IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 4 September 2023, accepted 2 October 2023, date of publication 5 October 2023, date of current version 11 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3322141

RESEARCH ARTICLE

Sequence Planning for Labeled Petri Nets With Time and Resource Constraints Using Basis Markings

YEJIA LIU [1], (Graduate Student Member, IEEE), XUNBO LI [1], AND AHMED M. EL-SHERBEENY [2]

[1]School of Mechanical and Electrical Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
[2]Industrial Engineering Department, College of Engineering, King Saud University, Riyadh 11421, Saudi Arabia

Corresponding author: Xunbo Li (xbli@uestc.edu.cn)

This work was supported by the Researchers Supporting Project of King Saud University under Grant RSP2023R133.

ABSTRACT This paper addresses the scheduling problem for discrete event systems modeled by labeled Petri nets with time and resource constraints where deadlocks are inevitable. For better resource utilization and shorter processing time, a heuristic algorithm is presented for designing a suitable transition sequence that starts from the initial marking to a set of target markings using basis markings. Specially, two reasons exist for deadlocks in the given system. One is resource exhaustion and the other is unreasonable resource allocation. We only focus on the former. First, the set of target markings, i.e., deadlocks caused by resource exhaustion, is identified using the notion of basis markings and resource-exhausted markings. The basis reachability graph instead of the conventional reachability graph is constructed to avoid state space explosion. An integer linear programming problem based on the notion of deadlocks is carried out to distinguish basis markings, where deadlocks can be reached by firing unobservable transitions only. Then, the A-star algorithm is applied to the basis marking space to schedule the transition firing sequences and the optimal results may be obtained. Unpromising searching areas are reduced and only a part of the markings is probed. Finally, a numerical case is studied to verify the effectiveness of the proposed algorithm. The main advantages of the proposed approach include that the exhaustive enumeration of the reachability space can be avoided and the calculation can be completed off-line.

INDEX TERMS Basis marking, discrete event system, labeled Petri net, resource consumption, sequence planning.

I. INTRODUCTION

Flexible manufacturing systems (FMSs) for rapid production of small and medium-sized orders have emerged in response to the current flexible and changing market needs. FMSs are mainly composed of computerized numerical control (CNC) machines and material conveying systems. It is a type of automatic manufacturing system that can adapt to the transformation of processing objects and processes. FMSs take the advantages of high utilization rate of production equipment, stable processing capacity, good product quality and easy operation of production line. Usually, it is necessary to apply operation commands through the system computer before the production process starts in order to successfully complete the corresponding orders. Without supervision and control, production lines are possible to reach deadlocks due to unreasonable resource allocation, which not only results in poor resource utilization but also leads to production standstill or even more severe production accidents.

The state evolution of an FMS is driven by events as all other discrete event systems (DESs) do. A common research method for DESs is the Petri net theory. Petri nets are widely used in DES modeling due to their rigorous mathematical expression and intuitive graphical representation [1], [2]. Numerous modeling approaches using Petri nets have been proposed for the different characteristics

The associate editor coordinating the review of this manuscript and approving it for publication was Cong Pu.

of FMSs. Zhou and Dicesare [3] introduced the modeling method for FMSs with shared resources and proposed the notions of parallel mutual exclusion (PME) and sequential mutual exclusion (SME). Boundedness, liveness, and reversibility were discussed in such models. Ezpeleta et al. [4] proposed the system of simple sequential process with resources (S^3PR) to model the concurrent execution of working processes in FMSs. Since the performance analysis of FMSs is highly correlated with time, Ramchandani [5] appended invariant constants to places or transitions to show fixed time consumption. Furthermore, stochastic Petri nets (SPNs) are proposed for non-deterministic time modeling in DESs [6]. The Petri net model expands dramatically as the system grows in size. For this reason, Jenson [7] proposed a colored Petri net (CPN) to compress large systems. Based on these numerous models, supervisory control problems [8], [9], [10], [11], scheduling problems [12], [13], [14], [15], [16], [17], and fault diagnosis problems [18], [19], [20], [21] are further investigated.

The shut-down of all equipment in DESs is presented as a dead marking in Petri nets. That is, tokens in the system are no longer flowing due to the disablement of transitions. To deal with the deadlocks that may or have already occurred in the system, treatments for deadlocks have been proposed by researchers based on the supervisory control theory originated from the works in [22], [23], and [24]. Both deadlock avoidance and deadlock prevention approaches are proposed to avoid the occurrence of deadlocks in advance [25], [26], [27], [28], [29].

The aforementioned works assume that tokens are conservative while the system is evolving. The occurrences of deadlocks are only caused by the improper allocation of resources during the operation of the system. However, in practical manufacturing processes, the consumption of resources is ineluctable. The resources in an FMS can be divided into durable resources (machines, equipment, etc.) and consumable resources (worn tools, etc.). The number of consumable resources decreases during the production process until they are exhausted, resulting in stagnation, even deadlocks in the system. In addition, since not all parts of the production line are equipped with sensors, the occurrence of some events cannot be sensed. Cabasino et al. [30] proposed the labeled Petri net (LPN) to describe Petri nets with unobservable transitions. In this paper, we model a resource-consuming DES by labeled Petri nets for sequence planning.

In the perspective of the DES, scheduling, i.e., determining the ideal sequence of events in a system, is not easy due to the existence of multiple paths, sharing and conflict of resources, sequential and parallel relationships between operations. With the help of the powerful modeling capabilities of Petri nets, we address the scheduling problem in resource-consuming systems. Scheduling in Petri nets is to determine the optimal transition sequence from the initial marking to the target marking. Scheduling methods are mainly classified into the mathematical programming [31], [32], [33], [34], artificial intelligence methods [35], [36], [37], and heuristic methods [38], [39], [40], [41], [42].

The work in [31] proposed an integer linear programming (ILP) method to solve a firing vector of an optimal sequence. Furthermore, an on-line approach was introduced in [32]. The work in [33] proposed a modified ILP in a live state machine to reduce computational overhead. He et al. [34] combined the structural characteristics and the ILP technique for path planning in timed Petri nets (TPN). As for artificial intelligence methods, the genetic algorithms were used to find suboptimal sequences in DESs [35]. The ant colony algorithms were applied for path optimization in [36] and solutions were quickly and effectively obtained. Dai et al. [37] proposed a neural network algorithm for human resource allocation in a Petri net model.

Heuristic methods such as Dijkstra searching algorithm [38] and Floyd-Warshall algorithm [39] were widely used for sequence planning. The works in [40] and [41] used the Dijkstra algorithm based on the reachable marking space. As the size of the system enlarges, state space explosion may happen. In this work, we use a heuristic method called the A-star algorithm [42] in the basis reachability graph to avoid state space explosion and improve searching efficiency.

Under the condition that resources are limited and continuously consumed, the system constantly tends to deadlocks. Based on such characteristics, we set the scheduling goal on the improvement of resource utilization. That is, we try to discover an ideal firing sequence that enforces the system to reach a marking that resources are exhausted and the system cannot continue to evolve. In addition, in order to improve productivity and reduce production time, we also consider time constraints in the system and try to schedule the sequence with high resource utilization and short production time.

In this work, we focus on the scheduling problem for labeled Petri nets with time and resource constraints. The major contributions are as follows:

- The resource-consuming DES is modeled by the labeled Petri net with time and resource constraints. The weights of arcs related to resource places are restricted and timed information is appended to activity places. Then, resource exhaustion on places in such systems is defined and calculated. Furthermore, the notion of resource-exhausted marking is given, which is regarded as a target marking in scheduling.
- The basis marking space is constructed and an integer linear programming problem for deadlock identification is addressed in the refined space to avoid state space explosion and reduce computational overhead.
- The A-star algorithm is proposed to schedule the transition sequences from the initial state to the resource-exhausted dead markings for a given system for high resource utilization and short processing time. The traditional searching space based on the reachability set is refined to the set of basis markings and unpromising search sub-space is discarded. The result is

proven to be optimal or sub-optimal due to different cost functions.

The remainder of the paper is organized in the following way. Section II reminds the readers of the preliminaries of Petri nets. In Section III, we give a few assumptions and formalize the problem to be dealt with in this work. Then, the methods for distinguishing resource-exhausted dead markings are introduced in Section IV. Section V proposes a scheduling algorithm for a Petri net system. A numerical example is calculated in Section VI. Finally, conclusions and future work are given in Section VII.

II. PRELIMINARIES

In this section, the formalism and preliminary results related to the study are recalled. For more about Petri nets, the readers are refered to [43] and [44].

A. PETRI NETS

A Petri net is a four-tuple $N = (P, T, Pre, Post)$, where P denotes a set of m places, and T is a set of n transitions. Places and transitions are depicted as circles and bars in graphs, respectively. Pre: $P \times T \to \mathbb{N}$ and $Post$: $P \times T \to \mathbb{N}$ are the pre- and post-incidence matrices that specify the directed arcs between places and transitions, respectively. The incidence matrix of N is denoted by $C = Post\text{-}Pre$. We denote the set of non-negative integers by $\mathbb{N}$.

Given a node $x \in P \cup T$, $\cdot x$ and $x\cdot$ denote the input and output sets of x, respectively. A path $x_1 \ldots x_r$ of a net exists if $x_i \in x_{i-1}\cdot$, $x_i \in P \cup T$ holds for $i = 1, \ldots, r$, $r \in \mathbb{N}$. If $x_1 = x_r$ and all other nodes are diverse, a circuit is constructed. A Petri net is acyclic if no circuit exists. We define $z = |Z|$ as the cardinality of the set Z.

M: $P \to \mathbb{N}$ is defined as a marking of N that assigns a non-negative integer number of tokens, represented graphically by black dots, to each place. $M(p)$ indicates the number of tokens in a place $p \in P$ at a marking M. A marking M can also be presented by $M = \Sigma_{p \in P} M(p) \cdot p$. The Petri net N and an initial marking M_0 forms a Petri net system $\langle N, M_0 \rangle$.

