

Research Article

Mixed Integer Programming-Based Liveness Test for FMS with Full Routing Flexibility

Lida Dong,^{1,2} Tianyang Chi,¹ Chengcheng Zhu,² and Jun Yin²

¹ Hangzhou Institution of Service Engineering, Hangzhou Normal University, Hangzhou 310027, China

² Institution of Electronic Circuit & Information Systems, Zhejiang University, Hangzhou 310027, China

Correspondence should be addressed to Tianyang Chi; tianyang.chi@hotmail.com

Received 31 December 2013; Revised 20 June 2014; Accepted 2 July 2014; Published 7 August 2014

Academic Editor: Javier Oliver

Copyright © 2014 Lida Dong et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Mixed integer programming (MIP) is an important technique to verify the liveness property of sequential flexible manufacturing systems (FMS) modeled by Petri nets. When there are some fully flexible routings in FMS, the existing MIP-based methods are not suitable for testing their liveness. This paper defines a subclass of S^*PR nets firstly, namely, OSC- S^*PR nets, and concludes that an OSC- S^*PR net is live if there exist no non-max'-controlled siphons. Accordingly, determining whether or not an OSC- S^*PR net is live can also be realized by using standardized mixed integer programming (MIP) tools. Furthermore, the liveness property of S^*PR nets can be tested in two steps: first, for a given S^*PR net, constructing an OSC- S^*PR net to ensure that if the latter is live then the former must be live; second, testing liveness of the constructed OSC- S^*PR net by the aforementioned MIP-based algorithm. In the end, the performance of the method is demonstrated by an application of FMS.

1. Introduction

Flexible manufacturing systems (FMS) utilize computers and shared resources (such as robots, machines, and automated guided vehicles) to automatically produce products [1]. Owing to the competition of shared resources, there may be some processes in FMS, which once started cannot be finished, that is, deadlocks [2]. Hence, it should be ensured that no deadlocks will occur in FMS; in other words, all working processes should be live. Usually, liveness property of FMS might be tested by analyzing their models (such as Petri nets) before using. Since the model complexity will increase with job shop flexibility (alternative machines, alternative operation sequences, and full routing flexibility) [3], the work for detecting liveness of FMS will also increase correspondingly.

At present, there are two categories of techniques for liveness test: one category is relied on the classic state enumeration analysis of Petri nets, which grows exponentially with the size of the net [4]; the other one is based on standardized tools of mixed integer programming (MIP), which is originally proposed by Chu and Xie [5] and is more efficient than the former. The basic idea of the latter is that

if there does not exist any siphon becoming empty in a Petri net, then the net is deadlock free. It has been verified that for some subclasses of ordinary Petri nets such as ES^2PR nets [6], S^3PR nets [7], LS^3PR [8] nets, and ES^3PR [9] nets the net is live if and only if it is deadlock free. So, the liveness of these subclasses of ordinary Petri nets can be tested by MIP-based algorithms [10, 11]. The natural thing is to extend MIP-based method of liveness test for general Petri nets. Park and Reveliotis [12] introduced an MIP-based algorithm for testing the weakly liveness of FMS with alternative machines flexibility modeled by S^3PGR^2 nets, which is in view of the conclusion that a net is weakly live if there exist no dead marked siphons (DMS). Shih et al. [13] has obtained some similar results for S^3PGR^2 nets, in which DMS is found on the grounds of the conception of max'-controlled siphons. Moreover, Liu and Li [14] provided a general MIP algorithm of liveness test for S^3PGR^2 nets, which is based on the result that a net is live if there is no extended deadly marked siphons (EDMS). Owing to the shortage of liveness conditions of S^*PR nets, which can model FMS with full routing flexibility [15, 16], there is no MIP-based method of liveness test for S^*PR net. The liveness of Gadara nets [17] (a subclass of

S*PR nets) may be detected by the original MIP approach [5], because of the conclusion that a Gadara net is live if and only if there does not exist any empty siphons [17, 18].

