020. Date of publication October 20, 2020; date of current version January 6, 2022. This article was recommended for publication by Associate Editor M. Wimmer and Editor J. Li upon evaluation of the reviewers' comments. This work was supported in part by the National Natural Science Foundation of China under Grant 61773206, in part by the Natural Science Foundation of Jiangsu Province of China under Grant BK20170131, in part by the Deanship of Scientific Research (DSR) at King Abdulaziz University, Jeddah, Saudi Arabia, under Grant RG-21-135-38, and in part by the Jiangsu Overseas Visiting Scholar Program for University Prominent Young and Middle-Aged Teachers and Presidents. *(Corresponding author: MengChu Zhou.)*

Bo Huang is with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: huangbo@njust.edu.cn).

MengChu Zhou is with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA, and also with the Center of Research Excellence in Renewable Energy and Power Systems, King Abdulaziz University, Jeddah 21481, Saudi Arabia (e-mail: zhou@njit.edu).

Abdullah Abusorrah and Khaled Sedraoui are with the Department of Electrical and Computer Engineering, King Abdulaziz University, Jeddah 21481, Saudi Arabia, and also with the Center of Research Excellence in Renewable Energy and Power Systems, King Abdulaziz University, Jeddah 21481, Saudi Arabia (e-mail: aabusorrah@kau.edu.sa; sedraoui@yahoo.com).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASE.2020.3026351

## I. Introduction

ROBOTIC cellular manufacturing (RCM) systems are very common in today's manufacturing industry, including automobiles, semiconductors, and electronics. Such systems provide us a great number of choices of resources and processing routes. They bring a challenging problem, i.e., how to allocate available resources to different robots and machines and schedule different operations to achieve the highest production efficiency. It belongs to nondeterministic polynomial-time (NP) hard problems [1].

RCM scheduling problems can be solved by mathematical programming [2]–[7] and metaheuristic scheduling methods [8], [9]. In general, a mathematical programming method ensures the optimality of its obtained solutions, but it has some limitations in handling complex systems since it is difficult to describe their structures, such as assembly, disassembly, and alternative routes. Metaheuristic methods, such as simulated annealing, genetic algorithm, and tabu search [10], [11], can be used to obtain suboptimal schedules for scheduling problems in a reasonable time. They can provide good solutions but hardly tell how close their solutions are to the optimal one. Thus, a desirable scheduling method for RCM systems should include both easy formulation of the systems and quick calculation of optimal solutions.

Petri nets (PNs) can well model discrete-event systems (DESs) with sequence, concurrency, conflict, and synchronization. Also, their reachability graphs fully reflect the behavior of their underlying systems [12]. Thus, they are an ideal tool to model and schedule RCM systems, which belong to DESs. Based on PNs' reachability graphs, Lee and DiCesare [13] pioneered in using a well-known intelligent A* search in the system scheduling. The method needs not explore the entire but a partial reachability graph to search for an optimal schedule by using an admissible heuristic function. The heuristic function is critical to the quality of the obtained results. If the used heuristic function is admissible, i.e., for any reachable marking, the remaining cost from the marking to the goal marking estimated by the heuristic function does not exceed the actual minimal one, the obtained schedule is optimal. In the literature, several admissible heuristic functions have been proposed for the A* search in PNs' reachability graphs to obtain optimal system schedules.

The work in [14] proposes a PN-related heuristic function based on a resource cost reachability matrix for the scheduling of manufacturing systems. It uses the minimal cost required by a token to move from its current place to its corresponding end place as the heuristic value of the current marking. However, according to [15], it is inadmissible due to its ignorance of the token remaining time in the heuristic function calculations. Thus, it cannot guarantee the optimality of its obtained schedules. Then, the work in [15] proposes two admissible heuristic functions based on the maximal cost required for processing a part from the current state to a goal state.

Xiong and Zhou [16] propose an admissible heuristic function by using the total operation cost required for the remaining operations that need the same resource to be executed from the current marking to a goal marking. It is highly informed and widely used [17]–[19]. However, it does not consider token remaining time in the PN evolution. The work in [20] utilizes the maximal value of the minimal completion time of a token in a marked place as the heuristic function of the current state. It is an admissible heuristic function that considers token remaining time in a net.

Huang *et al.* [21] propose a method that can simultaneously use several existing heuristics, which are admissible or inadmissible, to form a new heuristic function for the A* search in reachability graphs. The resultant heuristic function is guaranteed to be admissible and more informed than any of its constituents. Luo *et al.* [22] propose two admissible heuristics to deal with the deadlock-free scheduling of automated manufacturing systems. The first one uses the average time of a part token from its place to its end one as an estimated cost that underestimates the optimal cost from the current state to a goal one. The second one is similar to the first one but adds resource idle time to the heuristic function. Hence, its estimated heuristic value of a state is not less than that of the first function. Since resource idle time is necessary for any state transfer, the second admissible heuristic function is more informed than the first one.

The work in [23] proposes a combinational heuristic function for the scheduling of open shop manufacturing systems based on PN reachability graphs. It uses the maximal value in average remaining work time of a resource, maximal remaining work time of a job, and maximal remaining workload of a resource, each of which is a necessary cost of a state transfer from the current state to a goal one, as the estimated optimal cost. Therefore, it is admissible. In addition, it proposes another admissible heuristic function for the system scheduling by using some deadlock structures in ordinary PNs in which arc weights are all one.

Among the abovementioned heuristic functions, some of them, such as those in [20], [22], and [23], consider token remaining time, but all of them fail to handle the PNs with weighted arcs and multiple resource copies. In fact, RCM systems may have multiple robots and machines of the same type and operations that require multiple resources. In order to schedule such systems, this work proposes a timed PN-based A* search method. Its main contributions are as follows.

1) Timed PN models for the scheduling of RCM systems and their related features are revealed, which play an important role in their evolution.
2) Based on the presented features, an A* search algorithm with a novel heuristic function that evaluates and compares states in terms of both markings and token remaining time is proposed to search for the schedules of the underlying RCM systems.
3) In comparison with the existing methods [14]–[23], the proposed one can deal with the nets that have not only token remaining time but also weighted arcs and multiple resource copies and obtain their optimal schedules.

Section II reviews the preliminaries of PNs. The features of place-timed PNs for RCM scheduling and A* search in a PN's reachability graph by considering token remaining time are given in Section III. Section IV proposes an admissible heuristic function that can deal with token remaining time, weighted arcs, and multiple resource copies. Section V demonstrates the proposed method by scheduling some RCM systems. Finally, conclusions are given in Section VI.