A transition t is enabled at M if for all $p \in \cdot t_i$, $M(p) \geq Pre(p, t_i)$, which is denoted by $M[t_i\rangle$. A marking M' can be calculated by $M' = M + C(\cdot, t_i)$ if it is generated by firing an enabled transition t_i at M. Given a sequence of transitions $\sigma \in T^*$ and a marking M, $M[\sigma\rangle M'$ denotes that σ is enabled at M and M' is reachable from M by firing σ. The state equation $M = M_0 + C \cdot y(\sigma)$ shows that a new marking M yields by firing a transition sequence σ at the initial marking M_0, where the firing vector $y(\sigma)$: $T \to \mathbb{N}$ is a non-negative integer vector showing the number of firings of σ. For the transition sequence σ, $y(t) = k$ indicates that transition t fires k times in σ.

All markings reachable from the M_0 compose a reachability set of the Petri net, denoted by $R(N, M_0)$. The reachability graph (RG) is a deterministic graph that has as many nodes as the number of markings in the $R(N, M_0)$. An arc from M to M' in the RG is associated with a transition t such that $M[t\rangle\, M'$.

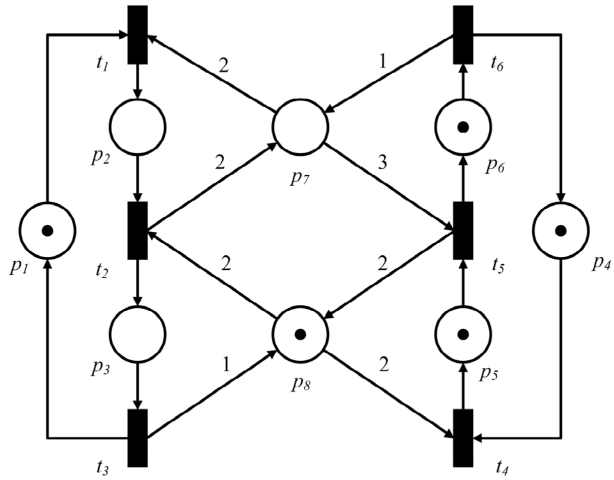

FIGURE 1. An GS^3 PR with two resource places.

A Petri net is bounded if there exists an integer $k \in \mathbb{N}$ such that for all $p \in P$, for all $M \in R(N, M_0)$, it holds $M(p) \leq k$. A system is dead if there does not exist $t \in T$ such that $M_0[t\rangle$. If for all $M \in R(N, M_0)$, there exists $t \in T$ such that $M[t\rangle$, the net system $\langle N, M_0 \rangle$ is deadlock-free.

For better modeling manufacturing systems by Petri nets, the generalized simple sequential process with resource (GS^2PR) is proposed, denoted by $N_R = (P_A \cup \{p^0\} \cup P_R, T, Pre, Post, W)$, where P_A is the set of activity places, p^0 is an idle place, P_R is the set of resource places, and W is the set of weights of arcs. Here $W = W_A \cup W_R$, where W_A: $((P_A \cup \{p^0\}) \times T) \cup (T \times (P_A \cup \{p^0\})) \to \{0,1\}$ and W_R: $(P_R \times T) \cup (T \times P_R) \to \mathbb{N}$. Note that N_R is a strongly connected state machine and every circuit in N_R contains p^0. Places in P_A represent operations, and places in p^0 and P_R represent the resource status. Transitions in N_R represent the start or end of an operation. A system of GS^2PRs is called a GS^3PR.

B. LABELED PETRI NETS WITH TIME AND RESOURCE CONSTRAINTS

Given a Petri net $N = (P, T, Pre, Post)$, the numbers of input and output transitions of a place $p \in P$ are denoted by a and b, respectively. We define the sum of the weight of arcs between the place p and its input and output transitions in (1) and (2), respectively.

$$Sum_i(p) = \sum_{i=1}^{a} W(t_i, p), t_i \in \cdot p \tag{1}$$

$$Sum_o(p) = \sum_{j=1}^{b} W(p, t_i), t_j \in p\cdot \tag{2}$$

For each place $p \in P$, if $Sum_i < Sum_o$, p is a token-consuming place. Given a Petri net N, if there exists at least one place $p \in P$ that is a token-consuming place, the Petri net N is defined as a Petri net with token-consumption. For any place in P_R of an GS^3PR N_R, if it is a token-consuming place, the GS^3PR is with resource constraints.

An event is observable in a DES if a sensor is deployed for it. It is normal that not all the events are equipped with sensors due to technical or economic considerations. We consider the

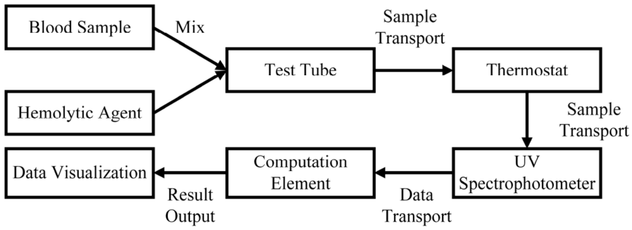

FIGURE 2. The schematic process of Hb determination.

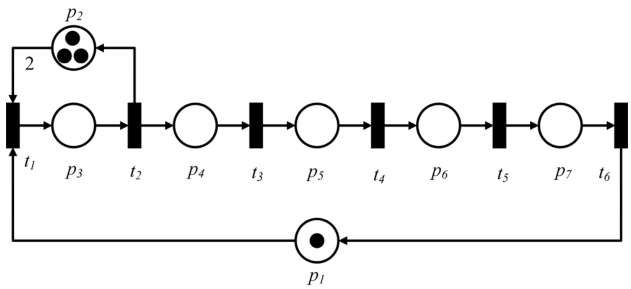

FIGURE 3. The Petri net model of Fig. 2.

existence of unobservable events in a DES to generalize the addressed problem. A labeled Petri net is denoted by a four-tuple $G = (N, M_0, E, \ell)$, where $\langle N, M_0\rangle$ is a Petri net system, E is the set of labels, and ℓ: $T \to E \cup \{\varepsilon\}$ is the labeling function that assigns to each transition $t \in T$ either a symbol from E or an empty word ε. Each label in E can be observed through sensors. Therefore, the transition set is classified into two disjoint sets $T = T_o \cup T_u$, where $T_o = \{t \in T | \ell(t) \in E\}$ contains observable transitions and unobservable transitions are included in $T_u = \{t \in T | \ell(t) \in \varepsilon\}$. No observation, i.e., a sequence of labels denoted by w, is generated if only transitions in T_u fire.

If an unobservable sequence $\sigma_u \in T_u^*$ is given, we define an n_u-dimensional vector y_u: $T_u \to \mathbb{N}$ as the unobservable firing vector and $y_u(t) = k$ if $t \in T_u$ is contained k times in y_u. An observable firing vector is defined as y_o: $T_o \to \mathbb{N}$ for $\sigma_o \in T_o^*$ in an analogous manner.

The transition-marking sequence is defined as an evolution of G from M and we have

$$\lambda(M) = Mt_{\alpha 1}M_1t_{\alpha 2}M_2 \ldots t_{\alpha L}M_{\mathrm{L}}$$

if

$$M[t_{\alpha 1}\rangle M_1[t_{\alpha 2}\rangle M_2 \ldots [t_{\alpha L}\rangle M_{\mathrm{L}}$$

where $\alpha_{\mathrm{i}} \in \{1,2,\ldots, n\}$, $t_{\alpha i} \in T$, $M_{\mathrm{i}} \in R(N, M_0)$, $i = 1,\ldots, L$, $L \in \{1,2,\ldots\}$. The transition sequence $t_{\alpha 1}\ t_{\alpha 2}\ldots t_{\alpha L}$ in $\lambda(M)$ is presented by $\sigma[\lambda(M)]$. All possible evolutions from M composes the set $\Lambda(N, M)$.

If all observable transitions are removed from N, the T_u-induced subnet of G is derived and this unobservable subnet is denoted by $N_{\mathrm{u}} = (P, T_{\mathrm{u}}, Pre_u, Post_u)$, where Pre_u and $Post_u$ are the restrictions of Pre and $Post$ to T_u, respectively. To avoid permanent lack of observation, we assume that the T_u-induced subnet is acyclic.

The LPN with time and resource constraints is a five-tuple $G_R = (N_R, M_0, E, \ell, \Gamma)$, where N_R is a GS^{2}PR, Γ: $P_A \to \mathbb{N}$ is a timed vector assigning a non-negative integer unit time to each place in P_A modeling the processing time of each operation, and token-consuming places exist in P_R.

C. TOKEN-EXHAUSTED MARKINGS

Resources are consumed as the system evolves in G_R. The definition of the weighted sum of the residual resources for resource places at current marking is given to figure out whether the place is token-exhausted or not. Given a net N_R, we denote the weighted sum of the residual resources for a resource place $p_r \in P_R$ at the current marking M by $S_M(p_r)$. Let $L(p_r)$ be the set of holders of p_r, i.e., the set of activity places that use tokens from p_r. Moreover, let $l(p_r)$ be a subset of $L(p_r)$ that all places in $l(p_r)$, together with their relevant transitions and p_r compose a circuit. The sum of tokens that belong to each $l(p_r)$ is denoted by $sum(l)$. Equation (3) calculates $S_M(p_r)$ and p_r is resource-exhausted if (4) is satisfied.

$$S_M(p_r) = \sum\nolimits_{l(p_r)\subseteq L(p_r)} sum(l)\cdot W(\cdot p_r, p_r) + M(p_r) \quad (3)$$

$$S_M(p_r) < \min\{W(p_r,\ p_r\cdot)\} \quad (4)$$

If $S_M(p_r)$ is less than $\min\{W(p_r,\ p_r\cdot)\}$, i.e., (4) is true, p_r is a resource-exhausted place. A marking M is regarded as a resource-exhausted marking if there exists at least one place which is resource-exhausted at M. We only focus on dead markings that are resource-exhausted. A Petri net is ineluctably to be dead since resources are limited. For the purpose of making more use of resources, we attempt to find the sequences from the initial state to states that are resource-exhausted.

Example 1: Given an GS^{3}PR where the current marking is $M = [1{,}0{,}0{,}1{,}1{,}1{,}0{,}1]^{\mathrm{T}}$, the sets of operation, idle, and resource places are $P_A = \{p_2, p_3, p_5, p_6\}$, $\{p^0\} = \{p_1, p_4\}$, and $P_R = \{p_7, p_8\}$, respectively. The structure of the net is displayed in Fig. 1, where the weights of arcs related to resource places p_7 and p_8 are marked by integer numbers.

