IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 3, 2019, accepted January 24, 2019, date of publication February 4, 2019, date of current version February 14, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2896254

Liveness of Disjunctive and Strict Single-Type Automated Manufacturing System: An ROPN Approach

HEFENG CHEN[1], NAIQI WU[2,3], (Fellow, IEEE), ZHIWU LI[2,4], (Fellow, IEEE), TING QU[5], AND HAILIAN XIAO[6]

[1]School of Applied Mathematics, Guangdong University of Technology, Guangzhou 510006, China
[2]Institute of Systems Engineering, Macau University of Science and Technology, Macau 999078, China
[3]National Key Laboratory of Precise Electronic Manufacturing Technology and Equipment, Guangdong University of Technology, Guangzhou 510006, China
[4]School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China
[5]School of Electrical and Information Engineering, Jinan University at Zhuhai, Zhuhai 519070, China
[6]School of Economics and Management, South China Normal University, Guangzhou 510006, China

Corresponding authors: Naiqi Wu (nqwu@must.edu.mo) and Zhiwu Li (zhwli@xidian.edu.cn)

This work was supported in part by the Science and Technology Development Fund (FDCT) of Macau under Grant 106/2016/A3, Grant 122/2017/A3, and Grant 011/2017/A, and in part by the National Science Foundation of China under Grant U1401240.

ABSTRACT The resource-oriented Petri net (ROPN) modeling method is applied in this paper to model a class of automated manufacturing systems (AMS) named disjunctive and strict single-type AMSs (DS-AMS), where multiple units of a single resource type are required to support a processing stage. This modeling method is more compact and intuitive than the existing ones in the contemporary literature, named weighted systems of simple sequential processes with resources. Based on the ROPN model, a significant result is re-proved in a different way, which is more succinct and understandable. Such a result presents a sufficient and necessary condition on the liveness of an ROPN. Furthermore, based on this result, we develop efficient strategies to ensure the liveness of an ROPN modeling a DS-AMS. To enforce the liveness of such an ROPN, only some specific transitions need to be controlled by taking advantage of its structural properties. Also, an efficient approach is developed to implement the proposed policy by identifying and controlling such transitions. Two examples are employed to demonstrate the efficiency of the developed method. Some limitations of the method are pointed out in the end, which will be studied in the future.

INDEX TERMS Automated manufacturing system, deadlock prevention, resource-oriented Petri net (ROPN).

I. INTRODUCTION

An automated manufacturing system (AMS) consists of various manufacturing resources such as machines, automated guided vehicles, robots, and buffers. These resources are limited and shared by multiple production processes. In an AMS, parts are processed concurrently according to pre-determined sequences for each part type and they compete for the limited resources, leading to potential deadlocks [1]. Deadlocks are highly undesirable in AMSs since the running of a system would be blocked when a deadlock occurs, resulting in economic loss and even though catastrophic consequences.

An AMS is considered as a typical discrete-event system and Petri nets are powerful in modeling and analyzing such a system [2]–[12] as well as opacity verification [13], supervisory control [14]–[16], and scheduling [17] of discrete-event systems. Four necessary conditions for the occurrence of deadlocks in a resource allocation system (RAS) are established by Coffman et al. [18]. Among them, only the last one, called "circular wait", stems from system resource request, allocation, and release, which can be prevented by controlling system resource allocation. Great attention has been paid to the deadlock resolution issue since 1990's [19]–[26].

One of the widely used deadlock resolution strategies is deadlock prevention. Two kinds of technologies are often employed to prevent deadlocks in AMSs modeled by Petri nets. One is reachability analysis, by which one can obtain a maximally permissive policy. However, it suffers from state explosion problem, since the number of reachable states grows exponentially with the size of the model. The other is structural analysis, which is usually much more efficient in computational complexity than the former. However, with such a type of methods, generally one cannot obtain a max-

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

imally permissive policy. This work deals with deadlock issues by structural analysis.

This study focuses on deadlock prevention problems for the disjunctive and strict single-type AMS (DS-AMS) [27]. A disjunctive (DIS-)AMS means that the execution logic of a part type may possess routing flexibility. That is to say, a part type may have selective processing routes, each of which is acyclic and connects a "source" to a "sink" processing stage. A strict single-type (SST-) AMS means that a single resource type does not support two successive processing stages and a processing stage of a part just requires a single resource type. However, one processing stage of a part might engage more than one unit of the required single resource type. A disjunctive and strict single-type (DS-) AMS possesses characteristics of both disjunction and strict single-type, which is usually described by weighted systems of simple sequential processes with resources (WS^3PR) [28]–[30], or ES^3PR [31], or a larger range taxonomy of Petri nets, such as S^4PR [32], [33], S^3PGR^2 [34].

Siphon is a special structure that may result in deadlocks [11], [33], [35]. Based on a structural analysis on WS^3PR, the concept of *max*-controlled* siphon is proposed in [28] to obtain a necessary and sufficient liveness condition. Circular wait is identified in [29] and [31] to reveal the relationship between deadlocks and blocked markings on resource-transition circuits. These achievements are significant in theory. However, their implementation is impracticable owing to their highly computational cost. In [30], a method to check the liveness of a WS^3PR is presented with no technique being given to make a non-live system live. An iterative method implemented by means of integer linear programming formulations is presented in [32] to forbid deadlock states and synthesize a controller to ensure the final live behavior of the system. The work in [34] develops a polynomial-complexity deadlock avoidance policy characterized as resource upstream neighborhood (RUN). The aforementioned policies are either computationally inefficient or too restrictive. Hence, it is necessary to develop new approaches to the deadlock resolution issues for DS-RASs such that it is more computationally efficient and less restrictive.

A resource-oriented Petri net (ROPN) modeling method is initiated in [36] and [37] to model the behaviors of single-unit AMSs [27]. Then, ROPNs are successfully applied for deadlock avoidance [38]–[48] and complex production scheduling problems [49]–[56] due to their good compactness, powerful modeling capability, and structural properties. By using ROPN modeling for AMSs, Wu *et al.* [57] develop a liveness verification procedure through cutting knots one by one in an ROPN model. They also develop an optimal deadlock control policy with polynomial complexity by a one-step-look-ahead method for a system where the capacity of a shared place chain in the ROPN is at least two.

This work aims to develop a liveness-enforcing policy for DS-AMS. Instead of using WS^3PR, or ES^3PR, or S^4PR etc., it employs ROPN to model such a system by taking the advantages of the structural properties of ROPNs. In an ROPN, a deadlock must occur in a strongly connected subnet if it is possible. Thus, we need to control the individual strongly connected subnets in an ROPN such that deadlock never occurs. Based on this observation, we identify the transitions in a strongly connected subnet whose firing can leads to a deadlock. Then, the subnet is controlled such that the firing of these transitions never leads to a deadlock. By doing so, an efficient approach is proposed. The main contributions are summarized as the following aspects.

1) A new modeling method for DS-AMS by using ROPN is proposed in this paper for the first time.
2) A sufficient and necessary condition for liveness of AMSs modeled by an ROPN is established, which is more succinct than that in the existing studies.
3) An efficient approach to liveness-enforcing of an ROPN is developed for deadlock resolution by controlling a small number of transitions, which is of polynomial complexity and not too conservative.

The next section briefly recalls the basic concepts and notation related to PNs, multi-sets, and theory on digraphs. In Section III, the procedure of constructing an ROPN model for a DS-AMS is presented first, then we also expatiate the operational rules of the ROPN model and crystallize several definitions related to deadlock issues. Section IV develops the sufficient and necessary condition for liveness of an ROPN model, which forms the foundation for developing an efficient policy to prevent deadlocks in the ROPN model. Based on this condition, the liveness-enforcing approach is presented. Two frequently-employed examples are tested by the proposed policy to show the efficiency and effectiveness in Section V. Section VI concludes this work and points out some issues that should be studied in the future.

II. PRELIMINARIES

Let $\mathbb{N} = \{0, 1, 2, \ldots\}$ be the set of non-negative integers and $\mathbb{N}_s = \{1, 2, \ldots, s\}$ be the set of positive integers that are not greater than s ($s \in \mathbb{N}$), respectively. The operator $|*|$ obtains the cardinality of a set "$*$". The k-Cartesian products of $\mathbb{N}$ is denoted as $\mathbb{N}^k$, $k \in \mathbb{N}$.

A. PETRI NET

A place/transition Petri net (PN), depicted by a directed bipartite graph, is defined as a triple $\mathcal{N} = (P, T, W)$, with P and T being two disjoint, finite, and non-empty sets of places and transitions, respectively. $W : (P \times T) \cup (T \times P) \rightarrow \mathbb{N}$ specifies the flow relation. If W-value is non-zero, such a flow relation is depicted by a directed edge that link every nodal pair; if W-value equals 0, no directed edge on which the function W is defined exists. All the edges form a set, denoted by F. A marked PN is denoted as $(\mathcal{N}, M)$ when a marking (or state) M is associated with it. $M(p)$ presents the number of tokens in p at M and usually M_0 is used to denote the initial marking. The pre-set and post-set of t (p) are denoted by ${}^\bullet t$ (${}^\bullet p$) and $t^\bullet$ ($p^\bullet$), respectively, where ${}^\bullet t = \{p|(p, t) \in F\}$, $t^\bullet = \{p|(t, p) \in F\}$, ${}^\bullet p = \{t|(t, p) \in F\}$, and $p^\bullet = \{t|(p, t) \in F\}$. More details can be found in [58]. By introducing more elements to a place/transition PN, some extended PN can be obtained. For example, by introducing colors to discriminate tokens, we have a colored PN [59]; by specifying the capacity of places, we get a finite capacity PN [60]; by associating probabilities with transitions, we have a stochastic PN [61].