## II. Basics of Petri Nets

PNs are a graphical and mathematical tool suitable to describe, model, and analyze the behavior of DESs [24], [25]. A PN is defined as a four-tuple $N = (P, T, F, W)$ in which $P = \{p_1, p_2, \ldots, p_m\}$ and $T = \{t_1, t_2, \ldots, t_n\}$ are, respectively, the finite nonempty sets of places and transitions such that $m, n \in \mathbb{Z}^+$ and $P \cap T = \emptyset$, where $\mathbb{Z}^+$ denotes the set of positive integers. $F \subseteq (P \times T) \cup (T \times P)$ denotes the set of directed arcs between places and transitions. $W : F \to \mathbb{Z}^+$ is a weight assignment for arcs.

For a place $p \in P$, ${}^\bullet p = \{t \in T | (t, p) \in F\}$ is called the set of $p$'s pretransitions and $p^\bullet = \{t \in T | (p, t) \in F\}$ is called the set of $p$'s posttransitions. Similarly, for a transition $t \in T$, ${}^\bullet t = \{p \in P | (p, t) \in F\}$ and $t^\bullet = \{p \in P | (t, p) \in F\}$ are called the sets of $t$'s preplaces and postplaces, respectively.

Let $\mathbb{N}$ be the set of natural numbers, i.e., $\mathbb{N} = \{0, 1, 2, \ldots\}$. For a net $N$, $M : P \to \mathbb{N}$ is called a marking, which is a token distribution in places. $M_0$ is used to denote an initial marking of $N$. A place $p$ is said to be marked at $M$ if $M(p) > 0$. At a marking $M$, if $\forall p \in {}^\bullet t, M(p) \geqslant W(p, t)$, then $t$ is enabled at $M$. If a transition $t$ enabled at $M$ is selected to fire, it generates a new marking $M'$ such that $\forall p \in P$, $M'(p) = M(p) - W(p, t) + W(t, p)$. This process is denoted as $M[t\rangle M'$. If firing a sequence of transitions at a marking $M$ generates a new marking $M'$, then $M'$ is called reachable from $M$. The set of markings that are reachable from $M$ is denoted by $R(N, M)$. A pair $(N, M_0)$ is called a net system.

Given a net system $(N, M_0)$, a place invariant $I : P \to \mathbb{Z}$ is a $|P|$-dimensional integer vector such that $I \neq [0]$, $I^T[N] = [0]^T$, and $\forall M \in R(N, M_0)$, $I^T M = I^T M_0$, where $[0]$ denotes a $|P|$-dimensional zero vector and $\mathbb{Z}$ denotes the set of integers. $I$ is called a P-semiflow if all elements of $I$ are nonnegative.

A net system $(N, M_0)$ is bounded if $\forall M \in R(N, M_0)$, $\forall p \in P$, $\exists k \in \mathbb{Z}^+$, $M(p) \leq k$. For a place $p \in P$, if $\forall M \in R(N, M_0)$, $M(p) \leq k$, then $p$ is said to be $k$-bounded. A net system $(N, M_0)$ is $k$-bounded if $\forall p \in P$, $p$ is $k$-bounded.

Given a net system $(N, M_0)$, $G(N, M_0)$ denotes its reachability graph in which each vertex represents a reachable marking and each edge denotes a marking transfer $(M, M')$ in which $M$ is called an immediate predecessor of $M'$ and $M'$ is called an immediate successor of $M$.

## III. System Scheduling With Timed PNs

This section summarizes the features of place-timed PNs for the scheduling of RCM systems. A* search in reachability graphs of place-timed PNs is developed by considering token remaining time in net evolution and state comparisons.

### A. Timed PNs for System Scheduling

There exist many classes of PNs for the modeling and control of manufacturing systems [24], e.g., LS$^3$PR, S$^3$PR, S$^4$PR, and S*PR. Generally, they have three kinds of places, i.e., idle ones $P_0$, activity ones $P_A$, and resource ones $P_R$. The differences among them are the number of resources required by an operation and the net structures of process subnets. They are all untimed PNs and cannot be directly used for system scheduling. However, they can be slightly modified for system scheduling by dividing each idle place into a start place and an end one, putting tokens of idle places into their corresponding start places, and adding operation time into activity places. The characteristics of the resultant place-timed PNs are summarized as follows.

*Definition 1:* A place-timed PN for the system scheduling is $\mathcal{N} = (P_S \cup P_E \cup P_R \cup P_A, T, F, W, D)$ that consists of a set of process subnets $\mathcal{N}^x = (P_S^x \cup P_E^x \cup P_R^x \cup P_A^x, T^x, F^x, W^x, D^x)$, and $x \in \mathbb{N}_J = \{i \in \mathbb{Z}^+ | \mathcal{N}^i$ is the $i$th process subnet of $\mathcal{N}\}$, such that the subnets share some resource places and the following statements are satisfied.

1) $P_S^x$ (resp. $P_E^x$) is called the start (resp. end) place of $\mathcal{N}^x$. $P_A^x$ (resp. $P_R^x$) is called the set of activity (resp. resource) places of $\mathcal{N}^x$.
2) $P_S^x \cap P_A^x = P_E^x \cap P_A^x = \emptyset$ and $(P_S^x \cup P_E^x \cup P_A^x) \cap P_R^x = \emptyset$.
3) $W = W_A \cup W_R$ with $W_A : F \cap ((P_A \cup P_S \cup P_E) \times T) \cup (T \times (P_A \cup P_S \cup P_E)) \to \{1\}$ and $W_R : F \cap ((P_R \times T) \cup (T \times P_R)) \to \mathbb{Z}^+$.
4) $\forall r \in P_R$, there exists a unique minimal P-semiflow $I_r$ such that $\|I_r\| \cap P_R = \{r\}$, $\|I_r\| \cap (P_A \cup P_S \cup P_E) \neq \emptyset$, and $I_r(r) = 1$. In addition, $P_A \subseteq \bigcup_{r \in P_R}(\|I_r\| \setminus \{r\})$.
5) The places shared by $\mathcal{N}^x$ and $\mathcal{N}^y (x \neq y)$ belong to $P_R$.
6) $D : P_A \to \mathbb{N}$ is an assignment of operation time to activity places.