This paper defined a subclass of S*PR nets firstly, namely, OSC-S*PR nets, in which all output transitions of each choice working place are restricted to use the same type and quantity of resources. The main structure characteristic of OSC-S*PR nets is that an OSC-S*PR net is live if there exists no non-max'-controlled siphons. Accordingly, an MIP-based algorithm is proposed to determine whether or not an OSC-S*PR net is live. It is worth to say that OSC-S*PR nets are an extension of Gadara nets, which are usually used to model multithreaded programs [17]. Considering the structure complexity of S*PR nets, it is difficult to develop an MIP-based algorithm for testing their liveness directly. So, an indirect method is given here, which comprises two steps as follows: first, for a given S*PR net, constructing an OSC-S*PR net to ensure that if the latter is live then the former must be live; second, testing liveness of the constructed OSC-S*PR net by the aforementioned MIP-based algorithm. In order to verify the performance of our method, a typical case of FMS is taken as example.

This paper is organized as follows. Section 2 introduces the basic knowledge about Petri nets. Section 3 presents OSC-S*PR nets and their structure characteristics. Section 4 proposes an MIP-based algorithm for testing liveness of OSC-S*PR nets. Section 5 discusses an indirect method for testing liveness of S*PR nets. A typical example is presented in Section 6.

2. Preliminaries

This section, we briefly introduce the basic definitions and notations of a Petri net, which will be discussed in the rest of this paper. More details can be found in [4, 19].

A generalized Petri net is a four-tuple $N = (P, T, F, W)$, where P and T are finite, nonempty and disjoint sets. P denotes the set of places and T denotes the set of transitions with $P \cup T \neq \emptyset$ and $P \cap T = \emptyset$. $F \subseteq (P \times T) \cup (T \times P)$ is called a flow relation of the net, represented by arcs with arrows from places to transitions or from transitions to places. $W : (P \times T) \cup (T \times P) \rightarrow \mathbf{N}$ is a mapping that assigns a weight to an arc: $W(x, y) > 0$ if and only if $(x, y) \in F$, and $W(x, y) = 0$ otherwise, where $x, y \in P \cup T$ and \mathbf{N} denotes the set of natural numbers.

A marking M of a Petri net N is a mapping from P to \mathbf{N} . $M(p)$ denotes the number of tokens in the place p . We usually describe markings and vectors using a multiset (bag) or formal sum notation for economy of space. As a result, $\sum_{p \in P} M(p)p$ is used to denote vector M . For instance, a marking that puts four tokens in place p_2 and two tokens in place p_4 only in a net with $P = \{p_1 - p_6\}$ is denoted by $4p_2 + 2p_4$ instead of $(0, 4, 0, 2, 0, 0)$. A place p is marked by marking M if and only if $M(p) > 0$. A subset $S \subseteq P$ is marked by M if and only if at least one place in S is marked by M . The sum of tokens of all places in S is denoted by $M(S)$; that is, $M(S) = \sum_{p \in S} M(p)$. S is said to be empty at M if and only if $M(S) = 0$. (N, M_0) is called a net system or marked net and

M_0 is called an initial marking of N . In general, (N, M_0) is directly called a net where there is no confusion.

Let $x \in P \cup T$ be a node of net $N = (P, T, F, W)$. $\cdot x = \{y \in P \cup T \mid (y, x) \in F\}$ denotes the preset of x , while $x \cdot = \{y \in P \cup T \mid (x, y) \in F\}$ denotes its postset of x . Furthermore, this notation can be extended to a set of nodes. For example, given $X \subseteq P \cup T$, $\cdot X = \bigcup_{x \in X} \cdot x$ and $X \cdot = \bigcup_{x \in X} x \cdot$. Given a place p , we denote $\max\{W(p, t) \mid t \in p \cdot\}$ by \max_p and $\min\{W(t, p) \mid t \in \cdot p\}$ by \min_p .

A transition $t \in T$ is enabled at marking M if and only if for all $p \in \cdot t$, $M(p) \geq W(p, t)$. This fact is denoted by $M[t]$. Firing it yields a new marking M' such that for all $p \in P$, $M'(p) = M(p) - W(p, t) + W(t, p)$, as denoted by $M[t]M'$. M' is called a immediately reachable marking from M . Marking M' is said to be reachable from M if there exists a sequence of transitions $\sigma = t_0 t_1 \dots t_n$ and markings M_1, M_2, \dots , and M_n such that $M[t_0]M_1[t_1]M_2 \dots M_n[t_n]M'$ holds. The set of markings reachable from M in N is called the reachability set of Petri net (N, M) and denoted by $R(N, M)$.