According to (1) and (2), we have:

$$Sum_i(p_7) = 3 < Sum_o(p_7) = 5$$
$$Sum_i(p_8) = 3 < Sum_o(p_8) = 4$$

Therefore, both p_7 and p_8 are resource-consuming places. For each resource place, we have $L(p_7) = \{p_2,\ p_6\}$ and $L(p_8) = \{p_3,\ p_5\}$. The place p_7 is contained in two basic circuits, which are $p_7t_1p_2t_2p_7$ and $p_7t_5p_6t_6p_7$.

Hence, $L(p_7)$ is divided into two disjoint subsets where $l_1(p_7) = \{p_2\}$ and $l_2(p_7) = \{p_6\}$. Similarly, $p_8t_2p_3t_3p_8$ and $p_8t_4p_5t_5p_8$ are two circuits for p_8. According to (3), we have:

$$S_M(p_7) = M(p_2) \times W(t_2, p_7) + M(p_6) \times W(t_6, p_7) + \mathrm{M}(p_7) = 1$$

$$S_M(p_8) = M(p_3) \times W(t_3, p_8) + M(p_5) \times W(t_5, p_8) + \mathrm{M}(p_8) = 3$$

The usage of resources is examined by (4) and we have $S_M(p_7) < \min\{W(p_7, p_7\cdot)\}=2$ and $S_M(p_8) > \min\{W(p_8, p_8\cdot)\}=2$. Therefore, p_7 is the only resource-exhausted place and the current marking M is resource-exhausted.

III. PROBLEM STATEMENT

A. MOTIVATION

Determining target markings is quite important for scheduling problems. In the existing works, deadlocks are usually caused by unreasonable resource allocation that are definitely undesired. If a system enters such deadlocks, the system is difficult to recover and such deadlocks should be avoided. In this paper, the target deadlocks we choose are only caused by resource exhaustion, not by unreasonable resource allocation.

For a system with resource consumption, it inevitably reaches deadlocks due to its inherent token consumption characteristics. If the system gets stuck due to resource shortage, it can be easily recovered by appending resources. That is to say, if the number of tokens in the resource places are enough initially, the system will be not in deadlocks by applying a reasonable scheduling plan. Or we can say that the proposed scheduling plan is able to prevent the system from entering deadlocks caused by unreasonable resource allocation and try to use the resources as many as possible. Based on these advantages, we choose the resource-exhausted deadlock as a target marking while scheduling.

We propose a practical resource-consuming system and show the resource-exhausted dead status of the system. Consider an automated medical analysis system (AMAS) as an example. In general, an AMAS is a system that connects multiple medical equipment for complete medical analysis, which is very important for health care systems. It can accurately complete a series of operations such as sample extraction, centrifugation, cup separation, reagent addition, mixing, reaction, and result processing.

We take the process of hemoglobin (Hb) determination in an AMAS as an example in detail. First, add a certain amount of the blood sample (p_1) and the hemolytic agent (p_2) to the test tube (p_3). The mixed sample is incubated in a thermostat (p_4). Then, the mixed sample is transported to the ultraviolet (UV) Spectrophotometer (p_5) and it is studied by the Lambert-Beer's law. Finally, the data is calculated by the computation element (p_6) and the Hb concentration is outputted (p_7).

Note that in this system, the hemolytic agents are regarded as consumable resources. If they are exhausted, the system is not able to deal with the blood sample any further. In the mapping Petri net, a resource-exhausted deadlock exists. Fig. 2 show the process of Hb determination and Fig. 3 shows the mapping Petri net. The initial marking $M_0 = [1,3,0,0,0,0,0]^{\mathrm{T}}$ and t_1 is only able to fire twice because of the limitation of $M(p_2)$. If multiple paths exist in a system and the resources are limited, reasonable scheduling plans are needed for high resource utilization and efficient processing.

B. PROBLEM STATEMENT

In DESs such as the manufacturing systems, resource consumption is an inevitable phenomenon that occurs during processing. The consumption of raw materials, wear and tear of cutting tools, and the use of consumables all belong to resource consumption. The system halts when resources are exhausted. Mapping to the Petri net systems, the number of tokens in resource places decreases as the system runs, and deadlocks are ineluctably reached. Note that there are two possible reasons for deadlocks in the resource consuming systems: one is resource exhaustion and the other is unreasonable resource allocation. We only focus on the deadlocks that caused by resource exhaustion. In addition, time constraints for activity places are considered since the production takes time.

This paper tackles the sequence planning problem for DESs modeled by LPNs with time and resource constraints. The optimal or suboptimal sequence in such an LPN is the one that costs the least time and leads to a resource-exhausted dead marking. The given net G_R contains unobservable transitions and the basis marking calculation can be used to effectively reduce the state space. The premise of the sequence planning problem is to distinguish the resource-exhausted dead markings using the basis marking analysis. Then, a heuristic algorithm is proposed to schedule the firings of transitions for shorter time and higher resource utilization. In this paper, two assumptions are made:

1) The labeled Petri net is bounded.
2) The T_u-induced subnet is acyclic.

Assumption 1 ensures that the number of reachable basis markings of the net is finite and the calculation of the BRG can be terminated. If the T_u-induced subnet is cyclic, the system may evolve by firing unobservable transitions only and no label can be observed. Assumption 2 avoids such a situation. We formalize the scheduling problem in LPN as follows:

Problem: Given a labeled Petri net system with time and resource constraints G_R under Assumptions 1 and 2, the problem lies in scheduling the transition sequences that guarantee the shortest processing time and highest resource utilization for the system.

IV. RESOURCE-EXHAUSTED DEADLOCKS IDENTIFICATION

Deadlock prevention and avoidance problems are widely concerned with token-conservation Petri nets. However, for resource-consuming systems, due to the continuous consumption of tokens in resource places, the system will inevitably evolve to deadlocks. The existence of deadlocks stems from either resource exhaustion or unreasonable resource allocation. In this section, we present a method with the help of the basis reachability graph (BRG) to identify deadlocks caused by resource exhaustion in the given system. Under the premise of limited resources, making full use of resources is expected. The existence of the resource-exhausted deadlock provides the prerequisite for the algorithm to determine an ideal transition sequence from M_0 to the set of target markings.

A. BASIS REACHABILITY GRAPH

Assuming that the given LPN is bounded, infinite calculation of the BRG is avoided. Similar to the reachability graph, the BRG is a deterministic graph. All possible basis markings reachable from M_0 are included in the BRG. That is to say, the number of basis markings equals the node number [45]. Before we introduce the BRG construction algorithm, related notions are pre-defined.

Definition 1: Consider a marking $M \in R(N, M_0)$ and an observable transition $t \in T_o$. We define

$$\Sigma(M, t) = \{\sigma \in T_u * | \, M[\sigma\rangle M',\ M' \geq\ Pre(\cdot,\ t)\}$$

as a set of explanations of t at M. The firing vectors corresponded to σ are called e-vectors and the set of e-vectors is denoted by $Y(M, t) = \pi(\Sigma(M, t))$. All sequences in $\Sigma(M, t)$ are unobservable and they can fire at M reaching a state that enables t. We only concern about the minimal explanations since they serve as the crucial part of the BRG construction.

Definition 2: Given a marking M and a transition $t \in T_o$, we define

$$\Sigma_{min}(M, t) = \{\sigma \in \Sigma(M, t) | \nexists \sigma' \in \Sigma(M, t) : \pi(\sigma') < \pi(\sigma)\}$$

as the set of minimal explanations of t at M and define

$$Y_{min}(M,\ t) =\ \pi(\Sigma_{min}(M,\ t))$$

as the set of minimal e-vectors. We assume that the unobservable subnet is acyclic. Hence, the computation method for $Y_{min}(M, t)$ in [46] is applicable in this paper.

We define the set of basis markings based on the notions of explanations. All basis markings are derived from the initial marking M_0 by firing an observable sequence σ and the corresponding unobservable sequence that is strictly necessary to enable σ.

Definition 3: Given an LPN $G = (N, M_0, E, \ell)$, we define $\mathcal{M}_B$ as the set of basis markings of G such that:

1) $M_0 \in \mathcal{M}_B$.
2) $\forall M \in \mathcal{M}_B, \forall t \in T_o, \forall y_u \in Y_{min}(M, t)$, it holds $M' \in \mathcal{M}_B$, where $M' = M + C(\cdot,\mathrm{t}) + C_u \cdot y_u$.

Obviously, the initial marking M_0 is one of the basis markings. After firing an observable transition $t \in T_o$ together with its minimal explanation, a basis marking is generated. Markings that are reachable by firing unobservable transitions only are unconcerned. The size of $\mathcal{M}_B$ is much smaller than that of $R(N, M_0)$. More specifically, $\mathcal{M}_B$ is a subset of $R(N, M_0)$ and the BRG is a compact representation of the RG.

The construction approach of the BRG is shown in Algorithm 1 by pseudocode. $\mathcal{M}_B$ is updated iteratively until all markings are tagged. We denote the BRG as $\mathcal{B} = (\mathcal{M}_B, E, \Delta, M_0)$, where the set $\mathcal{M}_B$ contains all basis markings, E is the event alphabet, $\Delta \subseteq \mathcal{M}_B \times E \times \mathcal{M}_B$ denotes the transition relation among basis markings, and M_0 is the initial marking.

Algorithm 1 Construction of an BRG [45]

```
Input: A bounded labeled Petri net G= (N, M0, E, ℓ).
Output: A BRG B = (𝓜B, E, Δ, M0)
Let 𝓜B = {M0}, Δ = {};
Assign no tag to M0;
while markings with no tag exist
Select a marking M ∈ 𝓜B with no tag;
  for all t ∈ To and Ymin(M, t) ≠ ∅
    for all yu ∈ Ymin(M, t)
      M′ = M0+ Cu · yu+ C(·, t);
      if M′ ∉ 𝓜B
        𝓜B = 𝓜B∪M′;
        Assign no tag to M′;
      end if
    Δ = Δ∪{(M, ℓ(t), M′)};
    end for
    Tag node M"old";
  end for
end while
Remove all tags.
```

We give a brief description of Algorithm 1 as follows. The initial marking M_0 is regarded as the first basis marking such that the set of basis markings $\mathcal{M}_B = \{M_0\}$ is initialized. All observable transitions $t \in T_o$ should be checked for the existence of minimal e-vectors $Y_{min}(M, t)$. If it is not empty, a new basis marking M' yields by the calculation $M' = M_0 + C_u \cdot y_u + C(\cdot, t)$. The mark "old" is appended to a marking if it has not been studied yet. The BRG construction process will definitely halt sometime since the given net system satisfies the boundedness. Diverse from the traditional reachability graph construction, the core of Algorithm 1 is to calculate the firing vectors of observable transitions together with the minimal e-vectors. In the perspective of computational effort, the proposed algorithm has a great advantage over the RG calculation. In the worst case, the size of the BRG and that of the RG are identical when unobservable transitions do not exist.