B. MULTI-SET

A multi-set is an extension of a normal set in mathematics. Each member in a normal set has only one membership, while a member in a multi-set can have more than one memberships. Metaphorically speaking, a multi-set can be considered as a bag containing multiple instances of a member. It becomes a normal set if the multiplicity of every member is unitary. An empty multi-set is denoted by ϵ. Multi-sets are employed to describe a marking of PNs. For example, a non-empty marking $M = \sum_{i=1}^{l} a_i p_i$ is a multi-set with $p_i \in P$, $a_i \in \mathbb{N}$, $i \in \mathbb{N}_l$, $l \in \mathbb{N}\backslash\{0\}$ and P is the underlying set in which M lies. The support of M, denoted by $||M||$, comprises $P' \subseteq P$ s.t. for any $p_i \in P'$, $a_i \neq 0$ and for any $p_j \in P\backslash P'$, $a_j = 0$. We use $[*]$ appended to a multi-set to acquire the coefficient corresponding to " $*$ ". Thus, $M[p_i]$, which equals a_i numerically, represents the coefficient of p_i in above multi-set M. Define $|M| = \sum_{i=1}^{l} a_i$, $i \in \mathbb{N}_l$, as the cardinality of M. Let $\alpha = \sum_{i=1}^{l} x_i p_i$ and $\beta = \sum_{i=1}^{l} y_i p_i$ be two multi-sets defined on an identical underlying set. α is covered by β, or β covers α, denoted by $\alpha \leq \beta$ or $\beta \geq \alpha$, if $y_i \geq x_i, \forall i \in \mathbb{N}_l$. Define $\alpha + \beta = \sum_{i=1}^{l}(x_i + y_i)p_i$ and further $\alpha - \beta = \sum_{i=1}^{l}(x_i - y_i)p_i$ when $\alpha \geq \beta$. The inner product operation is denoted by $\alpha \cdot \beta = \sum_{i=1}^{l}(x_i \times y_i)$, which is a scalar and not a multi-set any more. Given an underlying set $\mathcal{Q} = \{q_1, q_2, \ldots, q_l\}$, the set of all multi-sets defined over $\mathcal{Q}$ is represented by $\mathfrak{M}_{\mathcal{Q}}$, i.e., $\mathfrak{M}_{\mathcal{Q}} = \{\sum_{i=1}^{l} x_i q_i | x_i \in \mathbb{N}, q_i \in \mathcal{Q}\}$. For more details, the readers can refer to [62].

C. GRAPH THEORY REVIEW

Let $G = (V, E)$ be a digraph with V and E being the sets of nodes and directed edges, respectively. Let $v = x_1 x_2 \ldots x_k$, $x_i \in V$, $i \in \mathbb{N}_k$ and $k \in \mathbb{N}\backslash\{0, 1\}$, be a string. Then, it is a directed path if for any $x \in \mathcal{U}(v)$[1] and for any $i \in \mathbb{N}_k\backslash\{1\}$, $\langle x_{i-1}, x_i\rangle$[2] $\in E$. If $x_1, x_2, x_3, \ldots, x_k$ on a directed path v are distinct except for the pair of x_1 and x_k, then v is called an elementary path. If $x_1 = x_k$ in an elementary path v, then v forms an elementary circuit. A graph $\tilde{G} = (\tilde{V}, \tilde{F})$ is a sub-graph of $G = (V, E)$ if $\tilde{V} \subseteq V$ and $\tilde{F} \subseteq E$, denoted by $\tilde{G} \subseteq G$. If the "if-condition" in the above two equations does not hold simultaneously, it is denoted as $\tilde{G} \subset G$. A sub-graph $\tilde{G}$ is said to be strongly connected if any two nodes are mutually reachable through a directed path in $\tilde{G}$. A strongly connected sub-graph (SCS) is composed of some elementary circuits among which a circuit shares at least one common node with at least another one.

III. ROPN FOR DS-AMS

[1]Function $\mathcal{U}(v)$ returns the set of all letters in string v.

[2]$\langle a, b\rangle$ denotes a directed edge via its node-pair a, b usually, while (a, b) represents an undirected edge.

A. ROPN

Consider a DS-AMS with m resources and s part types, assume that a part type X_i, $i \in \mathbb{N}_s$, has l_i processing stages.

Definition 1: An ROPN for modeling a DS-AMS is defined as $\mathcal{N} = (P \cup \{p_0\}, T, F, C, W, K, I, O)$, *where*

- $P = \{p_1, p_2, \ldots, p_m\}$ *is a set of places modeling the* m *resources types, i.e.,* p_i *models resource type* R_i*. The idle place* p_0 *models the LOAD/UNLOAD center and holds raw-tokens as many as possible.*
- $T = \{t_1, t_2, \ldots, t_n\}$ *is a finite set of transitions modeling the part delivery processes among resources.*
- $F \subseteq ((P \cup \{p_0\}) \times T) \cup (T \times (P \cup \{p_0\}))$ *is the set of directed arcs modeling the relationship of parts-flows among the resources in the DS-AMS.*
- $C = \{X_i^j | i \in \mathbb{N}_s, j \in \mathbb{N}_{l_i}, l_i \in \mathbb{N}\backslash\{0\}\}$ *is the set of colors modeling processing stages of loaded parts, where* X_i^j *denotes the processing stage, labeled by number j, of part type* X_i *and* l_i *is the total processing stages of part type* X_i*.*
- $W : P \rightarrow \mathfrak{M}_C$ *is a resource requirement function.* $W(p_i)$*,* $p_i \in P$*, is a multi-set mathematically describing resource requests of the potential processing stages of any part type supported by place* p_i*.*
- $K : P \cup \{p_0\} \rightarrow \mathbb{N}\backslash\{0\}$ *is a capacity function describing the units of a resource type, defined by strictly positive integers. It means that the maximal number of spaces in a place* p *usable for all possible parts is* $K(p)$ *at any states. Usually,* $K(p_0) = \infty$*.*
- I *(resp.* O*)* $: F \rightarrow C \cup \{X_i, i \in \mathbb{N}_s\}$ *is a function that assigns a color to an edge to determine the token flows in the net.* $I(p, t)$ *(resp.* $O(p, t)$*) models one token with the specific color* $I(p, t)$ *(resp.* $O(p, t)$*) taken from (resp. put into) a place* p *with respect to* t*.*[3]

All of the possible stages of any parts supported by p is denoted by $\langle p\rangle$. An AMS with parts being processed can be modeled by a marked ROPN $(\mathcal{N}, M)$, where $M : P \rightarrow \mathfrak{M}_C$ is a marking function. Usually, M_0 represents the initial marking. $M(p_i)$ is a multi-set lying on $\langle p_i\rangle$. $W(p_i)$ is also a multi-set lying on $\langle p_i\rangle$. Let M be a reachable marking. $\forall p_i \in P$, we have $W(p_i) \cdot M(p_i) \leq K(p_i)$. An example will be employed to interpret the detailed components of an ROPN in the next subsection.

B. MODELING DS-AMS VIA ROPN

The WS^{3}PR is a popular tool to analyze the behaviors of DS-AMS. It will be shown that an ROPN model is more compact than the WS^{3}PR model. In this study, we explore such issues via an ROPN. Algorithm 1 gives a procedure to construct an ROPN from a WS^{3}PR.

Example 1: It is from [63] with some variations on weights of edges and initial marking. The WS^3*PR* $\mathcal{PN}$ $=$

[3]$I(p, t)$ $(resp. O(p, t))$ is also considered as a single-unit multi-set, which is composed of one item only with coefficient one, i.e., a single-unit multi-set is taken to be identical to the single color attached to itself. For example, that $I(p, t) = X_i^k$ means that a part with color X_i^k is taken away from a place p while emanating a transition t.

Algorithm 1 Procedure of Modeling DS-AMS via ROPN

Input: A WS^3PR $\mathcal{PN} = (P^0 \cup P_A \cup P_R, T_1, F_1, W_1)$ with an initial marking M_0

// P^0, P_A and P_R are the set of idle, operation and resource places, respectively; and T_1, F_1 and W_1 are the set of transitions, arcs, and weights on arcs, respectively.

Output: An ROPN $\mathcal{N} = (P_2 \cup \{p_0\}, T_2, F_2, C, W, K, I, O)$

1: Delete $P_A \cup P^0$ as well as their connected arcs from the given WS^3PR $\mathcal{PN} = (P^0 \cup P_A \cup P_R, T_1, F_1, W_1)$ and obtain the residual net $\mathcal{PN}_1$;

2: Reverse arrowhead of each arcs in $\mathcal{PN}_1$ and obtain a digraph $G' = (P_2 \cup T_2, F_2)$, where $P_2 = P_R$, $T_2 = T_1$;

3: Create idle place p_0 and draw arcs from p_0 to each transition in G' which has no input place and arcs from each transition in G' which has no output place to p_0.

4: Calculate $W(p)$ for any $p \in P_2$. If $W_1(p, t_k) \neq 0$ for any $(p, t_k) \in F_1$ and any $p \in P_R$ in $\mathcal{PN}$, set $W(p)[X_i^j] := W_1(p, t_k)$;

// Assume that processing stage X_i^j should consume $W_1(p, t_k)$ resources of p.

5: Create set $\{X_i | i \in \mathbb{N}_{|P^0|}\}$ indicating the set of part types to be processed in the AMS and $C := \{X_i^j | i \in \mathbb{N}_{|P^0|}, j \in \mathbb{N}_{l_i}\}$ with l_i being the number of processing stages of part type X_i;

6: Set $K(p) = M_0(p)$ for any $p \in P_2$;

7: Calculate I and O and obtain ROPN $\mathcal{N} = (P_2 \cup \{p_0\}, T_2, F_2, C, W, K, I, O)$;

//I and O are pre-determined by processing routes of part types

TABLE 1. The input and output relation of the ROPN model in Figure 3.

$W(p_{10}) = X_1^1$ $K(p_{10}) = 1$	$W(p_2) = X_1^5$ $K(p_2) = 1$	$W(p_{11}) =$ $2X_1^2 + X_2^3$
$W(p_{12}) =$ $2X_1^3 + 3X_2^2$	$W(p_{13}) =$ $X_1^4 + X_2^1$	$I(p_{10}, t_2) = X_1^1$ $K(p_{13}) = 1$
$O(p_{10}, t_1) = X_1^1$ $I(p_{10}, t_4) = X_1^1$	$O(p_{11}, t_4) = X_1^2$ $I(p_{11}, t_5) = X_1^2$	$O(p_{12}, t_5) = X_1^3$ $I(p_{12}, t_6) = X_1^3$
$O(p_{13}, t_6) = X_1^4$ $I(p_{13}, t_7) = X_1^4$	$O(p_2, t_2) = X_1^5$ $I(p_2, t_3) = X_1^5$	$O(p_{13}, t_{11}) = X_2^1$ $I(p_{13}, t_8) = X_2^1$
$O(p_{12}, t_8) = X_2^2$ $I(p_{12}, t_9) = X_2^2$	$O(p_{11}, t_9) = X_2^3$ $I(p_{11}, t_{10}) = X_2^3$	$K(p_{11}) = 3$ $K(p_{12}) = 5$

$(P^0 \cup P_A \cup P_R, T_1, F_1, W_1)$ modeling DS-AMS is shown in Figure 1, where $P^0 = \{p_0, p_{14}\}$ and $P_R = \{p_{10} - p_{13}\}$, the other places belong to P_A, and $T_1 = \{t_1, t_2, \cdots, t_{11}\}$. The processing sequence of part type X_1 with two possible routes is modeled by p_0 t_1 p_1 t_2 p_3 t_3 p_0 and p_0 t_1 p_1 t_4 p_4 t_5 p_5 t_6 p_6 t_7 p_0, and that for part type X_2 is $p_{14}t_{11}p_9$ t_8 p_8 t_9 p_7 $t_{10}p_{14}$. $M_0(p_0) = M_0(p_{14}) = 10$, $M_0(p_2) = M_0(p_{10}) = M_0(p_{13}) = 1$, $M_0(p_{11}) = 3$, $M_0(p_{12}) = 5$, and $M_0(p) = 0, \forall p \in P_A$. $W_1(t_5, p_{11}) = W_1(p_{11}, t_4) = W_1(p_{12}, t_5) = W_1(t_6, p_{12}) = 2$, $W_1(t_9, p_{12}) = W_1(p_{12}, t_8) = 3$ and the weights on the other arcs are one.