Let $N = (P, T, F, W)$ be a Petri net. N is pure (self-loop free) if and only if for all $x \in P \cup T$, $\cdot x \cap x \cdot = \emptyset$. N is called an ordinary net, denoted by $N = (P, T, F)$ if for all $f \in F$, $W(f) = 1$. $N = (P, T, F)$ is called a state machine if for all $t \in T$, $|t| = |t'| = 1$. It is strongly connected if for all $x, y \in P \cup T$, there is a sequence of nodes x, a, b, \dots, c, y such that $(x, a), (a, b), \dots, (c, y) \in F$, where $\{a, b, \dots, c\} \subseteq P \cup T$. An extended free-choice net is an ordinary Petri net such that for all $t_1, t_2 \in T$, $\cdot t_1 \cap \cdot t_2 \neq \emptyset \Rightarrow \cdot t_1 = \cdot t_2$. A pure net $N = (P, T, F, W)$ can be represented by its incidence matrix $[N]$, where $[N]$ is a $|P| \times |T|$ integer matrix with $[N](p, t) = W(t, p) - W(p, t)$. For a place p (transition t), its incidence vector, a row (column) in $[N]$, is denoted by $[N](p, \bullet)$, $([N](\bullet, t))$. The structural bound of a place p is defined as $B(p) = \max\{M(p) \mid M = M_0 + [N] \cdot Y, M \geq 0, Y \geq 0\}$.

A P -invariant Y is a P -dimensional vector which satisfies $Y \neq 0$, $Y^T \cdot [N] = 0^T$. For any reachable marking $M' \in R(N, M)$ and a P -invariant Y of N , $M^T \cdot Y = M'^T \cdot Y$. $\|Y\| = \{p \mid Y(p) \neq 0\}$ is called the support of Y . $\|Y\|^+ = \{p \mid Y(p) > 0\}$ denotes the positive support of P -invariant Y and $\|Y\|^- = \{p \mid Y(p) < 0\}$ denotes the negative support of P -invariant Y . Y is said to be minimal if $\|Y\|$ is not a superset of the support of any other one and its components are mutually prime. If for all $p \in \|Y\|$, $Y(p) > 0$, then Y is a P -semiflow.

Let S be a nonempty subset of places in N . S is a siphon (trap) if and only if $\cdot S \subseteq S \cdot$ ($S \subseteq \cdot S$). A siphon S is said to be minimal if and only if it contains no other siphons as its proper subset. A minimal siphon is strict if it does not contain a trap (SMS for short).

Given a net (N, M_0) , a transition $t \in T$ is live under M_0 if and only if for all $M \in R(N, M_0)$, $\exists M' \in R(N, M)$ such that $M'[t]$. A transition $t \in T$ is said to be dead under marking $M \in R(N, M_0)$ if and only if $\nexists M^* \in R(N, M)$ such that $M^*[t]$. (N, M_0) is quasi-live under M_0 if and only if for all $t \in T$, $\exists M \in R(N, M_0)$ such that $M[t]$. (N, M_0) is live if and only if for all $t \in T$, t is live. In this paper, we will denote the set of live transition under M by T_L and the set of dead transition for marking M by T_D . From the definition above,

it can be known that a live transition can be fired infinitely in $R(N, M_0)$. A dead transition for marking M can never be fired in $R(N, M)$.

A number of nodes n_1, n_2, \dots, n_k is called a path of N if for all $i \in \{1, 2, \dots, k-1\}$, $n_{i+1} \in n_i^*$. A simple path (not a circuit) from n_1 to n_k is a path whose nodes are all different, which can be denoted as $SP(n_1, n_k)$. A circuit is a simple path with $n_1 = n_k$.

3. Definitions and Properties of OSC-S*PR Nets

In order to model FMS with full routing flexibility and multiple copies of different resources, S*PR nets are proposed by Ezpeleta et al. [15], which are defined as follows.

Definition 1 (see [15]). Let $I_N = \{1, 2, \dots, m\}$ be a finite set of indexes. An S*PR net is a connected generalized self-loop-free Petri net $N = (P, T, F, W)$ where