The complexity of the calculation of the set $\Sigma(M,\ t)$ is shown in (5) [47]:

$$\mathrm{O}\,(B) = \mathrm{O}\left(\frac{\left(\sum_{\mathrm{i}=1}^{\mathrm{n}} y(i)\right)!}{\prod_{\mathrm{i}=1}^{\mathrm{n}} (y\,(i)\,!)}\right) \tag{5}$$

where $y(i)$ is the number of firings of transition t_i in the sequence σ. Therefore, the complexity of the BRG construction is in (6):

$$\mathrm{O}(r \cdot |\gamma| + B \cdot |\varepsilon|) \tag{6}$$

where r is the maximum number of arcs starting from any node in the BRG, $|\gamma|$ is the number of nodes in the BRG, and $|\varepsilon|$ is the number of arcs in the BRG. The size of the BRG can be an order of magnitude smaller than that of the RG. Therefore, the computational overhead and the state space are significantly reduced when the BRG is constructed.

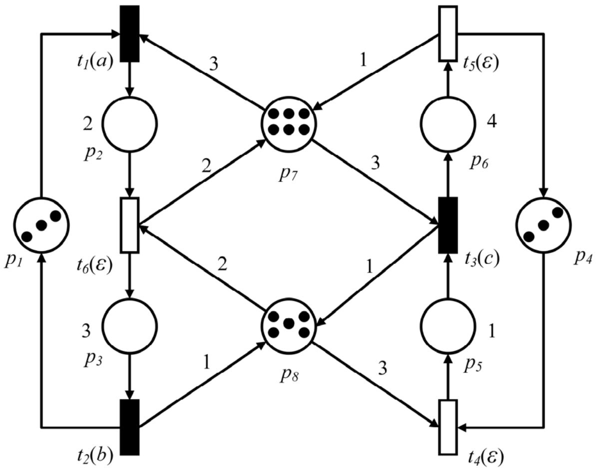

FIGURE 4. A labeled petri net with three unobservable transitions.

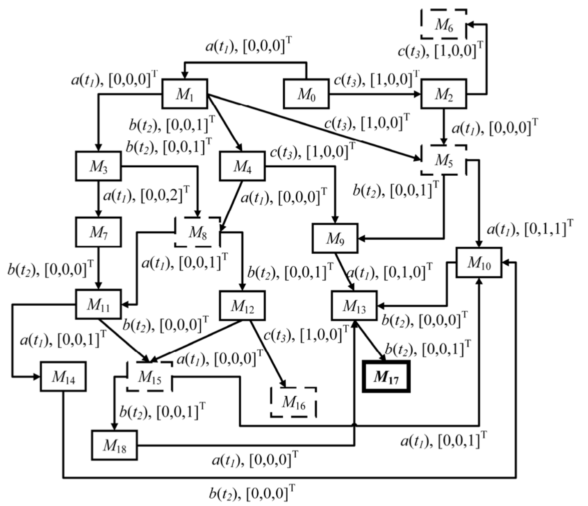

FIGURE 5. The BRG of the LPN in Fig. 4.

Example 2: Given an LPN $G = (N, M_0, E, \ell)$ with $M_0 = [3,0,0,3,0,0,6,5]^T$, the set T contains two disjoint subsets, where $T_o = \{t_1, t_3, t_5\}$ with the set of labels $E = \{$'a', 'b', 'c'$\}$, and $T_u = \{t_2, t_4, t_6\}$. The weights of arcs related to resource places are marked on the LPN in Fig. 4. The RG contains 38 reachable markings and the BRG contains 19 basis markings that is shown in Fig. 5. Table 1 shows the basis markings of the BRG in detail.

The capacity of the given net system affects the size of the RG and BRG. We apply different capacities to the LPN in Fig. 4 and the variation is shown in Table 2. Note that "o.t." stands for "overtime" if the computation is not completed within 8 hours. The computation is conducted by an Intel(R) Xeon(R) CPU E5-2650 v2 with 2.60 GHz in frequency and 128GB in storage under the operation system of Windows Server 2008 R2 Enterprise with 64 bit. Obviously, the size of the RG grows rapidly with respect to the capacity of the system while basis marking space expands slowly. As the system

TABLE 1. The basis markings in the BRG of Fig. 4.

Index	Markings	Index	Markings
M_0	$[3,0,0,3,0,0,6,5]^T$	M_{10}	$[1,1,1,3,0,0,0,1]^T$
M_1	$[2,1,0,3,0,0,3,5]^T$	M_{11}	$[1,1,1,3,0,0,1,2]^T$
M_2	$[3,0,0,2,0,1,3,3]^T$	M_{12}	$[3,0,0,3,0,0,4,3]^T$
M_3	$[1,2,0,3,0,0,0,5]^T$	M_{13}	$[2,1,0,3,0,0,0,2]^T$
M_4	$[3,0,0,3,0,0,5,4]^T$	M_{14}	$[0,1,2,3,0,0,0,0]^T$
M_5	$[2,1,0,2,0,1,0,3]^T$	M_{15}	$[2,1,0,3,0,0,1,3]^T$
M_6	$[3,0,0,1,0,2,0,1]^T$	M_{16}	$[3,0,0,2,0,1,1,1]^T$
M_7	$[0,1,2,3,0,0,1,1]^T$	M_{17}	$[3,0,0,3,0,0,2,1]^T$
M_8	$[2,1,0,3,0,0,2,4]^T$	M_{18}	$[3,0,0,3,0,0,3,2]^T$
M_9	$[3,0,0,2,0,1,2,2]^T$		

capacity increases, the advantages of the BRG construction become prominent.

B. RESOURCE-EXHAUSTED DEADLOCKS

To distinguish resource-exhausted deadlocks, we need to find out all deadlocks using the BRG first. Two situations are considered. One is that the deadlock is in the BRG, which is easy to distinguish. The other is that a dead marking is reachable by firing an unobservable sequence from one of the basis markings. An integer linear programming problem is proposed to help us.

The deadlock existing in a manufacturing system reflects that multiple processes are blocked due to unreasonable resource allocation or resource exhaustion. A dead marking enables none transitions in Petri nets. Inspired by this property, an ILP problem is proposed for deadlock identification and it is shown in (7).

$$\left\{\begin{array}{ll}\left\{\begin{array}{l} M + C_u \cdot y_{u1} \geq 0 \\ y_{u1} \geq 0, y_{u1} \in \mathrm{N} \end{array}\right. & \text{(a)} \\ \left\{\begin{array}{l} (M + C_u \cdot y_{u1}) + C_u \cdot y_{u2} \geq 0 \\ y_{u2} \geq 0, y_{u2} \in \mathrm{N} \end{array}\right. & \text{(b)} \end{array}\right. \tag{7}$$

Given a basis marking M, it is not a deadlock when the inequality group (a) has feasible solutions. The existence of a non-negative integer vector y_u that satisfies $M + C \cdot y_u \geq 0$ is the necessary and sufficient condition for the marking M' $= M + C \cdot y_u$ in acyclic nets [48]. Therefore, a new marking exists, which is generated by $M' = M + C_u \cdot y_{u1}$. When feasible solutions do not exist in the inequality group (b), M' is proven to be a deadlock. Here (M, M') is a pair that M is a basis marking and M' is a deadlock reachable from M by firing an unobservable sequence. After locking all existing deadlocks, we use (4) to distinguish resource-exhausted deadlocks and regard them as target markings. Our goal is to find sequences from M_0 to resource-exhausted deadlocks for the purpose of high resource utilization and short time.

The set of deadlocks is denoted by M_D and it is divided into two disjoint subsets. The set of resource-exhausted dead markings is denoted by M_{RED} and the set of resource-unexhausted dead markings is denoted by M_{RUD}, i.e., $M_D = M_{RED} \cup M_{RUD}$. Fig. 6 shows the trajectory from

TABLE 2. The size of the RG and the BRG of the LPN in Fig. 4.

Capacity	M_0	|RG|	|BRG|	Computation time of the RG	Computation time of the BRG	|BRG|/|RG|
8	$[1,0,0,1,0,0,3,3]^T$	8	4	0.097s	0.075s	50%
17	$[3,0,0,3,0,0,6,5]^T$	38	19	0.126s	0.122s	50%
24	$[3,0,0,3,0,0,9,9]^T$	179	52	0.290s	0.170s	29.1%
40	$[5,0,0,5,0,0,15,15]^T$	1164	193	2.994s	0.359s	16.6%
56	$[7,0,0,7,0,0,21,21]^T$	4493	472	30.529s	0.965s	10.5%
72	$[9,0,0,9,0,0,27,27]^T$	-	937	o.t.	2.533s	-

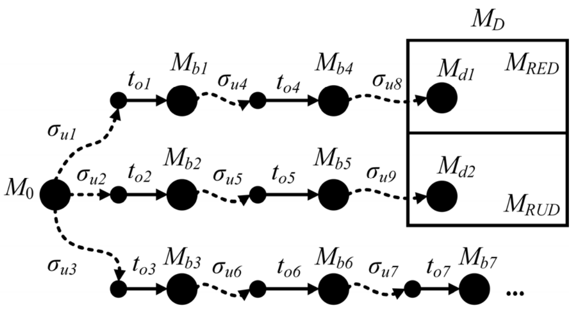

FIGURE 6. The trajectory from M_0 to the target marking.

M_0 to the set of target markings and the classification of deadlocks. Moreover, the pseudocode for finding the set of target markings is given in Algorithm 2.