Applying Algorithm 1 to Example 1, we obtain the corresponding ROPN model as shown in Figure 2. The processing route of part type X_1 is $p_0t_1p_{10}t_2p_2t_3p_0$ and $p_0t_1p_{10}t_4p_{11}t_5p_{12}t_6p_{13}t_7p_0$; and the one for part type X_2 is

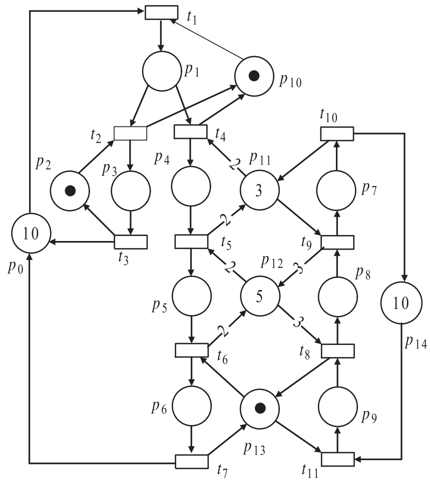


FIGURE 1. The WS^3PR model for an AMS of Example 1.

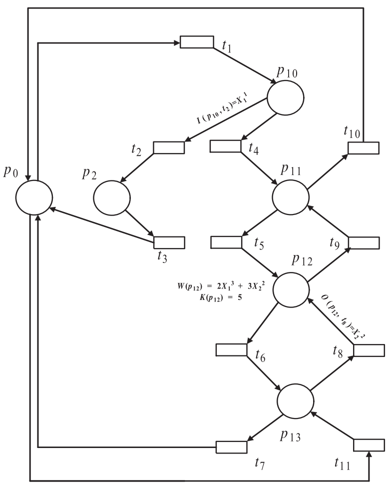


FIGURE 2. The ROPN of Example 1.

$p_0t_{11}p_{13}t_8p_{12}t_9p_{11}t_{10}p_0$. The other components of the ROPN, K, I, O and W, are listed in Table 1. $\langle p_{12} \rangle = \{X_1^3, X_2^2\}$. For any reachable marking M, we have $M(p_{12}) \cdot W(p_{12}) \leq K(p_{12}) = 5$. Notice that the idle place p_0 has infinite capacity and does not cause any deadlock, it can be removed to obtain a reduced ROPN as shown in Figure 3. The reduced ROPN is also named ROPN for simplicity. When an ROPN is mentioned, it sometimes means a directed graph without confusion. Thus, to focus the structure of an ROPN, we represent it as $\mathcal{N} = (P \cup T, F)$. There is no loop between a couple of transition t and place p in the ROPN due to the "strict" property of the considered AMS, i.e., there is no pair t and p, s.t. ${}^\bullet t = t^\bullet = \{p\}$. Furthermore, we have $|{}^\bullet t| = |t^\bullet| = 1$.

C. EVOLUTION RULES OF AN ROPN

Definition 2: A transition t in an ROPN is enabled at marking M if the following two conditions are met:

$$M(p) \geq I(p, t), \quad \forall p \in {}^\bullet t \tag{1}$$

and

$$K(p) \geq (M(p) + O(p, t)) \cdot W(p), \quad \forall p \in t^\bullet \tag{2}$$

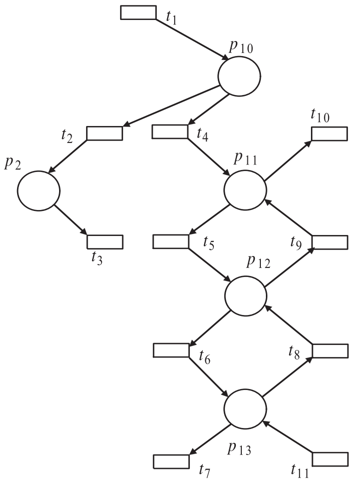


FIGURE 3. The reduced ROPN of Example 1.

When an enabled transition $t \in T$ in an ROPN at M fires, it yields a marking M' according to

$$M'(p) = \begin{cases} M(p) - I(p,t), & p \in {}^\bullet t; \\ M(p) + O(p,t), & p \in t^\bullet; \\ M(p), & p \notin {}^\bullet t \cup t^\bullet. \end{cases} \quad (3)$$

Definition 2 indicates that t is enabled and can fire if there are enough tokens in ${}^\bullet t$ with respect to $I(p,t)$ and enough free spaces in $t^\bullet$ at M. When Condition (1) is met, t is said to be pre-enabled; while Condition (2) holds, t is post-enabled. Thus, t is enabled if it is both pre- and post-enabled, which is denoted as $M[t\rangle$; otherwise, it is denoted as $\neg M[t\rangle$. An enabled transition can fire. $M[t\rangle M'$ means M' is reached from M after firing t or a one-step evolution. Let σ be a string with $\mathcal{U}(\sigma) \subseteq T$. The firing of transitions in σ at marking M successively results in marking M', which is denoted by $M[\sigma\rangle M'$. In a reduced ROPN, a transition t is said to be a source (resp. sink) one if ${}^\bullet t = \varnothing$ (resp. $t^\bullet = \varnothing$). We take it by default that the source (resp. sink) transitions are always pre- (resp. post-) enabled. Firing a source (resp. sink) transition models loading (resp. unloading) a part into (resp. from) the AMS. The set of source (resp. sink) transitions in T is denoted by $\overline{T}$ (resp. $\underline{T}$).

In the above mentioned ROPN in Figure 3, $\overline{T} = \{t_1, t_{11}\}$ and $\underline{T} = \{t_3, t_7, t_{10}\}$. Suppose that $M(p_{10}) = X_1^1$, $M(p_{11}) = X_1^2$, $M(p_{12}) = X_1^3 + X_2^2$ and $M(p_2) = M(p_{13}) = \epsilon$. By Conditions (1) and (2) in Definition 2, t_2, t_6 and t_9 are enabled, while t_4 and t_5 are pre-enabled but not post-enabled. Specifically, $M(p_{12})+O(p_{12}, t_5) = 2X_1^3+X_2^2$ and $(M(p_{12})+O(p_{12}, t_5)) \cdot W(p_{12}) = 7 > K(p_{12}) = 5$. Thus, t_5 is not post-enabled.

D. TERMINOLOGIES RELATED TO LIVENESS OF AN ROPN

Let $R(\mathcal{N}, M)$ be the set of reachable markings from M for ROPN $\mathcal{N}$ and we write $R(\mathcal{N}, M_0)$ by $R(\mathcal{N})$ in short. Also, let $PR(\mathcal{N})$ be the set of potential markings due to the capacity restriction of each place in P of $\mathcal{N}$ only. Thus, $R(\mathcal{N}) \subseteq PR(\mathcal{N})$. A marking $M \in R(\mathcal{N})$ is safe if $M_0 \in R(\mathcal{N}, M)$; otherwise, it is unsafe. A transition $t \in T$ in $\mathcal{N}$ is live iff for any $M \in R(\mathcal{N})$, there exists $M' \in R(\mathcal{N}, M)$, s.t. $M'[t\rangle$; otherwise it is non-live. A transition $t \in T$ is dead at $M \in R(\mathcal{N})$ if there is no $M' \in R(\mathcal{N}, M)$, $s.t.$ $M'[t\rangle$. $\mathcal{N}$ is live iff for any $t \in T$, t is live. It is deadlock-free if for any $M \in R(\mathcal{N})$, there is $t \in T$, $M[t\rangle$. Liveness implies deadlock-freeness, but not vice-versa. An ROPN $\mathcal{N}$ is said to be well configured if for any $t \in T$, there exists $M' \in R(\mathcal{N})$, s.t. $M'[t\rangle$. In the sequel, we consider well configured ROPNs only and aim to enforce the liveness of such ROPNs.

A partial deadlock in an AMS is a state at which a set of processes is entangled in a circular waiting pattern with each process in this set requiring, in order to advance to its next processing stage, some resource units that are held by other processes in the set [27], [64]. Such situations can be described by means of ROPN properties precisely. Furthermore, markings which are reachable under an enforced supervisory control policy and from which progress is inhibited by the policy-imposed constraints and not by the AMS structure, are called (policy-) induced deadlocks [27]. A deadlock control policy is correct if and only if for every marking admitted by the policy, there is a policy-admissible transition whose firing does not load a new job into the system. The following definitions are necessary to deduce deadlock resolution policies.

Definition 3: Let $\mathcal{N}$ be an ROPN. Transitions $t_1, t_2 \in T$ are said to be brothers if $I(p, t_1) = I(p, t_2) \neq \epsilon$, $p \in P$ and, in this case, t_1 (t_2) is the brother of t_2 (t_1). The set of all brothers of transition t in $\mathcal{N}$, including t itself, is denoted by $\mathcal{B}(t)$. Suppose that $\tilde{G}_{\mathcal{N}} = (\tilde{V}, \tilde{F}) \subseteq \mathcal{N}$ is a sub-net of $\mathcal{N}$. A transition $t \in \tilde{V}$ is said to be partitioned with respect to $\tilde{G}_{\mathcal{N}}$, if $\mathcal{B}(t) \backslash \tilde{V} \neq \varnothing$. The set of all partitioned transitions with respect to $\tilde{G}_{\mathcal{N}}$ is denoted by $\mathcal{P}a(\tilde{G}_{\mathcal{N}})$.

In Figure 3 of Example 1, we have $\mathcal{B}(t_2) = \mathcal{B}(t_4) = \{t_2, t_4\}$. Suppose that $\tilde{G}_{\mathcal{N}} = (\tilde{V}, \tilde{F}) \subseteq \mathcal{N}$ such that $t_2 \notin \tilde{V}$ and $t_4 \in \tilde{V}$, then $t_4 \in \mathcal{P}a(\tilde{G}_{\mathcal{N}})$.

Definition 4: A sub-net $\tilde{G}_{\mathcal{N}} = (\tilde{V}, \tilde{F}) \subseteq \mathcal{N}$ is said to be complete if any transition $t \in \tilde{V}$, $\mathcal{B}(t) \subseteq \tilde{V}$ holds. The set of all strongly connected and complete sub-nets (SCCSs) in $\mathcal{N}$ is denoted by $\mathcal{G}_{\mathcal{N}}^{C}$.