- (i) $P = P_A \cup P_0 \cup P_R$ is a partition such that (a) $P_A = \bigcup_{i \in I_N} P_{A,i}$ is called the set of operation places, where $P_{A,i} \neq \emptyset$ and $P_{A,i} \cap P_{A,j} = \emptyset$ for all $i \neq j$; (b) $P_0 = \bigcup_{i \in I_N} \{p_{0,i}\}$ is called the set of idle places; (c) $P_R = \bigcup_{i \in I_N} P_{R,i} = \{r_1, r_2, \dots, r_m\}$ is called the set of resource places;
- (ii) $T = \bigcup_{i \in I_N} T_i$, $T_i \neq \emptyset$; for all $j \neq i$, $T_i \cap T_j = \emptyset$;
- (iii) $W = W_A \cup W_R$, where

$$W_A : ((P_A \cup P_0) \times T) \cup (T \times (P_A \cup P_0)) \rightarrow \{0, 1\} \text{ and}$$

$$\text{for all } j \neq i, W_A : ((P_{A,i} \cup \{p_{0,i}\}) \times T_j) \cup (T_j \times (P_{A,i} \cup \{p_{0,i}\})) \rightarrow \{0\};$$

$$W_R : (P_R \times T) \cup (T \times P_R) \rightarrow \mathbf{N}, \text{ where } \mathbf{N} \text{ is the set of natural number;}$$

- (iv) for all $i \in I_N$, the subnet N_i generated by $P_{A,i} \cup \{p_{0,i}\} \cup T_i$ is a strongly connected state machine;
- (v) for all $r \in P_R$, there exists a unique minimal P -semiflow I_r , such that $\|I_r\| \cap P_R = \{r\}$, $\|I_r\| \cap P_0 = \emptyset$, $\|I_r\| \cap P_A \neq \emptyset$ and $I_r(r) = 1$;
- (vi) $P_A = \bigcup_{r \in P_R} (\|I_r\| \setminus P_R)$;
- (vii) N is pure and strongly connected.

Definition 2 (see [15]). A well-marked S*PR net (N, M_0) is a marked Petri net $N = (P, T, F, W)$ with initial marking M_0 such that for all $p \in P_A$, $M_0(p) = 0$; for all $r \in P_R$, $M_0(r) \geq \max_{p \in \|I_r\|} I_r(p)$; and $M_0(r) \geq \max_{p \in \|I_r\|} I_r(p)$, $M_0(p_{0,i}) \geq 1$.

Due to the structure complexity of S*PR nets, the liveness problem usually relies on the classic state enumeration analysis. Obviously, it is time-consuming due to the state explosion problem. Fortunately, Gadara nets, a subclass of S*PR nets, may utilize some efficient methods such as MIP-based approach for testing their liveness. In fact, Gadara nets are a simplified subclass of S*PR nets with the restriction that no resources are required in all branches and quantity of the

same type of resources equals one. The definition is shown as follows.

Definition 3 (see [10]). Let $I_N = \{1, 2, \dots, m\}$ be a finite set of indexes. A Gadara net is an ordinary, self-loop-free Petri net $N = (P, T, F, M_0)$ where

- (i) $P = P_A \cup P_0 \cup P_R$ is a partition such that (a) $P_A = \bigcup_{i \in I_N} P_{A,i}$ is called the set of operation places, where $P_{A,i} \neq \emptyset$ and $P_{A,i} \cap P_{A,j} = \emptyset$ for all $i \neq j$; (b) $P_0 = \bigcup_{i \in I_N} \{p_{0,i}\}$ is called the set of idle places; (c) $P_R = \bigcup_{i \in I_N} P_{R,i} = \{r_1, r_2, \dots, r_m\}$ is called the set of resource places;
- (ii) $T = \bigcup_{i \in I_N} T_i$, $T_i \neq \emptyset$; for all $j \neq i$, $T_i \cap T_j = \emptyset$;
- (iii) for all $i \in I_N$, the subnet N_i generated by $P_{A,i} \cup \{p_{0,i}\} \cup T_i$ is a strongly connected state machine;
- (iv) for all $p \in P_A$, if $|p^*| > 1$, then for all $t \in p^*$, $t \cap P_R = \emptyset$;
- (v) for all $r \in P_R$, there exists a unique minimal P -semiflow I_r , such that $\|I_r\| \cap P_R = \{r\}$ (for all $p \in \|I_r\|$, $I_r(p) = 1$), $\|I_r\| \cap P_0 = \emptyset$, $\|I_r\| \cap P_A \neq \emptyset$, and $I_r(r) = 1$;
- (vi) $P_A = \bigcup_{r \in P_R} (\|I_r\| \setminus P_R)$;
- (vii) for all $r \in P_R$, $M_0(r) = 1$. For all $p \in P_A$, $M_0(p) = 0$. For all $p_0 \in P_0$, $M_0(p_0) \geq 1$.