Algorithm 2 Target Markings Identification

Input: A bounded LPN G_R, the set of basis markings $\mathcal{M}_B$.
Output: Set of target markings M_{RED}.
Stage 1: Calculation of M_D.
Let $M_D = \{\}$;
for each $M \in \mathcal{M}_B$
  if all $p \in \cdot t_i$, $t_i \in T$, i=1,..., n
    $M(p) < Pre(\cdot, t_i)$;
    $M_D = M_D \cup \{M\}$;
  else
    Solve the ILP (7);
    if y_{u1} exists and y_{u2} does not exist
      $M_D = M_D \cup \{M + C_u \cdot y_{u1}\}$;
    end if
  end if
end for
Stage 2:M_{RED}Identification.
for each $M_d \in M_D$
  for all $p_r \in P_R$
    $S_M(p_r) = \sum_{l(p_r) \in L(p_r)} sum(l) \cdot W(\cdot p_r, p_r) + M(p_r)$;
    if $S_M(p_r)$ <min$\{W(p_r, p_r \cdot)\}$
      $M_{RED} = M_{RED} \cup \{M_d\}$;
    end if
  end for
end for

We explain Algorithm 2 in detail. The proposed algorithm contains two stages. One is to find out deadlocks and the other is to distinguish resource-exhausted markings. The basis markings set $\mathcal{M}_B$ is obtained by Algorithm 1 and it is the input of Algorithm 2. In stage 1, each basis marking should be checked to figure out if it is a deadlock by using the transition enabling condition. Since deadlocks may not exist in $\mathcal{M}_B$, for each basis marking $M_b \in \mathcal{M}_B$ that enables the transitions in T, the ILP is solved, trying to find deadlocks by firing an unobservable sequence at M_b. All dead markings are grouped into set M_D. Then, we need to distinguish the resource-exhausted deadlocks for further study in stage 2. Each resource place should undergo the exhaustion check by using (3) and (4). If any resource-exhausted place exists at the marking $M \in M_D$, the marking M is said to be a resource-exhausted dead marking and it serves as a target marking for scheduling. All resource-exhausted deadlocks compose the set M_{RED} and it coincides with the set of target markings.

TABLE 3. The pairs of basis markings and corresponding deadlocks in the LPN of Fig. 4.

Basis markings	Unobservable firing vectors	Deadlocks
M_6=$[3,0,0,1,0,2,0,1]^T$	$[0,2,0]^T$	M_{d1}=$[3,0,0,3,0,0,2,1]^T$
M_{16}= $[3,0,0,2,0,1,1,1]^T$	$[0,1,0]^T$	
M_{17}=$[3,0,0,3,0,0,2,1]^T$	$[0,0,0]^T$	
M_{15}=$[2,1,0,3,0,0,1,3]^T$	$[1,0,0]^T$	M_{d2}=$[2,1,0,2,1,0,1,0]^T$
M_5=$[2,1,0,2,0,1,0,3]^T$	$[1,1,0]^T$	
M_8=$[2,1,0,3,0,0,2,4]^T$	$[1,0,0]^T$	M_{d3}= $[2,1,0,2,1,0,2,1]^T$

Example 3: The LPN in Fig. 4 is considered again. Applying Algorithm 2 to this example, three deadlocks are contained in M_D, which are $M_{d1} = [3,0,0,3,0,0,2,1]^T$, $M_{d2} = [2,1,0,2,1,0,1,0]^T$, and $M_{d3} = [2,1,0,2,1,0,2,1]^T$. Obviously, $M_{d1} = M_{17}$ is in the BRG and marked with a bold bar in Fig. 5. By solving the ILP, M_{d2} is reached from the basis markings $M_{15} = [2,1,0,3,0,0,1,3]^T$ and $M_5 = [2,1,0,2,0,1,0,3]^T$ by firing the unobservable sequences t_4 and t_4t_5, respectively, i.e., $M_{15}[t_4\rangle M_{d2}$ and $M_5[t_4t_5\rangle M_{d2}$. M_{d3} is reached from the basis marking $M_8 = [2,1,0,3,0,0,2,4]^T$ by firing $\sigma_u = t_4$. Moreover, M_{d1} can be reached by firing t_5t_5 and t_5 at M_6 and M_{16}, respectively. All basis markings that may reach a deadlock are marked with dotted bars in Fig. 5. Table 3 shows the pairs of basis markings and corresponding deadlocks that are reached by firing unobservable sequences.

For $M_{d1} = [3,0,0,3,0,0,2,1]^T$, we calculate $S_M(p_r)$ for each resource place using (3) and (4) and we have

$$2 \times M(p_2) + 1 \times M(p_6) + M(p_7)$$

$$= 2 < \min\{W(p_7,\ p_7\cdot)\} = 3$$
$$1 \times M(p_3) + 1 \times M(p_5) + M(p_8)$$
$$= 1 < \min\{W(p_8,\ p_8\cdot)\} = 2$$

M_{d1} is a resource-exhausted marking since both p_7 and p_8 are resource-exhausted. Similarly, $S_M(p_7) = 3$ and $S_M(p_8) = 1$ in M_{d2}. Therefore, only p_8 is resource-exhausted such that M_{d2} also belongs to M_{RED}. As for M_{d3}, $S_M(p_7) = 4$ and $S_M(p_8) = 2$, the deadlock is not generated by resource exhaustion. Table 4 shows dead markings calculated by Algorithm 2 and their classification.

TABLE 4. Classification of deadlocks in the LPN of Fig. 4.

M_D	$S_M(p_7)$	$S_M(p_8)$	Resource exhausted places	Classification
$[3,0,0,3,0,0,2,1]^T$	2	1	$\{p_7, p_8\}$	M_{RED}
$[2,1,0,2,1,0,1,0]^T$	3	1	$\{p_8\}$	M_{RED}
$[2,1,0,2,1,0,2,1]^T$	4	2	/	M_{RUD}

V. SEQUENCE PLANNING USING BASIS MARKING ANALYSIS

According to the algorithms in Section IV, we can identify the target markings and marked them on the BRG together with the source marking. In this section, we propose a graph-based heuristic algorithm to find an ideal transition sequence from the initial marking M_0 to the target markings, for the purpose of the shortest process completion time and full use of resources.

The A-star algorithm is developed from the Dijkstra algorithm for better computational efficiency. Heuristic information is used in the A-star algorithm in the basis marking space such that traversing all nodes in the search area is avoided. For the proposed graphic search algorithm, a cost function needs to be established to estimate the total cost of each basis marking, to determine the direction of the evolving. Once the target marking is reached, their parent markings are backtracked until M_0, constructing an evolution $\lambda(M_0)$. The transition sequence $\sigma[\lambda(M_0)]$ provides the sequence of all processes, forming a scheduling plan. Note that transitions may fire concurrently. Given a marking $M \in \mathcal{M}_B$, the total cost function of M is denoted by $F(M)$, as shown in (8).

$$F(M) = I(M) + H(M) \tag{8}$$

where $I(M)$ is the minimum cost, i.e., the shortest processing time, from M_0 to the current marking M, and $H(M)$ is the estimated cost from M to the target marking. In this paper, we define that $H(M) = -Dep(M)$, where $Dep(M)$ is the depth of the evolution, i.e., the number of fired transitions from M_0 to M. The marking with high cost will be compensated if the planning sequence is long enough.

The value of $I(M)$ depends on the delay time of the transition sequence $\sigma_u t_o$ where M' $[\sigma_u t_o\rangle M$, M', $M \in \mathcal{M}_B$. The algorithm always records the minimum $I(M)$ and the corresponding trajectory while computing. A transition cannot be fired immediately if its pre-places contain activity places with time constraints. Obviously, the retention time of tokens in its pre-places should be longer than or equal to the time constraints associated with the pre-places. The cost of each arc in the basis marking space is the maximum value of the remaining delay time in its pre-places. Given a marking M and its parent marking M', i.e., M' $[t\rangle M$, the cost from M' to M is denoted by $c(M', M)$, as shown in (9).

$$c(M',\ M) = \max(r_{1j}, r_{2j}, \ldots,\ r_{kj}) \tag{9}$$

where r_{ij} is the residual delay time of the pre-place p_{ij} of the transition t_j, i=1,2,…, k.

Algorithm 3 Heuristic Sequence Planning Algorithm

```
Input: An LPN G_R, the set of target markings M_RED.
Output: A feasible evolution λ(M) and σ[λ(M)].
Initialize the set Op={M_0}, Clo={};
while Op≠ Ø
  for M ∈ Op
    Op = Op \{M};
    Clo = Clo∪{M};
    if M ∈ M_RED
      Backtrace the current path;
      break;
    else
      Find the next generated nodes of M in the BRG;
      for any child node M_s
        F(M_s) = I(M_s) + H(M_s);
        if M_s ∈Op
          Backtrace the path with F_min(M);
        else if M_s ∈Clo
          Backtrace the path with F_min(M);
        else
          Op = Op ∪ {M_s};
        end if
      end for
    end if
  Rank markings M_s ∈ Op in incremental order of F(M);
  end for
end while
```

Algorithm 3 gives the pseudocode of the scheduling algorithm. First, the BRG together with deadlocks is served as the search zone and M_0 is in the set Op that contains existed markings which are waiting for exploring. The basis marking space is the refinement of the RG-based search space such that state space explosion can be avoided. We choose the first marking M from the Op list and add it to the set Clo. If M is the target marking, the algorithm halts and the evolution is traced backwardly. Otherwise, child markings of M are noticed, and the cost function calculates the estimated total cost of each child marking.

If the newly generated marking M_s is neither in Op nor in Clo, it will be placed in the Op as a marking for inspection. If the new marking M_s already exists in Op, it indicates that another trajectory to M_s has been found, and we only

TABLE 5. Scheduling plan from M_0 to M_{17}.