In Definition 1, item 1 indicates that the net has no idle places; instead, it has a start place and an end place in each process subnet to differentiate the goal marking from the initial marking. Item 2 signifies that $P_A$, $P_R$, $P_S$, and $P_E$ are mutually exclusive. Item 3 denotes that the weights of arcs between transitions and activity places, start places, as well as end places are equal to one and the weights of the arcs between transitions and resource places may be greater than one. In other words, parts are produced one by one and each operation/process may require multiple resources. In addition, there is no arc between a transition of a process subnet and an activity, start, and end place of another process subnet. Item 4 enforces resource preservation of the system, indicating that there exists a P-semiflow on a resource place and some activity, start, and end places.

### B. A* Search With Tokens' Remaining Time

A PN's reachability graph fully reflects the behavior of its underlying systems. In [13], [14], and [16], an intelligent A* search is applied to the PN's reachability graph to find an optimal or suboptimal schedule from an initial marking to a goal marking. This approach needs not explore the entire reachability graph to obtain a schedule. However, it cannot be directly applied to a place-timed PN $\mathcal{N}$. The reason is as follows. The method uses two lists of markings, i.e., an OPEN list to contain markings that are generated but not expanded and a CLOSED list to contain markings that have been expanded. Whenever a new marking $M'$ is generated, $M'$ is compared with the markings in OPEN and CLOSED to check whether or not it already exists. If the same marking $M$ as $M'$ exists, their $g$-values are compared to judge whether the path already found from $M_0$ to $M$ is more promising than the path from $M_0$ to $M'$. However, the approach does not consider the tokens' availability in such comparisons. In a place-timed PN $\mathcal{N}$, time information is associated with places and each token in timed places has a time denoting how long it will be available in the place. Thus, more than one state may have the same marking but different token remaining time. Therefore, the token remaining time should also be considered in state comparisons; otherwise, it may prune some promising states that have the same marking but less token remaining time.

To apply A* search to the reachability graph of a place-timed PN $\mathcal{N}$, not only the tokens' distribution (i.e., a marking $M$) but also the token remaining time is necessary to represent a state of the net. In such a net, a token in a timed place $p$ means that this token is already available for the firing of a posttransition of $p$ or although the token is now unavailable for the firing, it will become available after a fixed time. Note that in the second case, a nonzero remaining time is needed for the token to be available. As time elapses, the token's remaining time decreases until it reaches zero. Then, the token becomes available for the firing. Therefore, token remaining time is as important as tokens' distribution for a net to "reach" a certain state of the underlying system.

*Definition 2:* Given a place-timed PN $\mathcal{N}$ whose activity places are $k$-bounded. A state of $\mathcal{N}$ is defined as $S = (M, Q)$, where $M$ denotes the marking of the state and $Q$ denotes a $|P| \times k$ token remaining time matrix that contains the token remaining time of all places. $S_0$ denotes an initial state of $\mathcal{N}$ and $(\mathcal{N}, S_0)$ is called a timed net system of $\mathcal{N}$.

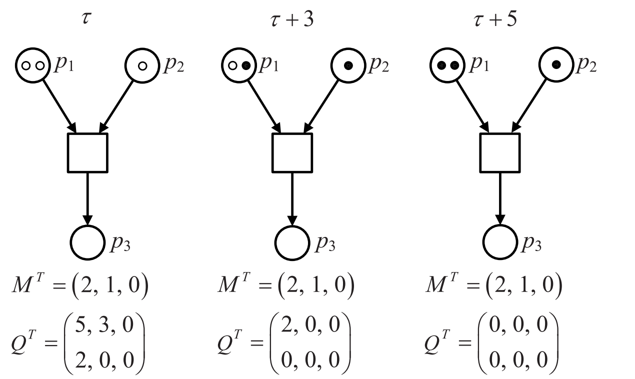


Fig. 1. Three states with a same marking but different token remaining time.

For example, consider three different states of a simple place-timed net as shown in Fig. 1, in which hollow dots in places denote unavailable tokens and black dots in places denote available tokens. Suppose that $p_1$ and $p_2$ are two-bounded timed places. We can see that all the states have the same marking $M$, but they are different states at the time $\tau$, $\tau + 3$, and $\tau + 5$ since they have different token remaining time. For the state in the left of Fig. 1, although a transition is enabled at its tokens' distribution $M$, it is not enabled at its state since not all the tokens required by the firing of the transition are available at that time. The transitions in the rest two states are enabled at both their tokens' distribution and their token remaining time, i.e., they are enabled at their states where black dots represent tokens that can be used to enable their related transitions.

In a place-timed PN $\mathcal{N}$, firing an enabled transition changes the tokens' distribution $M$ and the tokens' remaining time matrix $Q$ at a state $S$ and then generates a new state. Firing a sequence of transitions thus generates a sequence of states. Given a place-timed PN $\mathcal{N}$, all possible behaviors of the underlying system can be completely tracked by reachable states in the reachability graph of the timed net. Therefore, the scheduling problem of $\mathcal{N}$ becomes searching for a transition firing sequence in the reachability graph of $\mathcal{N}$ from its initial state $S_0$ to a given goal state $S_G$. The A* search in the reachability graph of $\mathcal{N}$ is given in Algorithm 1.

In this algorithm, $c(S, S')$ denotes the cost needed in the state transfer from $S$ to $S'$ and a heuristic function $h$ is to evaluate each generated state $S$ by using $f(S) = g(S) + h(S)$, where $f(S)$ is an estimate of the cost from $S_0$ to $S_G$ along an optimal path that goes through $S$, $g(S)$ is the cost of the marking transfer from $S_0$ to $S$, and $h(S)$ is an estimate of the optimal cost from $S$ to $S_G$. Then, the newly generated states are compared with the states in OPEN and CLOSED in terms of their markings, token remaining time, and $g$-values to decide how to process them following Steps 6.1–6.3.

## IV. Admissible Heuristic Function

For the proposed A* search, this section formulates a heuristic function that can deal with token remaining time, alternative routes, weighted arcs, and multiple resource copies in place-timed PNs of RCM systems.

**Algorithm 1** A* Search-Based Scheduling

Input: A place-timed PN $\mathcal{N}$, an initial state $S_0$, and a goal state $S_G$.

Output: A transition firing sequence from $S_0$ to $S_G$ if it exists.