A circuit without p_0 in an ROPN corresponds to an resource-transition circuit (RTC) which is originated in [11] according to the procedure of the ROPN construction in Algorithm 1. An SCCS in an ROPN corresponds to a structure named perfect resource-transition circuit (PRTC) in WS^{3}PR. Both of them are closely related to deadlocks raised in Petri net evolution [8], [9], [11], [39], [45]. However, there is some discrimination between SCCSs and PRTCs. The orientation of an edge in an SCCS is reversed from the corresponding edge that connects the same nodes in WS^{3}PR, which depicts the part-flow between various resources expediently. We can use the SCCS itself to make structural analysis on deadlock issue without resorting to extra components outside the circuit. However, operation places that are not on any PRTC and consume tokens in resource places need to take into consideration.

Definition 5: The available capacity of a place p at marking M is the number of free spaces in p, denoted as $V_M(p) = K(p) - M(p) \cdot W(p)$.

Definition 6: Let $\tilde{G}_{\mathcal{N}} = (\tilde{V}, \tilde{F}) \subseteq \mathcal{N}$ *be a sub-net and M be a marking on* $\mathcal{N}$. *The projection of marking M on* $\tilde{G}_{\mathcal{N}}$, *denoted by* $M\uparrow_{\tilde{G}_{\mathcal{N}}}$, *is defined as:*

$$M\uparrow_{\tilde{G}_{\mathcal{N}}}(p)[X_j^k] = \begin{cases} M(p)[X_j^k], & p \in \tilde{V}, \exists t \in \tilde{V}, \text{ s.t. } \\ & \mathcal{B}(t) \subseteq \tilde{V} \bigwedge I(p,t) = X_j^k; \\ 0, & \text{otherwise;} \end{cases} \quad (4)$$

In Figure 3, suppose that $M(p_{10}) = X_1^1$, $M(p_{11}) = X_1^2$, $M(p_{12}) = X_1^3 + X_2^2$ and $M(p_2) = M(p_{13}) = \epsilon$. Let $\tilde{G}_{\mathcal{N}} = (\tilde{V}, \tilde{F})$ be a sub-net with $\tilde{V} = (p_{11}, p_{12}, t_5, t_9)$ and $\tilde{F} = (\langle p_{11}, t_5\rangle, \langle t_5, p_{12}\rangle, \langle p_{12}, t_9\rangle, \langle t_9, p_{11}\rangle)$ being sets of nodes and directed edges, respectively. Then, $V_M(p_{11}) = K(p_{11}) - M(p_{11}) \cdot W(p_{11}) = 3 - 1 \times 2 = 1$ and

$$M\uparrow_{\tilde{G}_{\mathcal{N}}}(p) = \begin{cases} X_1^2, & p = p_{11} \\ X_2^2, & p = p_{12} \\ \epsilon, & \text{otherwise} \end{cases} \quad (5)$$

Definition 7: Let $\tilde{G}_{\mathcal{N}} = (\tilde{V}, \tilde{F}) \subseteq \mathcal{N}$ *be a sub-net of a marked ROPN. The potentially available capacity of a place* $p \in \tilde{V}$ *at marking M with respect to* $\tilde{G}_{\mathcal{N}}$ *are defined as* $V_M^\circ(p)_{\tilde{G}_{\mathcal{N}}} = K(p) - M\uparrow_{\tilde{G}_{\mathcal{N}}}(p) \cdot W(p)$. *A transition* $t \in \tilde{V}$ *is said to be potentially post-enabled at M if* $O(p,t) \cdot W(p) \leq V_M^\circ(p)_{\tilde{G}_{\mathcal{N}}}$

For the above marking M and SCCS $\tilde{G}_{\mathcal{N}}$, $V_M^\circ(p_{12})_{\tilde{G}_{\mathcal{N}}} = 5 - 1 \times 3 = 2$ while $V_M(p_{12}) = 0$. t_5 is potentially post-enabled although it is not post-enabled at M. From an ROPN dynamic standpoint, a formal definition of partial deadlock in a DS-AMS is as follows.

Definition 8: $\mathcal{N}$ *is in a partial deadlock at marking M if there exists an SCCS* $\tilde{G}_{\mathcal{N}} = (\tilde{V}, \tilde{F}) \subseteq \mathcal{N}$ *s.t.* $\forall t \in \tilde{V}$, *t is pre-enabled but not potentially post-enabled at M.*

The occurrence of partial deadlock means that each of the transitions in a specific SCCS cannot be enabled in the possible successive states, which corresponds to the saturation of a PRTC in WS^{3}R [8], [9], [11].

Definition 9: Let $\tilde{G}_{\mathcal{N}} = (\tilde{V}, \tilde{F}) \subseteq \mathcal{N}$. *A transition* $t \in \tilde{V}$ *is said to be non-blocked with respect to* $\tilde{G}_{\mathcal{N}}$ *if* $O(p,t) \cdot W(p) \leq min_{M \in R(\mathcal{N})}\{V_M^\circ(p)_{\tilde{G}_{\mathcal{N}}}\}$. *The set of non-blocked transitions with respect to* $\tilde{G}_{\mathcal{N}}$ *is denoted as* $\mathcal{N}b(\tilde{G}_{\mathcal{N}})$.

For the above mentioned SCCS $\tilde{G}_{\mathcal{N}}$ in Figure 3, $\mathcal{N}b(\tilde{G}_{\mathcal{N}}) = \{t_5, t_9\}$. Transition t_5 is non-blocked since $O(p_{12}, t_5) \cdot W(p_{12}) = 2$ and $min_{M \in R(\mathcal{N})}\{V_M^\circ(p_{12})_{\tilde{G}_{\mathcal{N}}}\} = 2$, so is t_9.

Definition 10: Transition $t_2 \in T\backslash\overline{T}$ *is a successor of transition* $t_1 \in T\backslash\underline{T}$ *if there is a place* $p \in P$, *s.t.* $O(p, t_1) = I(p, t_2) \neq \epsilon$. *The set of successors of t is denoted as* $\mathcal{S}u(t)$. *Transition* $t_2 \in T\backslash\underline{T}$ *is a predecessor of a transition* $t_1 \in T\backslash\overline{T}$ *if there is a place* $p \in P$, *s.t.* $O(p, t_2) = I(p, t_1) \neq \epsilon$. *The set of predecessors of t is denoted as* $\mathcal{P}r(t)$.

In the ROPN shown in Figure 3, we have $\mathcal{S}u(t_1) = \{t_2, t_4\}$ and $\mathcal{P}r(t_4) = \{t_1\}$.

IV. LIVENESS-ENFORCING FOR ROPN

A. MAIN RESULT

To make automated operations of a DS-AMS possible, the liveness of its ROPN model should be guaranteed. Also, to make it run efficiently, a liveness-enforcing policy should not be too restrictive; otherwise it would greatly reduce the utilization of the limited resources. For example, extremely limiting the number of parts released into a considered AMS as less as possible does not cause any deadlocks. However, it can lead to a long resource idle time and low productivity. Ideally, a more permissive policy is pursued. Thus, we assume that the raw parts can enter into the system and access their required resources for the first processing stage as long as they are admitted by the enforcing policy. The following theorem paves an efficient way to optimize the behaviors of an ROPN. Although a similar result was presented by [29, Th. 2], the following theorem is obtained in a much more succinct way.

Theorem 1: An ROPN $\mathcal{N}$ *is live iff* $\forall M \in R(\mathcal{N})$ *and* $\forall \tilde{G}_{\mathcal{N}} \in \mathcal{G}_{\mathcal{N}}^C$, *no partial deadlock occurs in* $\tilde{G}_{\mathcal{N}}$ *at M.*

Proof: (⇒): It is trivial by the definition of liveness of an ROPN.

(⇐): We show it by contradiction. Assume that the ROPN is non-live. Then, there exist a $t \in T$ and $M \in R(\mathcal{N})$, s.t. t is dead at M. M must be an unsafe marking. By [27, Sec. 2.2, Proposition 2], we conclude that, an unsafe marking on the ROPN for a DS-AMS must evolve into a partial deadlock inevitably without firing transitions in $\overline{T}$. By Definition 8, there is an SCCS $\tilde{G}_{\mathcal{N}}^* \in \mathcal{G}_{\mathcal{N}}^C$ and a marking $M' \in R(\mathcal{N}, M)$, s.t. $\tilde{G}_{\mathcal{N}}^*$ is partially deadlocked at M'. It is a contradiction. ■

Theorem 1 is significant to ensure the liveness of an ROPN in theory. It fully characterizes all partial deadlocks. However, it is intractable to verify the liveness of an ROPN via this theorem. It is known that both $|R(\mathcal{N})|$ and $|\mathcal{G}_{\mathcal{N}}^C|$ are exponential with the size of the considered AMS. Checking the satisfiability of conditions in Theorem 1 by enumerating all the states is inefficient since the number of operations to do so is $|R(\mathcal{N})| \times |\mathcal{G}_{\mathcal{N}}^C|$. It has been proved that the safety problem of an AMS — given a well configured ROPN and a marking $M \in R(\mathcal{N})$ on it, whether there is a firing sequence of transitions brings the system to its initial marking M_0 — is NP-complete. On the other hand, if the system is examined to be non-live, no way is given to eliminate such partial deadlocks. Nevertheless, by Theorem 1, a partial deadlock must be occur in an SCCS. Then, we need to control the individual SCCSs, respectively. Based on this result, an efficient approach is developed next.

B. APPROACH FOR LIVENESS-ENFORCING OF ROPN

By Definition 8, a partial deadlock is formed if and only if the following two conditions hold: 1) there is an SCCS $\tilde{G}_{\mathcal{N}} = (\tilde{V}, \tilde{F})$ and 2) there is a reachable marking M on $\tilde{G}_{\mathcal{N}}$ such that for any $t \in \tilde{V}$, t is pre-enabled but not potentially post-enabled at this marking. An ROPN is determined by the processing sequences of the parts entangled in the ROPN. Thus, the SCCSs in it are irrevocable. However, we could prevent deadlock markings from being reached through imposing some firing restrictions on the ROPN. It is known that verifying $M \in R(\mathcal{N})$, for any given marking M, is NP-hard. We augment $R(\mathcal{N})$ to $PR(\mathcal{N})$. If a policy is taken to guarantee that, $\forall M \in PR(\mathcal{N})$, no partial deadlock occurs, the ROPN must be live. To do so, we control the SCCSs such that partial deadlocks never occur. In an SCCS, there are transitions such that if such a transition is pre-enabled but not potentially post-enabled, a partial deadlock may occur. If we can control an SCCS such that when such a transition is not potentially post-enabled, it is not pre-enabled either. Then, this SCCS does not involves deadlock. Based on this observation, we need to identify such transitions in an SCCS and control them. Thus, Algorithm 2 is developed to identify such transitions. To better understand the logic of Algorithm 2, we present the following remarks.