Index	Activity place	Operation time	Index	Activity place	Operation time
1	p_5	1	3	p_6	4
2	p_5	1	4	p_6	4
Accumulated operation time		10	Planned completion time		6

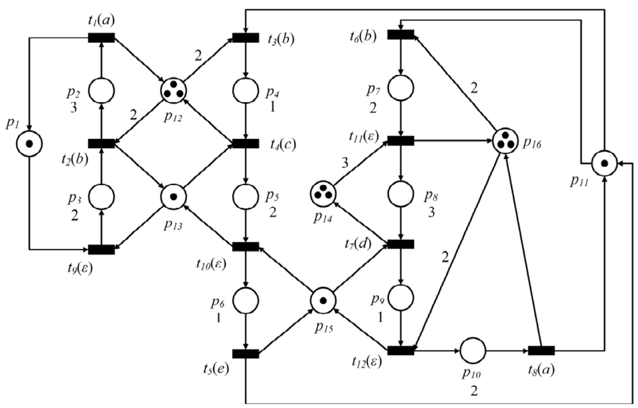

FIGURE 7. The LPN of an FMS.

choose the trajectory with the lowest cost. If the new marking is in *Clo* and the new cost is lower, then the new marking should be relocated in *Op*. The algorithm prioritizes the newly generated low-cost marking for subsequent searching and it ceases if the new calculated marking is the target marking.

The performance of Algorithm 3 is measured by its finiteness, validity and computational complexity.

Proposition 4: The A-star algorithm based on the basis marking space (Algorithm 3) halts in finite steps.

Proof: Algorithm 3 is under the assumption that the given LPN is bounded and the T_u-induced subnet is acyclic. The minimal explanations associated with a basis marking are finite since unobservable transitions are not able to be fired infinitely in an acyclic net. Hence, at each iteration of Algorithm 3, a finite transition sequence can be found and the calculation time is also finite. In the worst case, the iteration will stop when the whole BRG is traversed and the calculation is finite since the size of the BRG is limited for bounded nets. The finiteness of Algorithm 3 is guaranteed.

Proposition 5: Given an LPN with time and resource constraints whose initial marking M_0 is not a deadlock, an evolution from M_0 to a resource-exhausted deadlock, i.e., a target marking, must exist and the results of the A-star algorithm based on the basis marking space (Algorithm 3) is optimal or sub-optimal.

Proof: First, we prove the existence of the results of Algorithm 3. Since the given net system is resource-consuming, deadlocks caused by resource-exhaustion are inevitably reached. The existence of target markings is ensured. In the worst case, the size of the BRG constructed by Algorithm 1 is the same as that of the RG. The nodes in the BRG are linked by arcs representing fired transitions. Hence, the successor nodes are always reachable from their precursor markings at each iteration of Algorithm 3. The iteration stops when the precursor marking is the target marking and all reachable states from M_0 to the target marking together with fired transitions form a valid evolution. The existence of a corresponding sequence is ensured.

Then, we prove that the results of Algorithm 3 are optimal or sub-optimal, i.e., the cost of obtained sequence is least or nearly least. Each sequence from a basis marking to another in the BRG is shortest with least-cost. We prove this by contradiction. Assume that the heuristic information is ignored, i.e., the proposed algorithm is the traditional Dijkstra algorithm. We claim that the optimal sequence is σ with the calculated least cost $F_{\sigma'}(M)$, where $M \in M_{RED}$. First, the existence of a sequence $\sigma' \neq \sigma$ with $M_0[\sigma'\rangle M$ and $F_{\sigma'}(M) < F_{\sigma}(M)$ is supposed. We have the evolution of σ':

$$M_0[\sigma_1 t_1\rangle M_1[\sigma_2 t_2\rangle M_2 \ldots M_{x-1}[\sigma_x t_x\rangle M_x[\sigma_{x+1}\rangle M$$

If σ_1 is not a minimal explanation of t_1, M_1 is not a basis marking and there necessarily exists a minimal explanation satisfying

$$M_0[\sigma_{11} t_1\rangle M_{11}[\sigma_{12}\sigma_2 t_2\rangle M_2 \ldots M_{x-1}[\sigma_x t_x\rangle M_x[\sigma_{x+1}\rangle M$$

where $\sigma_{11} < \sigma_1$. Similarly, if σ_2 is not a minimal explanation of t_2, $\sigma_{21} < \sigma_2$ can be found as a minimal explanation. We rewrite the sequence of σ' repeatedly as:

$$M_0[\sigma_{11} t_1\rangle M_{11}[\sigma_{21} t_2\rangle M_{21} \ldots M_{x-1,1}[\sigma_{x,1} t_x\rangle M_{x1}[\sigma_{x+1,1}\rangle M$$

where each σ_{x1} is a minimal explanation of t_x and all $M_{x-1,1}$ are basis markings. This implies that σ' is not the shortest sequence, which contradicts the supposition that $F_{\sigma'}(M)$ is minimal. Therefore, the sequence calculated by the proposed algorithm in the basis marking space is shortest and the cost of the planning sequence is minimal. Note that the optimality of the algorithm is strongly influenced by the composition of the cost function. If $H(M) = -Dep(M)$ is used as the heuristic cost estimate, suboptimal solutions can be found in a relatively short period of time.

Proposition 6: Given an LPN G_R, the complexity of Algorithm 3 is $O(|\mathcal{M}_B|^2)$.

Proof: The complexity of Dijkstra algorithm is polynomial with respect to the node number. While doing a Dijkstra search in a digraph with n nodes, the complexity is $O(|n|^2)$ [40]. In the worst case, heuristic information does no contribution and the basis marking space will be fully traversed. Therefore, the complexity of Algorithm 3 is $O(|\mathcal{M}_B|^2)$ and the problem in this paper is NP-hard when $\mathcal{M}_B = R(G_R, M_0)$. Usually, it takes advantages over RG-based Dijkstra algorithm since the reachable marking space is generally much greater than the basis marking space in practice.

Example 4: Consider again the LPN in Fig. 4. The source marking is $M_0 = [3,0,0,3,0,0,6,5]^{\mathrm{T}}$ and we choose $M_{17} = [3,0,0,3,0,0,2,1]^{\mathrm{T}}$ as the target marking. The set of time constraints is $\Gamma = [2,3,1,4]^{\mathrm{T}}$ for $P_A = \{p_2, p_3, p_5, p_6\}$.

TABLE 6. The markings in the BRG of the net system in Fig. 7.

Index	Markings	Index	Markings	Index	Markings
M_0	$[1,0,0,0,0,0,0,0,0,0,1,3,1,3,1,3]^T$	M_{11}	$[0,1,0,0,0,0,0,0,1,0,0,1,1,1,0,2]^T$	M_{22}	$[0,1,0,0,0,0,0,0,1,0,0,0,1,1,0,2]^T$
M_1	$[0,1,0,0,0,0,0,0,0,0,1,1,1,3,1,3]^T$	M_{12}	$[0,1,0,0,0,1,0,0,0,0,0,0,1,3,0,3]^T$	M_{23}	$[1,0,0,0,0,0,0,0,0,0,1,2,1,1,1,1]^T$
M_2	$[1,0,0,1,0,0,0,0,0,0,0,1,1,3,1,3]^T$	M_{13}	$[1,0,0,0,0,0,0,0,0,0,1,3,1,1,1,1]^T$	M_{24}	$[1,0,0,0,1,0,0,0,0,0,0,2,0,1,1,1]^T$
M_3	$[1,0,0,0,0,0,1,0,0,0,0,3,1,3,1,1]^T$	M_{14}	$[1,0,0,0,0,0,0,0,0,0,1,1,1,3,1,3]^T$	M_{25}	$[1,0,0,0,0,0,0,0,1,0,0,1,1,1,0,2]^T$
M_4	$[1,0,0,0,0,0,0,0,0,0,1,2,1,3,1,3]^T$	M_{15}	$[0,1,0,0,0,0,1,0,0,0,0,0,1,3,1,1]^T$	M_{26}	$[0,1,0,0,0,0,0,0,0,0,1,0,1,1,1,1]^T$
M_5	$[0,1,0,0,0,0,1,0,0,0,0,1,1,3,1,1]^T$	M_{16}	$[1,0,0,0,1,0,0,0,0,0,0,1,0,3,1,3]^T$	M_{27}	$[1,0,0,1,0,0,0,0,0,0,0,0,1,1,1,1]^T$
M_6	$[1,0,0,0,1,0,0,0,0,0,0,2,0,3,1,3]^T$	M_{17}	$[1,0,0,0,0,0,0,0,1,0,0,2,1,1,0,2]^T$	M_{28}	$[0,1,0,0,0,1,0,0,0,0,0,0,1,1,0,1]^T$
M_7	$[1,0,0,0,0,0,0,0,1,0,0,3,1,1,0,2]^T$	M_{18}	$[0,1,0,0,0,0,0,0,0,0,1,1,1,1,1,1]^T$	M_{29}	$[1,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1]^T$
M_8	$[0,1,0,0,0,0,0,0,0,0,1,0,1,3,1,3]^T$	M_{19}	$[1,0,0,0,0,1,0,0,0,0,0,1,1,3,0,3]^T$	M_{30}	$[1,0,0,0,1,0,0,0,0,0,0,1,0,1,1,1]^T$
M_9	$[1,0,0,1,0,0,0,0,0,0,0,0,1,3,1,3]^T$	M_{20}	$[1,0,0,1,0,0,0,0,0,0,0,1,1,1,1,1]^T$	M_{31}	$[1,0,0,0,0,1,0,0,0,0,0,1,1,1,0,1]^T$
M_{10}	$[1,0,0,0,0,0,1,0,0,0,0,2,1,3,1,1]^T$	M_{21}	$[1,0,0,0,0,0,1,0,0,0,0,1,1,3,1,1]^T$		

TABLE 7. Deadlocks in the net system of Fig. 7.

Index	Markings	$S_M(p_{12})$	$S_M(p_{14})$	$S_M(p_{16})$	Resource exhausted places	$\in M_{RED}$
M_{d1}	$[0,0,1,1,0,0,0,0,0,0,0,1,0,3,1,3]^T$	2	4	3	/	N
M_{d2}	$[0,0,1,1,0,0,0,0,0,0,0,0,0,3,1,3]^T$	1	3	3	$\{p_{12}\}$	Y
M_{d3}	$[0,0,1,1,0,0,0,0,0,0,0,1,0,1,1,1]^T$	2	1	1	$\{p_{14},p_{16}\}$	Y
M_{d4}	$[0,0,1,1,0,0,0,0,0,0,0,0,0,1,1,1]^T$	1	1	1	$\{p_{12},p_{14},p_{16}\}$	Y
M_{d5}	$[0,0,1,0,0,0,0,0,0,0,1,1,0,1,1,1]^T$	1	1	1	$\{p_{12},p_{14},p_{16}\}$	Y

TABLE 8. Optimal sequences form M_0 to target markings.