In most cases, the modeling ability of Gadara nets is not enough. Here, a new subclass of S*PR nets, namely, OSC-S*PR nets, is introduced as follows.

Definition 4. Let (N, M_0) be a well-marked S*PR net. A place p is said to be a choice operation place if and only if $p \in P_A$ and $|p^*| \geq 2$. The set of choice operation places in (N, M_0) is denoted by P_C ; that is, $P_C = \{p \in P_A \mid |p^*| \geq 2\}$.

Definition 5. Let (N, M_0) be a well-marked S*PR net. A choice operation place $p \in P_C$ is said to be satisfied operation symmetry choice condition (OSC-Condition) if and only if

$$\begin{aligned} \forall t_1, t_2 \in p^*, \quad t_1 = t_2, \\ \forall p' \in t_1^* = t_2^*, \quad w(p', t_1) = w(p', t_2). \end{aligned} \quad (1)$$

(N, M_0) is said to be a well-marked OSC-S*PR net if and only if for all $p \in P_C$, p satisfies OSC-Condition.

The OSC-Condition confirms that each output transition of each choice operation place must use the same kind and quantity of resources. From Definition 5, it is apparent that OSC-S*PR nets are a subclass of S*PR nets. Since any branch in a Gadara net cannot be associated with any resource [17], it satisfies OSC-Condition as well; that is, Gadara nets are a special case of OSC-S*PR nets.

Example 6. Consider the net shown in Figure 1. For the choice operation places p_3 and p_7 , all output transitions t_{v1} , t_{v2} , t_{v3} , and t_{v4} use none of resources. So it is an OSC-S*PR net but is not a Gadara net because initial tokens in r_1 , r_2 , r_3 , r_A , and r_B are greater than one.

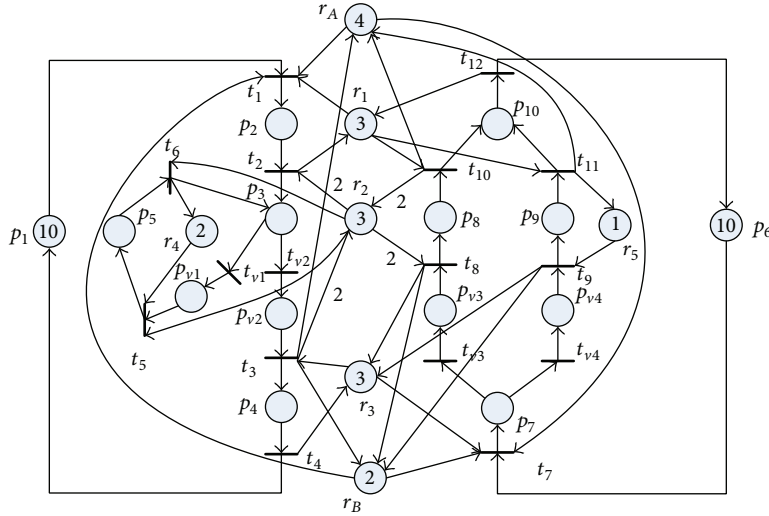


FIGURE 1: An OSC-S*PR net.

Similar to the technique in [20, 21], the definition of \max' -controlled siphon of S*PR nets is introduced as follows.

Definition 7. Let S be a siphon of a S*PR net (N, M_0) , where $S = S_R \cup S_A$, $S_R = S \cap P_R \neq \emptyset$, and $S_A = S \cap P_A \neq \emptyset$. The holder of resource r is defined as the difference of two sets $\|I_r\|$ and $\{r\}$: $H(r) = \|I_r\| - \{r\}$, and $[S] = \|\bigcup_{r \in S_R} H_r\| \setminus S$ is the complementary set of siphon S .

Definition 8. Let S be a siphon of a well-marked S*PR net (N, M_0) . S is said to be \max' -marked at marking $M \in R(N, M_0)$ if and only if $\exists p \in S_A$ such that $M(p) \geq 1$ or $\exists r \in S_R$ such that $M(r) \geq \max_{t \in r \cdot \cap [S]} W(r, t)$. S is said to be \max' -controlled if and only if it is \max' -marked under any reachable marking. (N, M_0) is said to be \max' -controlled if and only if each siphon of it is \max' -controlled.

Before proving the liveness condition of OSC-S*PR nets, some lemmas are established first.

Lemma 9. Let p be a choice operation place in a well-marked OSC-S*PR net (N, M_0) . For all $M \in R(N, M_0)$, the following conclusions hold.