1) Insert $S_0$ into the OPEN list.
2) If OPEN is empty, terminate with failure.
3) Remove the first state $S = (M, Q)$ from OPEN and put $S$ in the CLOSED list.
4) If $S$ equals $S_G$, construct the path from $S$ back to $S_0$ and terminate.
5) Find the set of transitions enabled at $M$.
6) For each transition $t$ enabled at $M$, decreases the token remaining time to enable $t$ at both $M$ and $Q$, generate an immediate succeeding state $S' = (M', Q')$ by firing $t$, set a pointer from $S'$ to $S$, and calculate $g(S') = g(S) + c(S, S')$, $h(S')$, and $f(S') = g(S') + h(S')$.
   6.1) If there exists a state $S^O = (M^O, Q^O)$ in OPEN such that $M' = M^O$, $Q' = Q^O$, and $g(S') < g(S^O)$, replace $S^O$ with $S'$.
   6.2) If there exists a state $S^C = (M^C, Q^C)$ in CLOSED such that $M' = M^C$, $Q' = Q^C$, and $g(S') < g(S^C)$, delete $S^C$ from CLOSED and insert $S'$ into OPEN.
   6.3) If there is no state $S'' = (M'', Q'')$ in both OPEN and CLOSED such that $M' = M''$ and $Q' = Q''$, insert $S'$ into OPEN.
7) Reorder OPEN in non-decreasing order of $f$-values.
8) Go to Step 2).

*Definition 3:* Let $h^*(S)$ be the lowest cost of the state transfer from $S$ to $S_G$. A heuristic function $h$ in an A* search is said to be admissible if, for any reachable state $S$, $h(S) \leqslant h^*(S)$ holds.

A useful feature of an A* algorithm is that if it adopts an admissible heuristic function, its obtained result is guaranteed to be optimal [26]. As introduced in Section I, several admissible heuristic functions have been proposed in [15], [16], and [21]–[23] for the A* search in the PNs' reachability graphs of manufacturing systems. However, those heuristic functions cannot handle the place-timed PNs of RCM systems with weighted arcs and multiple resource copies.

As an example, consider an RCM system in [20]. It has three types of shared robots $R_1$–$R_3$, each of which has 1, 1, and 2 units of robots, respectively. It has four jobs $J_1$–$J_4$, each of which has three tasks $K_1$–$K_3$. The job requirements of the system are shown in Table I in which operation time is given in parentheses. In the system, some tasks may have alternative processing routes and some operations may require multiple robots. For example, $K_3$ of $J_1$ can be performed with either $R_1$ for 80 units of time or $R_1$ with 57 units of time and two $R_3$ with 51 units of time. Note that between any two sequential operations, there is an intermediate buffer to store parts for its next operation. Its place-timed PN is shown

TABLE I
JOB REQUIREMENTS OF THE EXAMPLE SYSTEM

| | $J_1$ | $J_2$ | $J_3$ | $J_4$ |
|---|---|---|---|---|
| $K_1$ | $R_3(69)$ or $R_2(75)$ | $R_2(95)$ | $R_2(78)$ | $R_3(99)$ |
| $K_2$ | $R_3(85)$ | $R_2(85)$ or $R_3(97)$ | $R_3(75)$ | $R_3(76)$ |
| $K_3$ | $R_1(80)$ or $[R_1(57)$ and $2R_3(51)]$ | $R_1(98)$ or $R_3(92)$ | $R_1(68)$ or $[R_3(56)$ and $R_2(70)]$ | $R_1(93)$ |

in Fig. 2 where activity places without an integer number represent the intermediate buffers and resource places with the same name represent the same type of robot. This net has operation time associated with activity places. In addition, it has weighted arcs and multiple copies of resources, which have not been considered by any existing A* searches with admissible heuristic functions.

In the following, a new heuristic function for our proposed A* search is formulated to schedule the place-timed PNs of RCM systems. Before introducing it, we first define two matrices: a weighted operation time (WOT) matrix $\Theta$ and a weighted resource time (WRT) matrix $\Gamma$. $\Theta$ is a $|P_A| \times |P_R|$ matrix in a place-timed PN $\mathcal{N}$ such that if an activity place $p$ requires a kind of resource $r$, $\Theta(p,r)$ is the WOT needed by an available token in $p$ with $r$ if all units of $r$ are assumed to be concurrently used, i.e., $\Theta(p,r) = D(p) \times (U_p(r)/M_0(r))$, where $U_p(r)$ denotes the units of $r$ that are required by the operation of $p$. For example, in the net of Fig. 2, the activity place $p_5$ requires $R_3$, i.e., $p_{40}$, and the operation time of $p_5$ is $D(p_5) = 85$. Since $U_{p_5}(R_3) = 1$ and $M_0(R_3) = 2$, we have that $\Theta(p_5, R_3) = 85 \times \frac{1}{2} = 42.5$. Similarly, $\Theta(p_8, R_1) = 57$ since $D(p_8) = 57$ and $U_{p_8}(R_1) = M_0(R_1) = 1$. $\Gamma$ is a $|P \setminus P_R| \times |P_R|$ matrix such that $\forall p \in P \setminus P_R$ and $\forall r \in P_R$, $\Gamma(p,r)$ is the minimal total time of WOTs with $r$ that is needed by an available token in $p$ to go from $p$ to its end place. The WRT matrix of the net in Fig. 2 is shown as follows:

$$\Gamma = \left[\begin{array}{lrrr lrrr}
p_1: & 57 & 0 & 42.5; & p_{21}: & 0 & 78 & 37.5;\\
p_2: & 57 & 0 & 42.5; & p_{22}: & 0 & 0 & 37.5;\\
p_3: & 57 & 0 & 42.5; & p_{23}: & 0 & 0 & 37.5;\\
p_4: & 57 & 0 & 42.5; & p_{24}: & 0 & 0 & 0;\\
p_5: & 57 & 0 & 0; & p_{25}: & 0 & 0 & 0;\\
p_6: & 57 & 0 & 0; & p_{26}: & 0 & 0 & 0;\\
p_7: & 0 & 0 & 0; & p_{27}: & 0 & 70 & 0;\\
p_8: & 0 & 0 & 51; & p_{28}: & 0 & 70 & 0;\\
p_9: & 0 & 0 & 51; & p_{29}: & 0 & 0 & 0;\\
p_{10}: & 0 & 0 & 0; & p_{30}: & 0 & 0 & 0;\\
p_{11}: & 0 & 0 & 0; & p_{31}: & 93 & 0 & 87.5;\\
p_{12}: & 0 & 95 & 0; & p_{32}: & 93 & 0 & 38;\\
p_{13}: & 0 & 0 & 0; & p_{33}: & 93 & 0 & 38;\\
p_{14}: & 0 & 0 & 0; & p_{34}: & 93 & 0 & 0;\\
p_{15}: & 0 & 0 & 0; & p_{35}: & 93 & 0 & 0;\\
p_{16}: & 0 & 0 & 0; & p_{36}: & 0 & 0 & 0;\\
p_{17}: & 0 & 0 & 0; & p_{37}: & 0 & 0 & 0;\\
p_{18}: & 0 & 0 & 0; & & & & \\
p_{19}: & 0 & 0 & 0; & & & & \\
p_{20}: & 0 & 0 & 0; & & & &
\end{array}\right].$$