Target markings	Accumulated operation time	Planned completion time	Depth	Feasible optimal evolutions
M_{d2}	4	4	6	$M_0t_3M_2t_4M_6t_{10}t_5M_4t_3M_9t_9M_{d2}$
M_{d3}	8	8	7	$M_0t_6M_3t_{11}t_7M_7t_{12}t_8M_{13}t_3M_{20}t_9M_{d3}$
M_{d4}	13	8	10	$M_0t_6M_3t_9t_2M_5t_{11}t_7M_{11}t_1M_{17}t_{12}t_8M_{23}t_3M_{27}t_9M_{d4}$
M_{d5}	18	10	12	$M_0t_6M_3t_9t_2M_5t_{11}t_7M_{11}t_1M_{17}t_9t_2M_{22}t_{12}t_8M_{26}t_1M_{29}t_9M_{d5}$

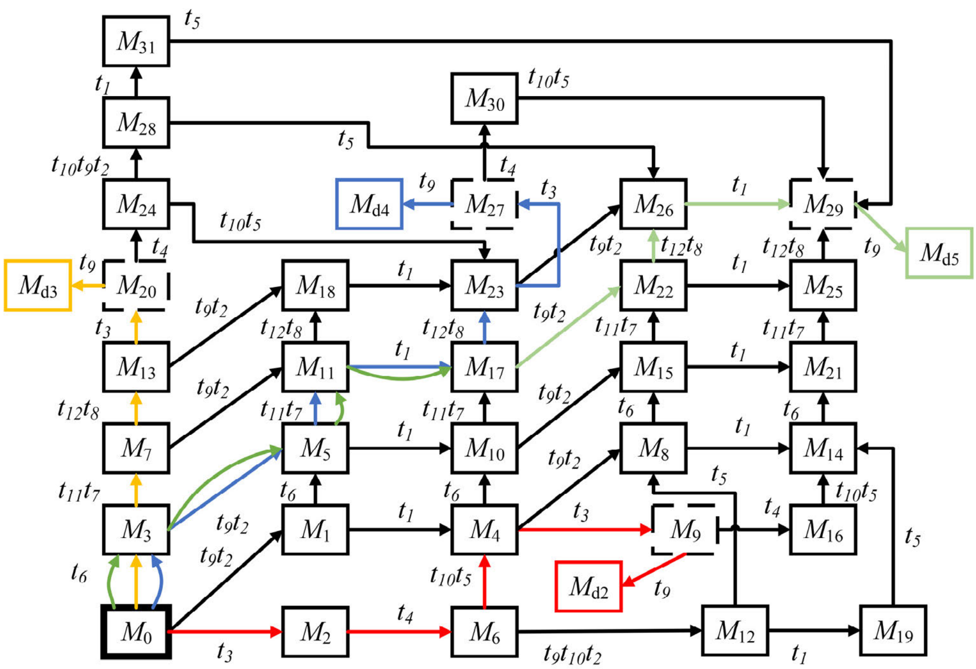

FIGURE 8. Scheduled sequences in the basis reachability space.

Algorithm 3 is used for sequence planning. The optimal evolution is obtained as $\lambda(M_0) = M_0t_3t_4M_2t_4t_3M_6t_5t_5M_{17}$ and the corresponding sequence is $\sigma[\lambda(M_0)] = t_4t_3t_4t_3t_5t_5$ and the total time is recorded as 6 units of time. The cost of the sequence is 6 and the depth of M_{17} is 6. Note that the planned completion time is much shorter than the accumulated operation time which means that transitions fire concurrently. Table 5 shows the orders of operation.

VI. NUMERICAL EXAMPLE

In this section, a numerical example is given to verify the validity of the algorithm. Let us consider an FMS modeled by the LPN in Fig. 7, where $M_0 = [1,0,0,0,0,0,0,0,0,0,1,3,1,3,1,3]^{\mathrm{T}}$. The LPN contains 16 places and 12 transitions. Three production lines are showed in the net system, which are $t_9p_3t_2p_2t_1$, $t_3p_4t_4p_5t_{10}p_6t_5$, $t_6p_7t_{11}p_8t_7p_9t_{12}p_{10}$, respectively. The sets of places are $p^0 = \{p_1, p_{11}\}$, $P_A = \{p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}\}$, and $P_R = \{p_{12}, p_{13}, p_{14}, p_{15}, p_{16}\}$. The set of transitions is divided into two disjoint subsets $T = T_{\mathrm{u}} \cup T_{\mathrm{o}}$, where $T_u = \{t_9, t_{10}, t_{11}, t_{12}\}$. The timed vector corresponded with activity places p_2 to p_{10} is $\Gamma = [3,2,1,2,1,2,3,1,2]^{\mathrm{T}}$. The weights of arcs related to resource places that do not equal 1 are marked in Fig. 7 and obviously, p_{12}, p_{14}, p_{16} are token-consuming resource places.

The size of $R(N, M_0)$ is 70 and the size of the BRG is 32 that is shown in Table 6 in detail. According to Algorithm 2, none of the basis markings is a deadlock but some of them lead to deadlocks by firing unobservable transitions only. Five deadlocks are found out and four of them are distinguished as resource-exhaustion deadlocks. Details show in Table 7, where "Y" denotes that the marking belongs to M_{RED}.

The planning sequence in the basis marking space is shown in Fig. 8 and Table 8 shows the results of the scheduled sequences for each target marking. In Fig. 8, colored lines show the trajectories from M_0 to each target marking. For example, M_{d5} can be reached by firing a sequence $(t_6t_9)(t_2t_{11})(t_7t_1)t_9t_2t_{12}t_8t_1t_9$, where the transitions in brackets can be fired concurrently to save time. The accumulated operation time is 18 units of time but the planned completion time only takes 10.

VII. CONCLUSION

This paper addresses the scheduling problem for labeled Petri nets with time and resource constraints. We take advantage of the basis reachability graph to avoid state space explosion. Dead markings caused by resource exhaustion are found using an integer linear programming problem and the notion of resource-exhausted markings. We formulate a heuristic algorithm to schedule the firings of transitions. The calculated sequence starts from the initial marking and ends at a resource-exhausted deadlock in the shortest time. The proposed method is able to find an optimal or suboptimal result without traversing all reachable states. In our future work, we try to focus on the supervisory control problem of labeled Petri nets with time and resource constraints to enforce the system evolving in desired sequences. Moreover, applications to flexible manufacturing systems will be considered.

ACKNOWLEDGMENT

The authors extend their appreciation to King Saud University, Saudi Arabia, for funding this work through Researchers Supporting Project number (RSP2023R133), King Saud University, Riyadh, Saudi Arabia.

REFERENCES

[1] L. E. Holloway, B. H. Krogh, and A. Giua, "A survey of Petri net methods for controlled discrete event systems," *Discrete Event Dyn. Syst.*, vol. 7, no. 2, pp. 151–190, Apr. 1997.

[2] M. Russo and A. Sasso, "Modeling, analysis, simulation and control of laboratory automation systems using Petri nets: Part 1. Modeling," *J. Assoc. Lab. Autom.*, vol. 10, no. 3, pp. 172–181, Jun. 2005, doi: 10.1016/j.jala.2005.04.005.

[3] M. Zhou and F. DiCesare, "Parallel and sequential mutual exclusions for Petri net modeling of manufacturing systems with shared resources," *IEEE Trans. Robot. Autom.*, vol. 7, no. 4, pp. 515–527, Aug. 1991.

[4] J. Ezpeleta, J. M. Colom, and J. Martinez, "A Petri net based deadlock prevention policy for flexible manufacturing systems," *IEEE Trans. Robot. Autom.*, vol. 11, no. 2, pp. 173–184, Apr. 1995, doi: 10.1109/70.370500.

[5] C. Ramchandani, "Analysis of asynchronous concurrent systems by timed Petri nets," Ph.D. dissertation, MIT, Cambridge, MA, USA, 1974.

[6] G. Florin and S. Natkin, "Evaluation based upon stochastic Petri nets of the maximum throughput of a full duplex protocol," in *Application and Theory of Petri Nets*, C. Girault and W. Resig, Eds. New York, NY, USA: Springer, 1982, pp. 280–288, doi: 10.1007/978-3-642-68353-4_45.

[7] K. Jensen, *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use*, vol. 2. Berlin, Germany: Springer, 1995.

[8] Y. Chen, Z. Li, and M. Zhou, "Optimal supervisory control of flexible manufacturing systems by Petri nets: A set classification approach," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 2, pp. 549–563, Apr. 2014, doi: 10.1109/TASE.2013.2241762.

[9] M. Bashir, J. Zhou, and B. B. Muhammad, "Optimal supervisory control for flexible manufacturing systems model with Petri nets: A place-transition control," *IEEE Access*, vol. 9, pp. 58566–58578, 2021, doi: 10.1109/ACCESS.2021.3072892.

[10] X. Cong, C. Gu, M. Uzam, Y. Chen, A. M. Al-Ahmari, N. Wu, M. Zhou, and Z. Li, "Design of optimal Petri net supervisors for flexible manufacturing systems via weighted inhibitor Arcs," *Asian J. Control*, vol. 20, no. 1, pp. 511–530, Jan. 2018, doi: 10.1002/asjc.1583.

[11] P. N. Pena, T. A. Costa, R. S. Silva, and R. H. C. Takahashi, "Control of flexible manufacturing systems under model uncertainty using supervisory control theory and evolutionary computation schedule synthesis," *Inf. Sci.*, vol. 329, pp. 491–502, Feb. 2016, doi: 10.1016/j.ins.2015.08.056.

[12] P. Lv, G. Luo, Z. Ma, S. Li, and X. Yin, "Optimal multi-robot path planning for cyclic tasks using Petri nets," *Control Eng. Pract.*, vol. 138, Sep. 2023, Art. no. 105600, doi: 10.1016/j.conengprac.2023.105600.

[13] Z. Ma, M. Zou, J. Zhang, and Z. Li, "Design of optimal control sequences in Petri nets using basis marking analysis," *IEEE Trans. Autom. Control*, vol. 67, no. 7, pp. 3685–3692, Jul. 2022, doi: 10.1109/TAC.2021.3106883.