Remark 1: For Statement 15 in Algorithm 2, we need to calculate $\mathcal{P}a(G)$ and $\mathcal{N}b(G)$. $\mathcal{P}a(G)$ is obtained by examining $t \in T(G)$ one by one. An integer linear program (ILP) is employed to examine whether $t \in \mathcal{N}b(G)$, which is given by Algorithm 3.

In Algorithm 3, $||I(p, t')||$ and $||O(p, t)||$ are the supports of $I(p, t')$ and $O(p, t)$, respectively, which are associated with a single color. $W(p)$ is a multi-set and $W(p)[*]$ represents the coefficient corresponding to Color " $*$ ". It is known that there is no efficient way to solve an ILP problem. In practice, the following properties are extraordinarily useful. For the sake of space saving, their proofs are omitted.

- Property 1. Let $G = (P(G) \cup T(G), E)$ be an SCCS and $t \in T(G)$ be a transition. If $\exists t' \in T(G)$, s.t. $O(p, t) = I(p, t') \neq \epsilon$, then $t \notin \mathcal{N}b(G)$.
- Property 2. Let $G = (P(G) \cup T(G), E)$ be an SCCS and $p \in P(G)$ be a place. If $\exists t' \in T(G)$, s.t. $W(p)[||I(p, t')||] = 1$, then for any $t \in {}^{\bullet}p$, $t \notin \mathcal{N}b(G)$.
- Property 3. Let $G = (P(G) \cup T(G), E)$ be an SCCS and $t \in T(G)$ be a transition. Suppose that $t^{\bullet} = \{p\}$. If $W(p)[||O(p, t)||] \geq min_{t' \in p^{\bullet} \cap T(G)} W(p)[||I(p, t')||]$, then $t \notin \mathcal{N}b(G)$.

Remark 2: For Statements 20 and 35 in Algorithm 2, we should select some specific transition $t \in T(G)$ carefully and design a proper procedure "$\mathbb{MKL}(TS)$". That is to say, the designed procedure ensures that, for any $t \in TS$, if it is not post-enabled at a reachable marking, then it is forced to be not pre-enabled either at that marking. Thus, it is always live. At the same time, this procedure does not result in induced deadlock. This issue will be discussed in the sequel. When a t in TS is controlled to be live, t can be removed from the considered SCCS in examining the liveness of the SCCS. Then, in the next iteration, we need to examine the liveness of the sub-SCCS of the remaining part of the SCCS after removing t. For Statements 25 in Algorithm 2, $\mathbb{SCC}(G)$ is implemented via Depth-First-Search [65] with polynomial complexity.

Algorithm 2 Policy to Ensure the Liveness of an ROPN

Input: An ROPN $\mathcal{N} = (P \cup T, F)$ for a DS-AMS.
Output: A live ROPN $\mathcal{N}_L$.

1: $G_0 \Leftarrow \mathcal{N}$;
2: **while** $(\exists x \in P \cup T, |{}^{\bullet}x| \times |x^{\bullet}| = 0)$ **do**
3: remove x from $P \cup T$ in G_0;
4: **end while**
5: *// make G_0 strongly connected*
6: *// an edge is deleted with deletion of either its nodes*
7: $\Psi \Leftarrow \{G_0\}$, $TS \Leftarrow \emptyset$, $TB \Leftarrow \emptyset$;
8: **while** $\Psi \neq \emptyset$ **do**
9: $G \Leftarrow Get(\Psi)$; *// $Get(\Psi)$ returns an element of the set Ψ arbitrarily*
10: $\Psi \Leftarrow \Psi \backslash \{G\}$;
11: **if** $|P(G) \cup T(G)| < 4$ **then**
12: *//deadlock occurs in a digraph with at least 4 nodes*
13: continue;
14: **else**
15: $TE \Leftarrow \mathcal{P}a(G) \cup \mathcal{N}b(G)$;
16: **if** $TE \neq \emptyset$ **then**
17: $T(G) \Leftarrow T(G) \backslash TE$, $TB \Leftarrow TB \cup TE$;
18: goto 25;
19: **end if**
20: $t \Leftarrow Select(T(G))$;
21: *//$Select(T(G))$ selects some specific transition t*
22: $T(G) \Leftarrow T(G) \backslash \{t\}$;
23: $TS \Leftarrow TS \cup \{t\}$;
24: **end if**
25: $\Phi \Leftarrow \mathbb{SCC}(G)$;
26: *// $\mathbb{SCC}()$ returns the set of strongly connected components in G*
27: $\Psi \Leftarrow \Psi \cup \Phi$;
28: **end while**
29: **if** $TS = \emptyset$ **then**
30: *print the AMS is live*;
31: $\mathcal{N}_L \Leftarrow \mathcal{N}$;
32: goto 37;
33: **else**
34: *print the AMS is not live*;
35: $\mathbb{MKL}(TS)$ and obtain $\mathcal{N}_L$; *// $\mathbb{MKL}(TS)$ forces each transition in TS of $\mathcal{N}$ live*
36: **end if**
37: return $\mathcal{N}_L$;

Remark 3: In each "while-loop" of Algorithm 2, at least one transition is removed. Thus, Algorithm 2 can be terminated within $|T|$ times of iterations at most.

We begin the technical discussion of the following developments, by providing a constraint for every transition selected by Function $Select()$ in Algorithm 2.

Let TS be the set of transitions selected by Function $Select()$ in Algorithm 2. $\forall t_i \in TS$, suppose that $\{p_{i_1}\} = {}^{\bullet}t_i$, $\{p_{i_2}\} = t_i^{\bullet}$, $I(p_{i_1}, t_i) = X_{j_i}^{k_i}$, $O(p_{i_2}, t_i) = X_{j_i}^{l_i}$, $\langle p_{i_2} \rangle = \{X_{j_i}^{l_i}, X_{q_1}^{h_1}, X_{q_2}^{h_2}, \ldots, X_{q_c}^{h_c}\}$, $W(p_{i_1})[X_{j_i}^{k_i}] = w_{j_i}^{k_i}$ and $W(p_{i_2}) =$

Algorithm 3 Examination of If $t \in \mathcal{N}b(G)$

Input: An SCCS $G \subset \mathcal{N} = (P \cup T, F)$, $t \in T(G)$.
Output: *True* or *False*. // *if* $t \in \mathcal{N}b(G)$, *return True*

1: set $t^{\bullet} = \{p\}$, $T_p = \{t' \in T(G) | t' \in p^{\bullet}\}$;
2: solve the following ILP:

$$\min\ V = K(p) - \sum_{t' \in T_p} c_{t'} W(p)[||I(p,t')||]$$
$$\text{s.t.} \sum_{t' \in T_p} c_{t'} W(p)[||I(p,t')||] \le K(p), \quad \{p\} = t^{\bullet}$$
$$c_{t'} \in \mathbb{N}\ \forall t' \in T_p \tag{6}$$

3: **if** the optimal solution $V^* \ge W(p)[||O(p,t)||]$ **then**
4: return *True*;
5: **else**
6: return *False*;
7: **end if**

$w_{ji}^{l_i} X_{ji}^{l_i} + w_{q1}^{h_1} X_{q1}^{h_1} + w_{q2}^{h_2} X_{q2}^{h_2} + \ldots, + w_{qc}^{h_c} X_{qc}^{h_c}$. The following Constraint related to t_i is constructed.

$$w_{ji}^{l_i} M(p_{i_1})[X_{ji}^{k_i}] + w_{ji}^{l_i} M(p_{i_2})[X_{ji}^{l_i}] + w_{q1}^{h_1} M(p_{i_2})[X_{q1}^{h_1}] + w_{q2}^{h_2} M(p_{i_2})[X_{q2}^{h_2}] + \ldots, + w_{qc}^{h_c} M(p_{i_2})[X_{qc}^{h_c}] \le C(p_{i_2}) \tag{7}$$

To implement a deadlock prevention policy, $|TS|$ constraints in the form of (7) are formulated, which are called Constraints (7) related to TS. Note that every transition in TS must be in some SCCSs and it is these transitions that can make a system deadlocked. Thus, to prevent deadlocks is to control these transitions such that they cannot be dead, which is implemented by imposing Constraints (7) related to TS on the ROPN model. Then, we have the following result.

Theorem 2: Any SCCS involving partial deadlocks does not subsume $t_i \in TS$ if Constraint (7) related to t_i is imposed on the considered ROPN model.

Proof: We show it by contradiction. Assume that t_i belongs to an SCCS in a partial deadlock at marking M. Then, t_i is pre-enabled but not potentially post-enabled at M. However, since t_i is pre-enabled at marking M, we have $M(p_{i_1})[X_{ji}^{k_i}] \ge 1$. Constraint (7) implies that, $\forall t_i \in TS$, if there are $X_{ji}^{k_i}$'s in p_{i_1}, they would 'crash' into p_{i_2} since their spaces is revered in p_{i_2}. Thus, t_i must be post-enabled, which contradicts that t_i is not potentially post-enabled at M. ∎

By Theorem 2, it implies that, when Constraint (7) related to a transition in TS is imposed, then if it is pre-enabled, it must be post-enabled.

Theorem 3: Let TS be the set of transitions selected by Function Select() in Algorithm 2. Assume that the following two conditions hold:

(I) $\forall t_i \in TS$, $\mathcal{P}r(t_i) \subseteq \overline{T}$.
(II) $t_i^{\bullet} \cap t_j^{\bullet} = \varnothing$, $\forall t_i, t_j \in TS$.

Then, the policy, made up of the conjunction of Constraints (7) related to TS, does not result in any induced deadlocks.

Proof: By Algorithm 2, there is no SCCS involving partial deadlocks at any reachable marking $M (\neq M_0)$ admitted by Constraints (7) related to TS. That is to say, there exists $t \in T \backslash \overline{T}$, t is feasible at M. Since $t \notin \overline{T}$, firing t does not result in loading a new job into the system. It should be noticed that firing t takes one token from ${}^{\bullet}t$ and puts one token into $t^{\bullet}$. We prove that the resulting marking after firing t does not violate Constraints (7) related to TS by considering the following two cases:

Case 1: If there is an enabled $t_i \in TS$, the resulting marking after firing t_i at M does not violate Constraint (7) related to t_i since enough spaces in $t_i^{\bullet}$ are reserved for receiving tokens taken from ${}^{\bullet}t_i$ under the imposed constraint related to t_i. At the same time, firing t_i does not have an impact on Constraint (7) related to $t_j (j \neq i)$ by Condition (II). Thus, no induced deadlock exists.