For example, consider an available token in $p_1$. The token needs at least 57 units of operation time with $R_1$ to go from $p_1$ to its end place $p_{11}$, e.g., by using a transition firing sequence $t_1-t_2-t_5-t_6-t_9-t_{10}-t_{11}-t_{12}$. Similarly, it needs at least zero units of time with $R_2$ to reach $p_{11}$, e.g., by using a transition firing sequence $t_1-t_2-t_5-t_6-t_7-t_8$, and at least 42.5 units of time with $R_3$ to reach $p_{11}$, e.g., by using a transition firing sequence $t_3-t_4-t_5-t_6-t_7-t_8$. Therefore, $\Gamma(p_1,\cdot) = (57, 0, 42.5)$, meaning that $\Gamma(p_1, R_1) = 57$, $\Gamma(p_1, R_2) = 0$, and $\Gamma(p_1, R_3) = 42.5$.

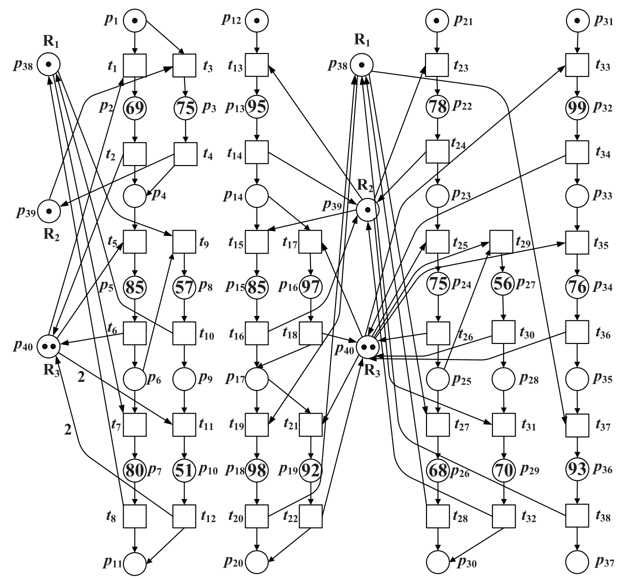

Fig. 2. Place-timed PN of an RCM system [20].

Suppose that all activity places are $k$-bounded. Based on the WRT matrix, a heuristic function for the proposed A* search to schedule RCM systems can be given as

$$h_H(S) = \max_{r \in P_R} \left\{ \sum_{p \in P \setminus P_R} \left[ M(p) \cdot \Gamma(p,r) + \sum_{x \in \mathbb{N}_k} Q(p,x) \cdot \frac{U_p(r)}{M_0(r)} \right] \right\} \quad (1)$$

where $\mathbb{N}_k = \{1, \ldots, k\}$. In (1), $M(p) \cdot \Gamma(p,r)$ denotes the total WRT with $r$ for all tokens in $p$ at $S$ to reach the end place of $p$, while the tokens are supposed to be available. $\sum_{x \in \mathbb{N}_k} Q(p,x) \cdot U_p(r)/M_0(r)$ denotes the remaining WRT with $r$ for all tokens in $p$ at $S$ to become available. Note that in the above calculations, all units of $r$ are assumed to be concurrently used. Thus, the expression in the braces of (1) represents the total WRT with $r$ for all tokens in nonresource places at $S$ to reach their end places.

Therefore, $h_H(S)$ represents the maximal total WRT required by the net to transfer from $S$ to $S_G$ with a certain kind of resource.

The design of $h_H(S)$ is motivated by the fact that a lower bound of the cost of the state transfer from $S$ to $S_G$ can be calculated as the maximal weighted usage of a kind of resource. The proposed heuristic function $h_H$ can be used for place-timed PNs of RCM systems that may have alternative routes, weighted arcs, and multiple resource copies. In addition, the token remaining time is also considered in the calculation of $h_H$. As an example, consider the net shown in Fig. 2. Suppose that $S_G$ is the state where all nonresource tokens are in their end places. Since $M_0(p_1) = M_0(p_{12}) = M_0(p_{21}) = M_0(p_{31}) = 1$, $\Gamma(p_1, \cdot) = (57, 0, 42.5)$, $\Gamma(p_{12}, \cdot) = (0, 95, 0)$, $\Gamma(p_{21}, \cdot) = (0, 78, 37.5)$, $\Gamma(p_{31}, \cdot) = (93, 0, 87.5)$, and $\forall r \in P_R$, $U_{p_1}(r) = U_{p_{12}}(r) = U_{p_{21}}(r) = U_{p_{31}}(r) = 0$, the estimated heuristic value of the initial state is $h_H(S_0) = \max\{57+0+0+93, 0+95+78+0, 42.5+0+37.5+87.5\} = 173$, i.e., the maximal total WRT needed by the net with a kind of resource to transfer from $S_0$ to $S_G$ is 173 with $R_2$. Next, we prove that $h_H$ is an admissible heuristic function, which guarantees that the obtained schedules are optimal.

*Theorem 1:* $h_H$ is admissible for a place-timed PN $\mathcal{N}$.

*Proof:* According to (1), $h_H(S)$ denotes the maximal total WRT for all tokens in nonresource places at $S$ to reach their end places with a kind of resource $r$ in $\mathcal{N}$. This calculation is under the assumption that all units of $r$ are concurrently used. In practice, the units of $r$ are usually not maximally used by operations that require $r$. In addition, there may exist some operations that require other types of resources and they also need operation time. Therefore, $h_H(S)$ is a lower bound of the time needed by the net to transfer from $S$ to $S_G$, i.e., $\forall S$, $h_H(S) \leqslant h^*(S)$. According to Definition 3, $h_H$ is admissible. ■

## V. Illustrative Experiments

In this section, some RCM systems are tested with the proposed A* search in the reachability graphs of their PN models. C# programs have been developed and run on a Windows computer with an 8-GB memory and an Intel i5 Core 2.5-GHz CPU. The source codes of the presented method and all experimental files of this article are available in [27].