[14] A. Casalino, A. M. Zanchettin, L. Piroddi, and P. Rocco, "Optimal scheduling of human–robot collaborative assembly operations with time Petri nets," *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 1, pp. 70–84, Jan. 2021, doi: 10.1109/TASE.2019.2932150.

[15] Z. He, R. Zhang, N. Ran, and C. Gu, "Path planning of multi-type robot systems with time windows based on timed colored Petri nets," *Appl. Sci.*, vol. 12, no. 14, p. 6878, Jul. 2022, doi: 10.3390/app12146878.

[16] M. Drakaki and P. Tzionas, "Manufacturing scheduling using colored Petri nets and reinforcement learning," *Appl. Sci.*, vol. 7, no. 2, p. 136, Feb. 2017, doi: 10.3390/app7020136.

[17] M. Kammoun, W. Ezzeddine, N. Rezg, and Z. Achour, "FMS scheduling under availability constraint with supervisor based on timed Petri nets," *Appl. Sci.*, vol. 7, no. 4, p. 399, Apr. 2017, doi: 10.3390/app7040399.

[18] G. Zhu, L. Feng, Z. Li, and N. Wu, "An efficient fault diagnosis approach based on integer linear programming for labeled Petri nets," *IEEE Trans. Autom. Control*, vol. 66, no. 5, pp. 2393–2398, May 2021, doi: 10.1109/TAC.2020.3008712.

[19] A. Al-Ajeli and D. Parker, "Fault diagnosis in labelled Petri nets: A Fourier–Motzkin based approach," *Automatica*, vol. 132, Oct. 2021, Art. no. 109831, doi: 10.1016/j.automatica.2021.109831.

[20] Y. Li, Y. Wang, G. Zhu, L. Yin, and H. Zhang, "Fault diagnosis of PLC-based discrete event systems using Petri nets," *Meas. Control*, vol. 55, nos. 9–10, pp. 960–973, Aug. 2022, doi: 10.1177/00202940221117098.

[21] F. Arichi, B. Cherki, M. Djemai, and S. M. Djouadi, "Fault diagnosis for discrete events systems described by partially observed Petri nets," *ISA Trans.*, vol. 128, pp. 220–228, Sep. 2022, doi: 10.1016/j.isatra.2021.09.010.

[22] P. J. G. Ramadge and W. M. Wonham, "The control of discrete event systems," *Proc. IEEE*, vol. 77, no. 1, pp. 81–98, Jan. 1989.

[23] P. J. Ramadge and W. M. Wonham, "Supervisory control of a class of discrete event processes," *SIAM J. Control Optim.*, vol. 25, no. 1, pp. 206–230, Jan. 1987.

[24] P. J. Ramadge and W. M. Wonham, "Modular feedback logic for discrete event systems," *SIAM J. Control Optim.*, vol. 25, no. 5, pp. 1202–1218, Sep. 1987.

[25] Y. Feng, K. Xing, M. Zhou, and H. Liu, "Liveness analysis and deadlock control for automated manufacturing systems with multiple resource requirements," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 2, pp. 525–538, Feb. 2020, doi: 10.1109/TSMC.2017.2767902.

[26] Y. Chen, Z. Li, K. Barkaoui, N. Wu, and M. Zhou, "Compact supervisory control of discrete event systems by Petri nets with data inhibitor arcs," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 2, pp. 364–379, Feb. 2017, doi: 10.1109/TSMC.2016.2521833.

[27] M. P. Cabasino, C. N. Hadjicostis, and C. Seatzu, "Probabilistic marking estimation in labeled Petri nets," *IEEE Trans. Autom. Control*, vol. 60, no. 2, pp. 528–533, Feb. 2015.

[28] T. C. Row and Y. L. Pan, "Maximally permissive deadlock prevention policies for flexible manufacturing systems using control transition," *Adv. Mech. Eng.*, vol. 10, no. 7, pp. 1–10, Jul. 2018, doi: 10.1177/1687814018787406.

[29] Y. S. Huang and T. C. Row, "Petri net channelized-based deadlock prevention policy for flexible manufacturing systems," *Adv. Mater. Res.*, vols. 317–319, pp. 552–555, Aug. 2011, doi: 10.4028/www.scientific.net/AMR.317-319.552.

[30] M. P. Cabasino, A. Giua, and C. Seatzu, "Diagnosability of discrete-event systems using labeled Petri nets," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 1, pp. 144–153, Jan. 2014.

[31] R. Sampath, H. Darabi, U. Buy, and L. Jing, "Control reconfiguration of discrete event systems with dynamic control specifications," *IEEE Trans. Autom. Sci. Eng.*, vol. 5, no. 1, pp. 84–100, Jan. 2008.

[32] J. Li, M. Zhou, T. Guo, Y. Gan, and X. Dai, "Robust control reconfiguration of resource allocation systems with Petri nets and integer programming," *Automatica*, vol. 50, no. 3, pp. 915–923, Mar. 2014.

[33] C. Mahulea and M. Kloetzer, "Robot planning based on Boolean specifications using Petri net models," *IEEE Trans. Autom. Control*, vol. 63, no. 7, pp. 2218–2225, Jul. 2018.

[34] Z. He, Y. Dong, G. Ren, C. Gu, and Z. Li, "Path planning for automated guided vehicle systems with time constraints using timed Petri nets," *Meas. Control*, vol. 53, nos. 9–10, pp. 2030–2040, Nov. 2020, doi: 10.1177/0020294020964840.

[35] H. Dezani, R. D. S. Bassi, N. Marranghello, L. Gomes, F. Damiani, and I. N. D. Silva, "Optimizing urban traffic flow using genetic algorithm with Petri net analysis as fitness function," *Neurocomputing*, vol. 124, pp. 162–167, Jan. 2014, doi: 10.1016/j.neucom.2013.07.015.

[36] Y. Gan, Y. He, L. Gao, and W. He, "Propagation path optimization of product attribute design changes based on Petri net fusion ant colony algorithm," *Expert Syst. Appl.*, vol. 173, Jul. 2021, Art. no. 114664, doi: 10.1016/j.eswa.2021.114664.

[37] W. Dai, Y. Hu, Z. Zhu, and X. Liao, "Human resource Petri net allocation model based on artificial intelligence and neural network," *Mobile Inf. Syst.*, vol. 2021, pp. 1–13, Aug. 2021, doi: 10.1155/2021/5988742.

[38] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Math.*, vol. 1, no. 1, pp. 269–271, Dec. 1959.

[39] Risald, A. E. Mirino, and Suyoto, "Best routes selection using Dijkstra and Floyd–Warshall algorithm," in *Proc. 11th Int. Conf. Inf. Commun. Technol. Syst. (ICTS)*, Oct. 2017, pp. 155–158.

[40] Q. Zhou, F. W. Si, and Y. B. Xiu, "Research on the parallel test task scheduling method with Petri nets and Dijkstra algorithm," *J. Electron. Meas. Instrum.*, vol. 29, no. 6, pp. 920–927, 2015.

[41] L. Xuan and H. Shengguo, "The simulation algorithm of military transportation shortest path based on Petri net," in *Proc. ISECS Int. Colloq. Comput., Commun., Control, Manage.*, vol. 2, Aug. 2009, pp. 111–114.

[42] F. Duchoň, A. Babinec, M. Kajan, P. Beňo, M. Florek, T. Fico, and L. Jurišica, "Path planning with modified a star algorithm for a mobile robot," *Proc. Eng.*, vol. 96, pp. 59–69, Jan. 2014.

[43] T. Murata, "Petri nets: Properties, analysis and applications," *Proc. IEEE*, vol. 77, no. 4, pp. 541–580, Apr. 1989.

[44] G. Y. Liu, "Supervisor synthesis for automated manufacturing systems based on structure theory of Petri nets" Ph.D. dissertation, Informatique, CNAM, Paris, France, 2014.

[45] M. P. Cabasino, A. Giua, and C. Seatzu, "Fault detection for discrete event systems using Petri nets with unobservable transitions," *Automatica*, vol. 46, no. 9, pp. 1531–1539, Sep. 2010.

[46] M. P. Cabasino, A. Giua, M. Pocci, and C. Seatzu, "Discrete event diagnosis using labeled Petri nets. An application to manufacturing systems," *Control Eng. Pract.*, vol. 19, no. 9, pp. 989–1001, Sep. 2011.

[47] H. Yue, S. Xu, G. Zhou, H. Hu, Y. Guo, and J. Zhang, "Estimation of least-cost transition firing sequences in labeled Petri nets by using basis reachability graph," *IEEE Access*, vol. 7, pp. 165387–165398, 2019.

[48] D. Corona, A. Giua, and C. Seatzu, "Marking estimation of Petri nets with silent transitions," in *Proc. 43rd IEEE Conf. Decis. Control*, Nassau, The Bahamas, Dec. 2004, pp. 966–971.

YEJIA LIU (Graduate Student Member, IEEE) received the B.S. degree from Xiangtan University, Xiangtan, China, in 2016. She is currently pursuing the Ph.D. degree in mechanical engineering with the School of Mechanical and Electrical Engineering, University of Electronic Science and Technology of China, Chengdu, China. Her current research interests include Petri nets, supervisory control, and fault diagnosis in discrete event systems.

XUNBO LI was born in Chengdu, China, in 1963. He received the B.S. degree in mechanical engineering, the M.S. degree in electronic engineering, and the Ph.D. degree in automatic control from the University of Electronic Science and Technology of China, Chengdu, in 1985, 1991, and 2001, respectively. He was a Visiting Professor with Victoria University, Melbourne, Australia, in 2006. He is currently a Professor with the School of Mechanical and Electrical Engineering, University of Electronic Science and Technology of China. Over the past decade, he has focused on research in intelligent industrial manufacturing. His current research interests include cyber-physical systems, machine vision, wireless sensor networks, intelligent scheduling, and fault diagnosis in flexible manufacturing systems.

AHMED M. EL-SHERBEENY received the master's and Ph.D. degrees in mechanical engineering from West Virginia University (WVU), in 2001 and 2006, respectively. He was a Graduate Teaching Assistant and a Research Assistant with WVU. Since 2010, he has been an Assistant Professor with the Industrial Engineering Department. He was the Former Head of the Alumni and Employment Unit, College of Engineering, King Saud University, from 2013 to 2018. His research interests include human factors engineering, manufacturing engineering, and engineering education.