Case 2: $\forall t_i \in TS$, t_i is not enabled at M. Then, any t_i is not pre-enabled at M according to the imposed Constraint (7) related to t_i. We deduce that there is an enabled $t \in T \backslash (TS \cup \overline{T})$. In what follows, we advance our technical arguments by three subcases:

- Subcase 2.1: If firing t at M does not modify any item of Constraint (7) related to each transition in TS, or only decreases one token for one item and does not increase one for any other item of Constraints (7) related to TS, the resulting marking after firing t still meets Constraints (7) related to TS. Thus, t's firing is not inhibited by the policy and no induced deadlock occurs.
- Subcase 2.2: If firing t at M just increases one token for an item in Constraint (7) related to some t_i, $t_i \in TS$, noticing that items in Constraint (7) related to t_i are associated with either ${}^{\bullet}t_i$ or $t_i^{\bullet}$, there are two sub-subcases:
  1) Sub-subcase 2.2.1: Firing t increases one token for $M(p_{i_1})[X_{ji}^{k_i}]$, where $\{p_{i_1}\} = {}^{\bullet}t_i$. According to Condition (I), such an event just corresponds to loading a new job. However, the argument of correctness of a policy at marking $M (\neq M_0)$ takes no consideration of loading a new job.
  2) Sub-subcase 2.2.2: Firing t at M increases one token for an item related to p_{i_2}, where $\{p_{i_2}\} = t_i^{\bullet}$, in Constraint (7). If such an event violates Constraint (7) related to t_i, then $M(p_{i_1})[X_{ji}^{k_i}] \ge 1$. Thus, t_i is enabled, which contradicts the premise of Case 2. Thus, the resulting marking after firing t_i does not violates Constraint (7) related to t_i.
- Subcase 2.3: If firing t at M decreases one token from an item in Constraint (7) related to t_i and increases one token to the item in Constraint (7) related to $t_j (i \neq j)$, it is similar to Subcase 2.2 since decreasing a token from an item of a constraint does not violate it.

The proof of Theorem 3 is completed by noticing that the above cases cover exhaustively all the possible enabled transitions of the considered ROPN. ∎

Based on the above discussion, if the TS selected in Algorithm 2 meets the two conditions in Theorem 3,

$\mathbb{MKL}(TS)$ in Algorithm 2 can be implemented via Constraints (7) related to TS. Noticed that $|TS| \leq |\overline{T}|$ ($|\overline{T}| = s$) and Constraints (7) related to TS have $|TS|$ inequities at most. When a new token enters into the system, we just reserve the spaces for it, ensuring the advancement to the second processing stage. Thus, Constraints (7) related to TS are not too conservative.

C. COMPLEXITY OF THE PROPOSED APPROACH

The proposed approach is sketched in Algorithm 2. Its computational complexity mainly depends on the loop Statements 8–28. Each loop deals with an SCS. We can partition them into four program blocks.

1) To implement Statement 15, it should scan each $t \in T(G)$ to calculate $\mathcal{P}a(G)$ and its computational complexity is $O(|T|)$. To calculate $\mathcal{N}b(G)$ via Algorithm 3, it needs to solve an ILP problem which is exponential in theory with the number of integer variables and constraints. However, the ILP problem contains only one constraint and the number of variables $|T_p|$ is finite, which can be solved in polynomial complexity [66]. On the other hand, three properties are proposed to verify $t \notin \mathcal{N}b(G)$, which is polynomial and significantly reduces the computational effort.
2) Statements 20–23 select some specific transitions, which can be completed with computational complexity $O(|T|)$.
3) Statement 25 calculates the strongly connected components in a digraph, which is linear with respect to the number of vertices in it. Thus, the computational complexity is $O(|T| + |P|)$.
4) To implement Statement 35, the two conditions in Theorem 3 should be checked and it can be completed with $O((|P| + |T|)^2)$.

Suppose that there are L SCCSs in an ROPN. Algorithm 2 can terminate with less than $L \bullet (|T| + |P|)^k$, $k \in \mathbb{N}$, times, which is polynomial to the scale of an ROPN.

V. ILLUSTRATIVE EXAMPLES

Example 1 (Continued): Apply Algorithm 2 to the subnet shown in Figure 3. We trace through the operation of the algorithm.

1) The SCCS, i.e., the sub-net in Figure 4, is obtained after execution of Statements 2–4.
2) By Statements 15–19, $\mathcal{N}b(G) = \{t_9\}$ is removed from the subnet in Figure 4. Then, Statements 25–27 are executed;
3) In the next iteration, Statement 20 is performed and we obtain $TS = \{t_8\}$ which meets the two conditions in Theorem 3.
4) Statement 35 is performed. $\mathbb{MKL}(TS)$ is developed by imposing Constraint (8) on the ROPN:

$$3M(p_{13})[X_2^1] + 3M(p_{12})[X_2^2] + 2M(p_{12})[X_1^3] \leq 5 \quad (8)$$

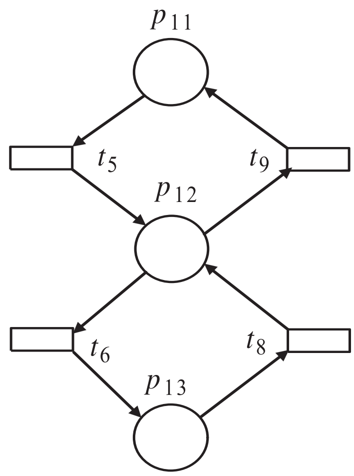


FIGURE 4. The connected sub-graph of Figure 3.

It can be verified that the ROPN in Figure 3 is prone to deadlock and the controlled ROPN is live. The system has 324 reachable states originally. After controlled by the proposed policy, 260 safe states are reachable. If the policy in [8] is applied to the system with the single control variable being minimal (set to 1) to pursuit more permissive, 228 safe states are reachable, which shows that the proposed policy is more permissive for this example.

Example 2: It is from [31] with a little modification. The WS^{3}PR model and the ROPN model of the AMS are shown in Figures 5 and 6, respectively. Two types of parts are being processed in the system. In the ROPN shown in Figure 6, for part type X_1, it has flexible processing routes. One is $t_1p_{12}t_2p_{13}t_3p_{16}t_5p_{15}t_7$ and the other is $t_1p_{12}t_2p_{13}t_{11}p_{14}t_6p_{15}t_7$. For part type X_2, it has only one processing route $t_8p_{15}t_9p_{14}t_{10}p_{13}t_{11}p_{12}t_{12}$. The other components of the ROPN is listed in the Table 2.

Let us trace through the operations as the logic presented in Algorithm 2. The ROPN can be verified to be prone to deadlock and the final controlled ROPN is live.

1) The set of transitions $\overline{T} \cup \underline{T} = \{t_1, t_8, t_7, t_{12}\}$ is removed from Figure 6 after the execution of Statements 2–4.
2) Statement 20 is performed twice in two successive "while-loop", $TS = \{t_2, t_9\}$ is calculated. Then, TS is removed. The remainder sub-net is shown in Figure 7.
3) In the next iteration, Statement 15 is executed. $\mathcal{N}b(G) = \{t_4\}$ is calculated and removed. Then, no SCCS exists.

TABLE 2. The input and output relation of the ROPN model in Figure 6.

$W(p_{12}) = X_1^1 + 2X_2^4$ $K(p_{12}) = 4$	$O(p_{12}, t_1) = X_1^1$ $I(p_{12}, t_2) = X_1^1$	$I(p_{16}, t_5) = X_1^4$ $O(p_{15}, t_5) = X_1^5$
$W(p_{15}) = X_1^5 + X_2^1$ $K(p_{15}) = 4$	$W(p_{16}) = X_1^4$ $K(p_{16}) = 1$	$I(p_{13}, t_3) = X_1^2$ $O(p_{16}, t_3) = X_1^4$
$W(p_{13}) = 2X_1^2 + X_2^3$ $K(p_{13}) = 4$	$O(p_{13}, t_2) = X_1^2$ $I(p_{13}, t_4) = X_1^2$	$O(p_{14}, t_4) = X_1^3$ $I(p_{14}, t_6) = X_1^3$
$W(p_{14}) = 2X_1^3 + 3X_2^2$ $K(p_{14}) = 5$	$O(p_{15}, t_6) = X_1^5$ $I(p_{15}, t_7) = X_1^5$	$O(p_{15}, t_8) = X_2^1$ $I(p_{15}, t_9) = X_2^1$
$O(p_{14}, t_9) = X_2^2$ $I(p_{14}, t_{10}) = X_2^2$	$O(p_{13}, t_{10}) = X_2^3$ $I(p_{13}, t_{11}) = X_2^3$	$O(p_{12}, t_{11}) = X_2^4$ $I(p_{12}, t_{12}) = X_2^4$

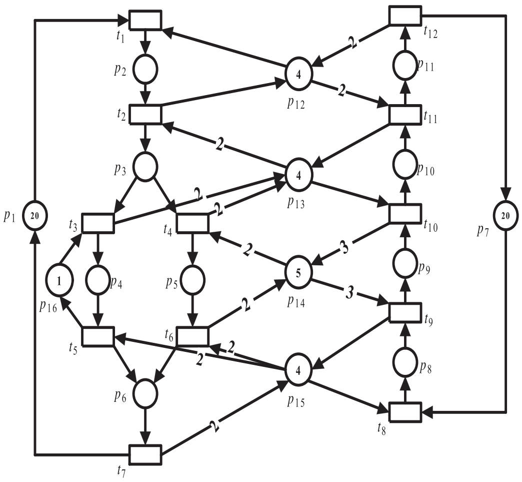


FIGURE 5. The WS^{3}PR model for an AMS of Example 2.

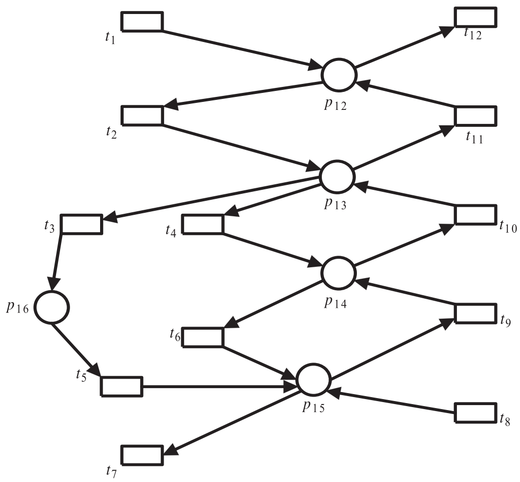


FIGURE 6. The ROPN of Example 2.