First, the place-timed PN shown in Fig. 2 is considered. The traditional A* search with an admissible heuristic function in [16] and [21]–[23] cannot optimally schedule it due to weighted arcs and multiple copies of resources in the net. For example, the work in [16] proposes an admissible heuristic function $h(S) = \max_{r\in P_R}\{\xi_r(S)\}$, in which $\xi_r(S)$ denotes the total operation time of the remaining operations to be executed with the resource $r$ from $S$ to $S_G$. However, it becomes inadmissible for the nets with weighted arcs and multiple resource copies. The reason is that weighted arcs and multiple resource copies may lead to concurrent usage of a kind of resources by different operations, and $\max_{r\in P_A}\{\xi_r(S)\}$ can thus exceed the actual minimal cost of the state transfer from $S$ to $S_G$. In contrast, the proposed A* search with its heuristic function $h_H$ is able to handle such cases with weighted arcs and multiple resource copies and optimally solve the net in about 45 min. The number of expanded timed states is 83 730. The obtained schedule is shown in Table II in terms of a series of transitions and their firing time. The makespan of the obtained schedule is 350, which is guaranteed to be minimal for the underlying system since $h_H$ is admissible.

TABLE II
Obtained Schedule for the Place-Timed PN in Fig. 2 (Makespan = 350)

| $t$ | Firing Time | $t$ | Firing Time | $t$ | Firing Time |
|---|---|---|---|---|---|
| $t_1$ | 0 | $t_{35}$ | 154 | $t_{11}$ | 230 |
| $t_{23}$ | 0 | $t_{14}$ | 173 | $t_{16}$ | 258 |
| $t_{33}$ | 0 | $t_{15}$ | 173 | $t_{31}$ | 258 |
| $t_2$ | 69 | $t_9$ | 173 | $t_{21}$ | 258 |
| $t_5$ | 69 | $t_{26}$ | 174 | $t_{12}$ | 281 |
| $t_{24}$ | 78 | $t_{29}$ | 174 | $t_{38}$ | 323 |
| $t_{13}$ | 78 | $t_{36}$ | 230 | $t_{32}$ | 328 |
| $t_{34}$ | 99 | $t_{30}$ | 230 | $t_{22}$ | 350 |
| $t_{25}$ | 99 | $t_{10}$ | 230 | | |
| $t_6$ | 154 | $t_{37}$ | 230 | | |

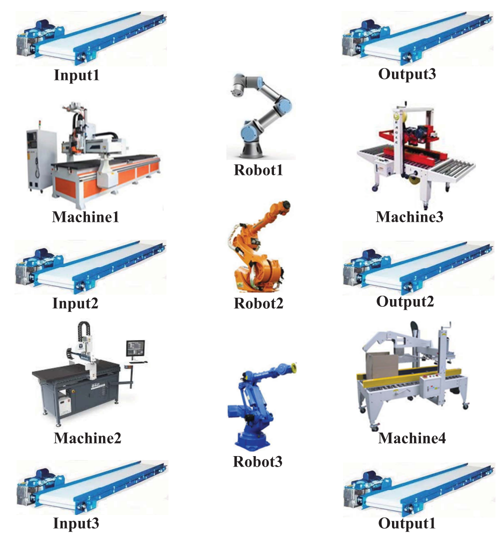


Fig. 3. RCM system [28], [29].

Next, we consider an RCM system adapted from [28] and [29] whose layout is shown in Fig. 3. The system has three robots $R_1$–$R_3$, four kinds of machines $M_1$–$M_4$ each of which has two units of machines, three loading buffers $I_1$–$I_3$, and three unloading buffers $O_1$–$O_3$. Three types of jobs or parts, $J_1$–$J_3$, are processed with machines and robots. The production sequences of the system are

$$J_1 : I_1 \rightarrow R_1(3) \rightarrow M_1(2) \rightarrow R_2(6) \rightarrow M_2(2) \rightarrow R_3(4) \rightarrow O_1$$
$$\text{or } I_1 \rightarrow R_1(3) \rightarrow M_3(1) \rightarrow R_2(3) \rightarrow M_4(4) \rightarrow R_3(4) \rightarrow O_1$$
$$J_2 : I_2 \rightarrow R_2(2) \rightarrow M_2(4) \rightarrow R_2(5) \rightarrow O_2$$
$$J_3 : I_3 \rightarrow R_3(6) \rightarrow M_4(3) \rightarrow R_2(4) \rightarrow M_3(6) \rightarrow R_1(2) \rightarrow O_3.$$

The operation time of each activity is given in parentheses. Suppose that each type of part has three units to be processed in the system. The place-timed PN of the system is shown in Fig. 4 where each idle place of the original net in [28], [29] has been split into a start place and an end place to distinguish the final state from other reachable states. The net has 20 transitions and 29 places in which $P_S = \{p_1, p_5, p_{14}\}$,

TABLE III
SCHEDULING RESULTS FOR EXAMPLE 2

| $p_1$ | $p_5$ | $p_{14}$ | A* search with $h = 0$ | | | A* search with $h = h_H$ | | |
|---|---|---|---|---|---|---|---|---|
| | | | Makespan | $N_E$ | $\mathcal{T}$ | Makespan | $N_E$ | $\mathcal{T}$ |
| 1 | 1 | 1 | 21 | 1347 | 5.01s | 21 | 517 | 2.16s |
| 2 | 2 | 2 | - | - | - | 30 | 2928 | 23.40s |
| 3 | 3 | 3 | - | - | - | 43 | 34112 | 648.91s |
| 4 | 4 | 4 | - | - | - | 57 | 65245 | 2119.48s |

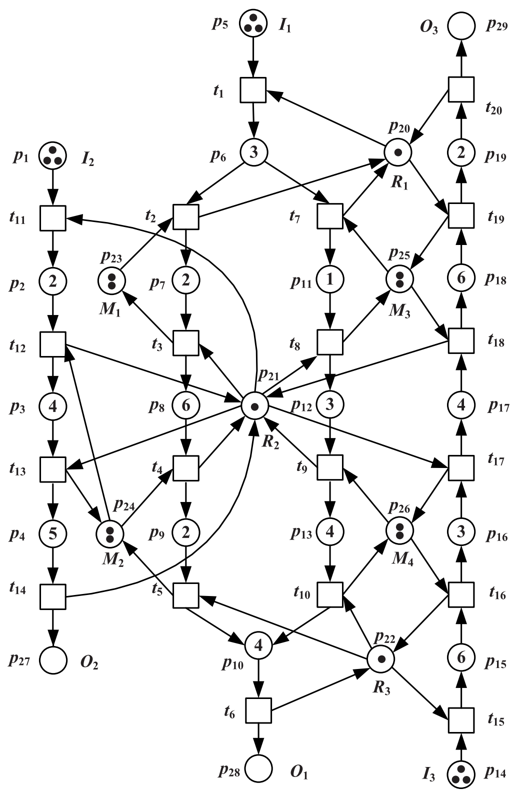


Fig. 4. Place-timed PN of the system in Fig. 3.

$P_E = \{p_{27}, p_{28}, p_{29}\}$, and $P_R = \{p_{20} - p_{26}\}$, and the remaining places are activity places $P_A$.

Four cases with different lot sizes are tested by the proposed method with $h = 0$ and $h = h_H$. The makespan of the obtained schedule, number of expanded states ($N_E$), and computational time ($\mathcal{T}$) are shown in Table III. From the results, we can see that the proposed method can deal with the PNs of RCM systems. The proposed heuristic function $h_H$, which is designed for the PNs of RCM systems, is highly informed and greatly improves search efficiency when compared with a noninformed function. For example, the search with $h = 0$ cannot even solve the case with $p_1 = p_5 = p_{14} = 2$ in an acceptable period of time, whereas the search with $h_H$ can optimally solve a bigger case $p_1 = p_5 = p_{14} = 4$ in about 35 min. Moreover, since $h_H$ is admissible, the obtained schedules are guaranteed to be optimal.

## VI. CONCLUSION

This article uses the timed PNs to naturally model the systems' structural properties and then utilizes the A* search with an admissible heuristic function to efficiently find optimal schedules for the underlying systems. When compared with the existing methods, the proposed approach can deal with token remaining time, weighted arcs, and multiple resource copies, which are common in the timed PNs of RCM systems. The admissibility of the proposed heuristic function is proved, which guarantees the optimality of the obtained schedules. This method can be used to schedule not only RCM systems but also conjunctive/disjunctive resource allocation systems (RASs) [12], which is a relatively general type of RASs allowing flexible routes and multiple resource acquisitions, e.g., automated manufacturing systems, intelligent transportation systems, and multithread software systems.

Future research can be extended in several directions. All A* search-based scheduling approaches have the state explosion problem, i.e., the number of reachable nodes increases exponentially with the system size. Thus, how to alleviate the problem and deal with some larger models with weighted arcs and multiple resource copies is an important issue. In addition, most of existing studies assume that resources do not fail. In fact, resource failures are inevitable in real systems [30]–[32]. Thus, developing a robust and efficient scheduling policy for the models with weighted arcs, multiple resource copies, and unreliable resources is critical.

## REFERENCES

[1] M. P. Fanti, G. Stecco, and W. Ukovich, "Scheduling internal operations in post-distribution cross docking systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 1, pp. 296–312, Jan. 2016.

[2] R. E. Burkard and J. Hatzl, "Review, extensions and computational comparison of MILP formulations for scheduling of batch processes," *Comput. Chem. Eng.*, vol. 29, no. 8, pp. 1752–1769, Jul. 2005.

[3] C. Pan, M. Zhou, Y. Qiao, and N. Wu, "Scheduling cluster tools in semiconductor manufacturing: Recent advances and challenges," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 2, pp. 586–601, Apr. 2018.

[4] J. Wang, H. Hu, C. Pan, Y. Zhou, and L. Li, "Scheduling dual-arm cluster tools with multiple wafer types and residency time constraints," *IEEE/CAA J. Automatica Sinica*, vol. 7, no. 3, pp. 776–789, May 2020.

[5] Q. Zhu, Y. Qiao, N. Wu, and Y. Hou, "Post-processing time-aware optimal scheduling of single robotic cluster tools," *IEEE/CAA J. Automatica Sinica*, vol. 7, no. 2, pp. 597–605, Mar. 2020.

[6] Y. Qiao, N. Wu, F. Yang, M. Zhou, Q. Zhu, and T. Qu, "Robust scheduling of time-constrained dual-arm cluster tools with wafer revisiting and activity time disturbance," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 6, pp. 1228–1240, Jun. 2019.

[7] F. Yang, N. Wu, Y. Qiao, M. Zhou, R. Su, and T. Qu, "Modeling and optimal cyclic scheduling of time-constrained single-robot-arm cluster tools via Petri nets and linear programming," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 3, pp. 871–883, Mar. 2020.

[8] K. Gao, Z. Cao, L. Zhang, Z. Chen, Y. Han, and Q. Pan, "A review on swarm intelligence and evolutionary algorithms for solving flexible job shop scheduling problems," *IEEE/CAA J. Automatica Sinica*, vol. 6, no. 4, pp. 904–916, Jul. 2019.

[9] Z. Cao, C. Lin, M. Zhou, and R. Huang, "Scheduling semiconductor testing facility by using cuckoo search algorithm with reinforcement learning and surrogate modeling," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 2, pp. 825–837, Apr. 2019.

[10] H. Yuan, J. Bi, W. Tan, M. Zhou, B. H. Li, and J. Li, "TTSA: An effective scheduling approach for delay bounded tasks in hybrid clouds," *IEEE Trans. Cybern.*, vol. 47, no. 11, pp. 3658–3668, Nov. 2017.

[11] X. Zuo *et al.*, "Optimizing hospital emergency department layout via multiobjective Tabu search," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 3, pp. 1137–1147, Jul. 2019.

[12] B. Huang and M. Zhou, *Supervisory Control and Scheduling of Resource Allocation Systems: Reachability Graph Perspective*. Hoboken, NJ, USA: Wiley, 2020.

[13] D. Yong Lee and F. DiCesare, "Scheduling flexible manufacturing systems using Petri nets and heuristic search," *IEEE Trans. Robot. Autom.*, vol. 10, no. 2, pp. 123–132, Apr. 1994.

[14] A. Reyes, H. Yu, G. Kelleher, and S. Lloyd, "Integrating Petri nets and hybrid heuristic search for the scheduling of FMS," *Comput. Ind.*, vol. 47, no. 1, pp. 123–138, Jan. 2002.

[15] J. Lee and J. S. Lee, "Heuristic search for scheduling flexible manufacturing systems using lower bound reachability matrix," *Comput. Ind. Eng.*, vol. 59, no. 4, pp. 799–806, Nov. 2010.

[16] H. Henry Xiong and M. Zhou, "Scheduling of semiconductor test facility via Petri nets and hybrid heuristic search," *IEEE Trans. Semicond. Manuf.*, vol. 11, no. 3, pp. 384–393, Aug. 1998.

[17] B. Huang, Y. Sun, and Y. M. Sun, "Scheduling of flexible manufacturing systems based on Petri nets and hybrid heuristic search," *Int. J. Prod. Res.*, vol. 46, no. 16, pp. 4553–4565, Aug. 2008.