4) Statement 35 is performed and $\mathbb{MKL}(TS)$ is developed by imposing Constraint (9) on the ROPN:

$$2M(p_{12})[X_1^1]+2M(p_{13})[X_1^2]+M(p_{13})[X_2^3] \le 4$$
$$2M(p_{15})[X_2^1]+2M(p_{14})[X_2^2]+3M(p_{14})[X_1^3] \le 5 \quad (9)$$

VI. CONCLUSION AND FUTURE WORK

The taxonomy of DS-AMS is very popular in the field of intelligent manufacturing. This work deals with the issues of modeling and operations of DS-AMSs. A new modeling method for such a system by using ROPN is developed based on the prevalent WS^{3}PR model in the literature. The ROPN model is more compact and intuitive in depicting the processing routes of jobs. It is meaningful to explore more structural analysis techniques on its deadlock prevention. A key structure named SCCS in an ROPN pertinent to deadlocks is proposed, which is very useful. Based on the sufficient and necessary condition on liveness of the ROPN, we develop an approach for deadlock prevention of ROPN models. Instead of enforcing all transitions in the ROPN to be live, only a small portion of them, i.e., the transitions picked out by Function *Select*(), are controlled to make the system live. In essence, partial deadlocks in an un-controlled ROPN are avoided in the controlled system. Furthermore, the computational complexity is polynomial. In all, this strategy has many advantages compared with the existing ones.

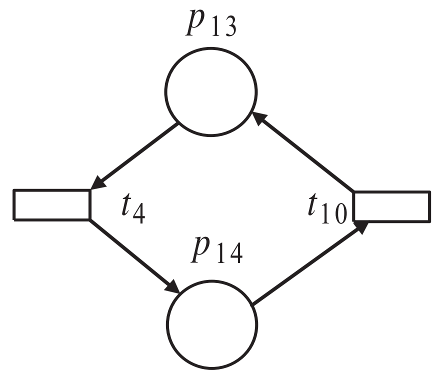


FIGURE 7. The remainder sub-graph of Example 2.

Some deficiency still exists in the proposed approach. The conditions proposed in Algorithm 3 are at some extent strict. If the selected *TS* does not meet them, deciding how to develop a more permissive policy remains an open problem and future work will focus on this. In the future work, we will use ROPN modeling paradigm to deal with the reconfigurable real-time systems [67] including wireless sensor networks [68], [69].

REFERENCES

[1] C. Gu, Z. Li, N. Wu, M. Khalgui, T. Qu, and A. Al-Ahmari, "Improved multi-step look-ahead control policies for automated manufacturing systems," *IEEE Access*, vol. 6, pp. 68824–68838, 2018, doi: 10.1109/ACCESS.2018.2872572.

[2] G. Zhu, Z. Li, N. Wu, and A. Al-Ahmari, "Fault identification of discrete event systems modeled by Petri nets with unobservable transitions," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 2, pp. 333–345, Feb. 2019.

[3] G. Zhu, Z. Li, and N. Wu, "Model-based fault identification of discrete event systems using partially observed Petri nets," *Automatica*, vol. 96, pp. 201–212, Oct. 2018.

[4] X. Cong, M. P. Fanti, A. M. Mangini, and Z. Li, "Decentralized diagnosis by Petri nets and integer linear programming," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 10, pp. 1689–1700, Aug. 2017.

[5] G. Liu, P. Li, Z. Li, and N. Wu, "Robust deadlock control for automated manufacturing systems with unreliable resources based on Petri net reachability graphs," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published. doi: 10.1109/TSMC.2018.2815618.

[6] H. Zhang, L. Feng, and Z. Li, "A learning-based synthesis approach to the supremal nonblocking supervisor of discrete-event systems," *IEEE Trans. Autom. Control*, vol. 63, no. 10, pp. 3345–3360, Oct. 2018.

[7] X. Y. Cong, M. P. Fanti, A. M. Mangini, and Z. W. Li, "On-line verification of current-state opacity by Petri nets and integer linear programming," *Automatica*, vol. 94, pp. 205–213, 2018.

[8] 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.*, to be published, doi: 10.1109/TSMC.2017.2767902.

[9] Y. Feng, K. Xing, Z. Gao, and Y. Wu, "Transition cover-based robust Petri net controllers for automated manufacturing systems with a type of unreliable resources," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 11, pp. 3019–3029, Nov. 2017.

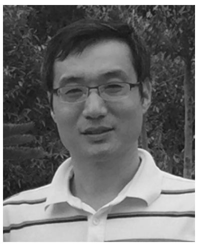
[10] C. Gu, X. Wang, and Z. Li, "Synthesis of supervisory control with partial observation on normal state-tree structures," *IEEE Trans. Autom. Sci. Eng.*, to be published, doi: 10.1109/TASE.2018.2880178.

[11] K. Xing, M. Zhou, H. Liu, and F. Tian, "Optimal Petri-net-based polynomial-complexity deadlock-avoidance policies for automated manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 39, no. 1, pp. 188–199, Jan. 2009.

[12] K. Xing, M. Zhou, F. Wang, H. Liu, and F. Tian, "Resource-transition circuits and siphons for deadlock control of automated manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 41, no. 1, pp. 74–84, Jan. 2011.

[13] Y. Tong, Z. Li, C. Seatzu, and A. Giua, "Verification of state-based opacity using Petri nets," *IEEE Trans. Autom. Control*, vol. 62, no. 6, pp. 2823–2837, Jun. 2017.
[14] Y. Tong, Z. Li, and A. Giua, "On the equivalence of observation structures for Petri net generators," *IEEE Trans. Autom. Control*, vol. 61, no. 9, pp. 2448–2462, Sep. 2016.
[15] Z. Ma, Z. Li, and A. Giua, "Characterization of admissible marking sets in Petri nets with conflicts and synchronizations," *IEEE Trans. Autom. Control*, vol. 62, no. 3, pp. 1329–1341, Mar. 2017.
[16] Z. Ma, Y. Tong, Z. Li, and A. Giua, "Basis marking representation of Petri net reachability spaces and its application to the reachability problem," *IEEE Trans. Autom. Control*, vol. 62, no. 3, pp. 1078–1093, Mar. 2017.
[17] X. Wang, I. Khemaissia, M. Khalgui, Z. Li, O. Mosbahi, and M. Zhou, "Dynamic low-power reconfiguration of real-time systems with periodic and probabilistic tasks," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 1, pp. 258–271, Jan. 2015.
[18] E. G. Coffman, M. Elphick, and A. Shoshani, "System deadlocks," *ACM Comput. Surv.*, vol. 3, no. 2, pp. 66–78, 1971.
[19] S. A. Reveliotis and P. M. Ferreira, "Deadlock avoidance policies for automated manufacturing cells," *IEEE Trans. Robot. Autom.*, vol. 12, no. 6, pp. 845–857, Dec. 1996.
[20] 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.
[21] Y. Chen, Z. Li, A. Al-Ahmari, N. Wu, and T. Qu, "Deadlock recovery for flexible manufacturing systems modeled with Petri nets," *Inf. Sci.*, vol. 381, pp. 290–303, Mar. 2017.
[22] H. Zhang, L. Feng, N. Wu, and Z. Li, "Integration of learning-based testing and supervisory control for requirements conformance of black-box reactive systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 1, pp. 2–15, Jan. 2018.
[23] Y. Chen, Z. Li, K. Barkaoui, and A. Giua, "On the enforcement of a class of nonlinear constraints on Petri nets," *Automatica*, vol. 55, pp. 116–124, May 2015.
[24] 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.
[25] M. Uzam, Z. Li, G. Gelen, and R. S. Zakariyya, "A divide-and-conquer-method for the synthesis of liveness enforcing supervisors for flexible manufacturing systems," *J. Intell. Manuf.*, vol. 27, no. 5, pp. 1111–1129, Oct. 2016.
[26] N. Ran, S. Wang, and W. Wu, "Event feedback supervision for a class of Petri nets with unobservable transitions," *IEEE Access*, vol. 6, pp. 6920–6926, 2018.
[27] S. A. Reveliotis, *Real-Time Management of Resource Allocation Systems: A Discrete Event Systems Approach*. New York, NY, USA: Springer, 2006.
[28] G. Liu and K. Barkaoui, "Necessary and sufficient liveness condition of GS^3PR Petri nets," *Int. J. Syst. Sci.*, vol. 46, no. 7, pp. 1147–1160, 2015.
[29] G. Liu, C. Jiang, M. Zhou, and A. Ohta, "The liveness of WS^3PR: Complexity and decision," *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. E96.A, no. 8, pp. 1783–1793, 2013.
[30] M. Liu *et al.*, "Deadlock and liveness characterization for a class of generalized Petri nets," *Inf. Sci.*, vol. 420, pp. 403–416, Dec. 2017.
[31] D. Liu, Z. Li, and M. Zhou, "Liveness of an extended S^3PR," *Automatica*, vol. 46, no. 6, pp. 1008–1018, 2010.
[32] F. Tricas, F. Garcia-Valles, J. M. Colom, and J. Ezpeleta, "A Petri net structure-based deadlock prevention solution for sequential resource allocation systems," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2005, pp. 271–277.
[33] S. Wang, D. You, and M. Zhou, "A necessary and sufficient condition for a resource subset to generate a strict minimal siphon in S^4PR," *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 4173–4179, Aug. 2017.
[34] J. Park and S. A. Reveliotis, "Deadlock avoidance in sequential resource allocation systems with multiple resource acquisitions and flexible routings," *IEEE Trans. Autom. Control*, vol. 46, no. 10, pp. 1572–1583, Oct. 2001.
[35] D. You, S. Wang, W. Dai, W. Wu, and Y. Jia, "An approach for enumerating minimal siphons in a subclass of Petri nets," *IEEE Access*, vol. 6, pp. 4255–4265, 2018.
[36] N. Wu, "Necessary and sufficient conditions for deadlock-free operation in flexible manufacturing systems using a colored Petri net model," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 29, no. 2, pp. 192–204, May 1999.
[37] N. Wu, *System Modeling and Control With Resource-Oriented Petri Nets*. Boca Raton, FL, USA: CRC Press, 2009.
[38] N. Wu and M. Zhou, "Avoiding deadlock and reducing starvation and blocking in automated manufacturing systems," *IEEE Trans. Robot. Autom.*, vol. 17, no. 5, pp. 658–669, Oct. 2001.
[39] N. Wu and M. Zhou, "Modeling and deadlock avoidance of automated manufacturing systems with multiple automated guided vehicles," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 35, no. 6, pp. 1193–1202, Dec. 2005.
[40] N. Wu and M. Zhou, "Shortest routing of bidirectional automated guided vehicles avoiding deadlock and blocking," *IEEE/ASME Trans. Mechatronics*, vol. 12, no. 1, pp. 63–72, Feb. 2007.
[41] N. Q. Wu and M. Zhou, "Modeling, analysis and control of dual-arm cluster tools with residency time constraint and activity time variation based on Petri nets," *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 2, pp. 446–454, Apr. 2012.
[42] N. Wu, F. Chu, C. Chu, and M. Zhou, "Petri net modeling and cycle-time analysis of dual-arm cluster tools with wafer revisiting," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 43, no. 1, pp. 196–207, Jan. 2013.
[43] N. Wu, M. Zhou, and Z. Li, "Short-term scheduling of crude-oil operations: Enhancement of crude-oil operations scheduling using a Petri net-based control-theoretic approach," *IEEE Robot. Autom. Mag.*, vol. 22, no. 2, pp. 64–76, Jun. 2015.
[44] L. Bai, N. Wu, Z. Li, and M. Zhou, "Optimal one-wafer cyclic scheduling and buffer space configuration for single-arm multicluster tools with linear topology," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 10, pp. 1456–1467, Oct. 2016.
[45] N. Wu, M. Zhou, and Z. Li, "Resource-oriented Petri net for deadlock avoidance in flexible assembly systems," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 1, pp. 56–69, Jan. 2008.
[46] H. Chen, N. Wu, and M. Zhou, "A novel method for deadlock prevention of AMS by using resource-oriented Petri nets," *Inf. Sci.*, vol. 363, pp. 178–189, Oct. 2016.
[47] Y. Hou, N. Wu, M. Zhou, and Z. Li, "Pareto-optimization for scheduling of crude oil operations in refinery via genetic algorithm," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 3, pp. 517–530, Mar. 2017.
[48] H. Chen, N. Wu, Z. Li, and T. Qu, "On a maximally permissive deadlock prevention policy for automated manufacturing systems by using resource-oriented Petri nets," *ISA Trans.*, to be published. doi: 10.1016/j.isatra.2018.11.025.
[49] N. Wu, M. Zhu, L. Bai, and Z. Li, "Short-term scheduling of crude oil operations in refinery with high-fusion-point oil and two transportation pipelines," *Enterprise Inf. Syst.*, vol. 10, no. 6, pp. 581–610, 2016.
[50] 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.
[51] N. Wu, L. Bai, and M. Zhou, "An efficient scheduling method for crude oil operations in refinery with crude oil type mixing requirements," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 3, pp. 413–426, Mar. 2016.
[52] Y. Qiao, N. Wu, F. Yang, M. Zhou, and Q. Zhu, "Wafer sojourn time fluctuation analysis of time-constrained dual-arm cluster tools with wafer revisiting and activity time variation," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 4, pp. 622–636, Apr. 2018.
[53] S. Zhang, N. Wu, Z. Li, T. Qu, and C. Li, "Petri net-based approach to short-term scheduling of crude oil operations with less tank requirement," *Inf. Sci.*, vol. 417, pp. 247–261, Nov. 2017.
[54] F. Yang, N. Wu, Y. Qiao, M. Zhou, and Z. Li, "Scheduling of single-arm cluster tools for an atomic layer deposition process with residency time constraints," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 3, pp. 502–516, Mar. 2017.
[55] Q. Zhu, M. Zhou, Y. Qiao, and N. Wu, "Petri net modeling and scheduling of a close-down process for time-constrained single-arm cluster tools," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 48, no. 3, pp. 389–400, Mar. 2018.
[56] N. Wu, Z. Li, and T. Qu, "Energy efficiency optimization in scheduling crude oil operations of refinery based on linear programming," *J. Cleaner Prod.*, vol. 166, pp. 49–57, Nov. 2017.
[57] N. Wu, M. Zhou, and G. Hu, "One-step look-ahead maximally permissive deadlock control of AMS by using Petri nets," *ACM Trans. Embedded Comput. Syst.*, vol. 12, no. 1, 2013, Art. no. 10.
[58] T. Murata, "Petri nets: Properties, analysis and applications," *Proc. IEEE*, vol. 77, no. 4, pp. 541–580, Apr. 1989.
[59] D. Liu, J. Wang, S. C. F. Chan, J. Sun, and L. Zhang, "Modeling workflow processes with colored Petri nets," *Comput. Ind.*, vol. 49, no. 3, pp. 267–281, 2002.