[18] B. Huang, Y. Sun, Y.-M. Sun, and C.-X. Zhao, "A hybrid heuristic search algorithm for scheduling FMS based on Petri net model," *Int. J. Adv. Manuf. Technol.*, vol. 48, nos. 9–12, pp. 925–933, Jun. 2010.

[19] O. T. Baruwa and M. A. Piera, "Identifying FMS repetitive patterns for efficient search-based scheduling algorithm: A colored Petri net approach," *J. Manuf. Syst.*, vol. 35, pp. 120–135, Apr. 2015.

[20] B. Huang, X.-X. Shi, and N. Xu, "Scheduling FMS with alternative routings using Petri nets and near admissible heuristic search," *Int. J. Adv. Manuf. Technol.*, vol. 63, nos. 9–12, pp. 1131–1136, Dec. 2012.

[21] B. Huang, R. Jiang, and G. Zhang, "Search strategy for scheduling flexible manufacturing systems simultaneously using admissible heuristic functions and nonadmissible heuristic functions," *Comput. Ind. Eng.*, vol. 71, pp. 21–26, May 2014.

[22] J. Luo, K. Xing, M. Zhou, X. Li, and X. Wang, "Deadlock-free scheduling of automated manufacturing systems using Petri nets and hybrid heuristic search," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 3, pp. 530–541, Mar. 2015.

[23] G. Mejía, J. P. Caballero-Villalobos, and C. Montoya, "Petri nets and deadlock-free scheduling of open shop manufacturing systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 6, pp. 1017–1028, Jun. 2018.

[24] Z. Li, N. Wu, and M. Zhou, "Deadlock control of automated manufacturing systems based on Petri nets—A literature review," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 4, pp. 437–462, Jul. 2012.

[25] N. Wu and M. Zhou, "Schedulability analysis and optimal scheduling of dual-arm cluster tools with residency time constraint and activity time variation," *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 1, pp. 203–209, Jan. 2012.

[26] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2010.

[27] B. Huang. (2020). *Source Codes and Experimental Files*. Accessed: Mar. 2020. [Online]. Available: http://github.com/huang-njust/RCM_Scheduling

[28] Y. Chen and Z. Li, "Design of a maximally permissive liveness-enforcing supervisor with a compressed supervisory structure for flexible manufacturing systems," *Automatica*, vol. 47, no. 5, pp. 1028–1034, May 2011.

[29] B. Huang, M. Zhou, G. Zhang, A. C. Ammari, A. Alabdulwahab, and A. G. Fayoumi, "Lexicographic multiobjective integer programming for optimal and structurally minimal Petri net supervisors of automated manufacturing systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 11, pp. 1459–1470, Nov. 2015.

[30] N. Du, H. Hu, and M. Zhou, "A survey on robust deadlock control policies for automated manufacturing systems with unreliable resources," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 1, pp. 389–406, Jan. 2020.

[31] J. Luo *et al.*, "Robust deadlock control of automated manufacturing systems with multiple unreliable resources," *Inf. Sci.*, vol. 479, pp. 401–415, Apr. 2019.

[32] P. Zhang, Y. Kong, and M. Zhou, "A domain partition-based trust model for unreliable clouds," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 9, pp. 2167–2178, Sep. 2018.

**Bo Huang** (Senior Member, IEEE) received the B.S. and Ph.D. degrees from the Nanjing University of Science and Technology (NUST), Nanjing, China, in 2002 and 2006, respectively.

In 2007, he joined NUST, where he is currently a Full Professor with the School of Computer Science and Engineering. He was a Visiting Scholar with the University of Missouri–Kansas City in 2013, and the New Jersey Institute of Technology from 2014 to 2015 and from 2019 to 2020. His research interests include discrete-event systems, Petri nets, intelligent automation, intelligent transportation, and multirobot systems. He has taken charge of more than ten projects and has published one book and more than 60 articles in his areas.

**MengChu Zhou** (Fellow, IEEE) received the B.S. degree in control engineering from the Nanjing University of Science and Technology, Nanjing, China, in 1983, the M.S. degree in automatic control from the Beijing Institute of Technology, Beijing, China, in 1986, and the Ph.D. degree in computer and systems engineering from the Rensselaer Polytechnic Institute, Troy, NY, USA, in 1990.

In 1990, he joined the New Jersey Institute of Technology (NJIT), Newark, NJ, USA, where he is currently a Distinguished Professor of electrical and computer engineering. He has over 900 publications, including 12 books, over 600 journal articles (over 450 in IEEE TRANSACTIONS), 26 patents, and 29 book chapters. His research interests are in Petri nets, intelligent automation, Internet of Things, big data, web services, and intelligent transportation.

Dr. Zhou is a fellow of the International Federation of Automatic Control (IFAC), the American Association for the Advancement of Science (AAAS), and the Chinese Association of Automation (CAA). He was a recipient of the Humboldt Research Award for U.S. Senior Scientists from the Alexander von Humboldt Foundation, the Franklin V. Taylor Memorial Award and the Norbert Wiener Award from the IEEE Systems, Man and Cybernetics Society, and the Excellence in Research Prize and Medal from NJIT. In 1999, he was the President of the Chinese Association for Science and Technology-USA, where he is a Life Member. He is also the Founding Editor of IEEE Press Book Series on Systems Science and Engineering and the Editor-in-Chief of the IEEE/CAA JOURNAL OF AUTOMATICA SINICA.

**Abdullah Abusorrah** (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the University of Nottingham, Nottingham, U.K., in 2007.

He is currently a Professor with the Department of Electrical and Computer Engineering, King Abdulaziz University, Jeddah, Saudi Arabia, where he is also the Head of the Center for Renewable Energy and Power Systems. His research interests include energy systems, smart grid, and system analyses.

**Khaled Sedraoui** received the B.S. degree from the Institute of Technology and Sciences Tunis (ESSTT), Tunis, Tunisia, in 1989, the M.Sc. degree from the École de Technologie Supérieure, Quebec University, Montreal, QC, Canada, in 1994, and the Ph.D. degree from ESSTT in 2010, all in electrical engineering.

Since 1997, he has been a Faculty Member with the College of Technology, Jeddah, Saudi Arabia. He is currently an Associate Professor of electrical and computer engineering with the College of Engineering, King Abdulaziz University, Jeddah. His research interests include renewable energies integration and power quality improvement, electric energy storage, and optimization for smart systems.