[60] B. Târnaucă, D. Puiu, V. Comnac, and C. Suciu, "Modelling a flexible manufacturing system using reconfigurable finite capacity Petri nets," in *Proc. Int. Conf. Optim. Elect. Electron. Equip.*, 2012, pp. 1079–1084.
[61] M. A. Marsan, A. Bobbio, and S. Donatelli, "Petri nets in performance analysis: An introduction," in *Lectures on Petri Nets I: Basic Models*. Berlin, Germany: Springer-Verlag, 1996.
[62] [Online]. Available: https://en.wikipedia.org/wiki/Multiset
[63] J. Ezpeleta, J. M. Colom, and J. Martínez, "A Petri net based deadlock prevention policy for flexible manufacturing systems," *IEEE Trans. Robot. Autom.*, vol. 11, no. 2, pp. 173–184, Apr. 1995.
[64] S. Reveliotis and A. Nazeem, "Deadlock avoidance policies for automated manufacturing systems using finite state automata," in *Formal Methods in Manufacturing*, J. Campos, C. Seatzu, and X. Xie, Eds. Boca Raton, FL, USA: CRC Press, 2014, pp. 169–195.
[65] [Online]. Available: https://en.wikipedia.org/wiki/Strongly_connected_component
[66] K. Li and Q. Li, "Polynomial algorithm for integer programming with a fixed number of variables," *Mini-Micro Syst.*, vol. 25, no. 7, pp. 1298–1302, 2004.
[67] W. Lakhdhar, R. Mzid, M. Khalgui, Z. W. Li, G. Frey, and A. Al-Ahmari, "Multiobjective optimization approach for a portable development of reconfigurable real-time systems: From specification to implementation," *IEEE Trans. Syst., Man, Cybern. Syst.*, to be published. doi: 10.1109/TSMC.2017.2781460.
[68] H. Grichi, O. Mosbahi, M. Khalgui, and Z. W. Li, "New power-oriented methodology for dynamic resizing and mobility of reconfigurable wireless sensor networks," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 48, no. 7, pp. 1120–1130, 2018.
[69] M. Gasmi, O. Mosbahi, M. Khalgui, L. Gomes, and Z. W. Li, "R-Node: New pipelined approach for an effective reconfigurable wireless sensor node," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 48, no. 6, pp. 892–905, Jun. 2018.



HEFENG CHEN received the B.S. degree in fundamental mathematics from the Hubei University of Nationalities, Ehshi, Hubei, China, in 1997, and the M.S. degree in application of computer science from the Guangdong University of Technology, Guangzhou, China, in 2003, where he is currently pursuing the Ph.D. degree in systems engineering. He joined the Guangdong University of Technology, in 2003. His current research interests include Petri nets, discrete event systems, and manufacturing system modeling and control.

NAIQI WU (M'04–SM'05–F'19) received the B.S. degree in electrical engineering from the Anhui University of Technology, Huainan, China, in 1982, and the M.S. and Ph.D. degrees in systems engineering from Xi'an Jiaotong University, Xi'an, China, in 1985 and 1988, respectively. From 1988 to 1995, he was with the Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China, and from 1995 to 1998, he was with Shantou University, Shantou, China. He moved to the Guangdong University of Technology, Guangzhou, China, in 1998. He joined the Macau University of Science and Technology, Macau, China, in 2013. He was a Visiting Professor with The Arizona State University, Tempe, AZ, USA, in 1999; the New Jersey Institute of Technology, Newark, NJ, USA, in 2004; the University of Technology of Troyes, Troyes, France, from 2007 to 2009; and Evry University, Evry, France, from 2010 to 2011. He is currently a Professor with the Institute of Systems Engineering, Macau University of Science and Technology. He has authored or co-authored one book, five book chapters, and over 150 peer-reviewed journal papers. His research interests include production planning and scheduling, manufacturing system modeling and control, discrete event systems, Petri net theory and applications, intelligent transportation systems, and energy systems. He was an Associate Editor of the IEEE Transactions on Systems, Man, and Cybernetics—Part C, the IEEE Transactions on Automation Science and Engineering, and the IEEE Transactions on Systems, Man, and Cybernetics: Systems, and the Editor-in-Chief of the *Industrial Engineering Journal*. He is an Associate Editor of *Information Sciences* and the IEEE/CAA Journal of Automatica Sinica.

ZHIWU LI (M'06–SM'07–F'16) received the B.S. degree in mechanical engineering, the M.S. degree in automatic control, and the Ph.D. degree in manufacturing engineering from Xidian University, Xi'an, China, in 1989, 1992, and 1995, respectively.

He joined Xidian University, in 1992. He has published two monographs in Springer, in 2009, and CRC Press, in 2013, with M. Zhou and Y. Chen, respectively. He is also currently with the Institute of Systems Engineering, Macau University of Science and Technology, Macau, China. His current research interests include Petri net theory and application, supervisory control of discrete event systems, workflow modeling and analysis, system reconfiguration, game theory, and data and process mining.

Dr. Li is listed in Marquis Who's Who in the World, 27th Edition, in 2010. He is the Founding Chair of the Xi'an Chapter of the IEEE Systems, Man, and Cybernetics Society.

TING QU received the B.S. and M.S. degrees in mechanical engineering from Xi'an Jiaotong University, Xi'an, China, in 2001 and 2004, respectively, and the Ph.D. degree in industrial and manufacturing systems engineering from The University of Hong Kong, Hong Kong, in 2008, where he was a Postdoctoral Research Fellow and then a Research Assistant Professor with the Department of Industrial and Manufacturing Systems Engineering, from 2008 to 2010. He joined the Guangdong University of Technology, Guangzhou, China, in 2010. He moved to Jinan University at Zhuhai, Zhuhai, China, in 2016, where he is currently a Professor with the School of Electrical and Information Engineering. He has published over 100 technical papers with more than half of them in reputable journals. His research interests include the IoT-based smart manufacturing systems, logistics and supply chain management, and industrial product/production service systems. He serves as an Associate Editor (Asia) of the *International Journal of Computer Integrated Manufacturing* and an Editorial Board Member of the *Industrial Engineering Journal*.

HAILIAN XIAO received the B.S. degree in physics and the M.S. degree in western economics from South China Normal University, Guangzhou, China, in 1999 and 2006, respectively, and the Ph.D. degree in accounting from Sun Yat-sen University, Guangzhou, in 2012. He joined South China Normal University, in 2013. His current research interests include sci-tech finance, internal controls, and the application of nonlinear time-varying factor models.

• • •