- (i) If $\exists t_1 \in p^\bullet$ such that $M[t_1]$, then for all $t_2 \in p^\bullet$ we have $M[t_2]$.
- (ii) If $\exists t_1 \in p^\bullet$ such that t_1 is disabled under M , then for all $t_2 \in p^\bullet$ we have that t_2 is disabled under M .

Proof. This is trivial. Because from Definition 5, for all $t_1, t_2 \in p^\bullet$, $t_1 = t_2$ and for all $p \in t_1$, $w(p, t_1) = w(p, t_2)$. Therefore, all of them are enabled for making M if one of them is enabled and all of them are disabled for making M if one of them is disabled. \square

Lemma 10. Let (N, M_0) be a well-marked S*PR net. For all $p_i \in P_A$, there must exist a simple path $SP(p_i, p_i^0)$ from p_i to p_i^0 .

Proof. This is trivial, since by condition (iv) in Definition 1, N_i generated by $P_{A_i} \cup \{p_i^0\} \cup T_i$ is strongly connected. \square

Lemma 11. Let (N, M_0) be a well-marked S*PR net. Then, N is quasi-live under M_0 .

Proof. From M_0 , by Definition 2, for all $t_0 \in P^{0*}$, t_0 is enabled. For the transition $t_{j,i} \in T_i \setminus P^{0*}$, since (N, M_0) is well-marked, we can construct a trivial firing sequence $s = t_{0,i}, t_{1,i}, \dots, t_{(j-1),i}$ from M_0 by only allowing the execution of only one sequential process i and only one token flowing into process i , reaching a marking M such that $M[t_{j,i}]$. This is valid for each $t \in T$. \square

Lemma 12. Let (N, M_0) be a well-marked S*PR net, and let $M \in R(N, M_0)$ be a reachable marking so that $t \in T$ is a dead transition under M . Then, $M_0 \notin R(N, M)$.

Proof. Assume that $M_0 \in R(N, M)$. By Lemma 11, $\exists M' \in R(N, M)$ such that $M'[t]$. This contradicts the hypothesis that $t \in T$ is a dead transition under M . Thus, Lemma 12 holds. \square

Similar to the method in [7], Lemma 13 is proposed which means that if there is a reachable marking such that a transition is dead, then there exists a set of processes where some tokens cannot evolve any more.

Lemma 13. Let (N, M_0) be a well-marked OSC-S*PR net. $M \in R(N, M_0)$ and $t^* \in T$ is a dead transition under M ; then there exist $M' \in R(N, M)$ and two subsets $J, H \in I_N$ such that $I_N = J \cup H$, $I_N = \{1, 2, \dots, n\}$, $J \cap H = \emptyset$, $J \neq \emptyset$, and (i) for all $h \in H$, $M'(p_h^0) = M_0(p_h^0)$; (ii) for all $j \in J$, $M'(p_j^0) < M_0(p_j^0)$, and $\Omega = \{p^\bullet \mid p \in P_A, M'(p) > 0\}$ is a set of dead transitions.

Proof. By condition (ii) of Definition 1, for all $i \in I_N$ and for all $M' \in R(N, M')$, $M'(p_i^0) \leq M_0(p_i^0)$. Therefore, there must

exist $M' \in R(N, M)$ and two subsets $J, H \in I_N$ such that $I_N = J \cup H$, $I_N = \{1, 2, \dots, n\}$, $J \cap H = \emptyset$ and (i) for all $h \in H$, $M'(p_h^0) = M_0(p_h^0)$; (ii) for all $j \in J$, $M'(p_j^0) < M_0(p_j^0)$.

First, prove $J \neq \emptyset$. Assume that $J = \emptyset$. Then, it can be concluded that $H = I_N$. Therefore, for all $h \in I_N$, we have $M'(p_h^0) = M_0(p_h^0)$, $M' = M_0$. Since (N, M_0) is a well-marked OSC-S*PR net, by Lemma 11, it is quasi-live. Therefore, for all $t'' \in T$, $\exists M'' \in (N, M')$, such that $M''[t'']$. This contradicts the condition that $t^* \in T$ is a dead transition. Thus, $J \neq \emptyset$.

Then, prove $\exists M' \in R(N, M)$ such that for all $j \in J$, $\Omega = \{p^* \mid p \in P_A, M'(p) > 0\}$ is a set of dead transitions. From the marking M' mentioned before and move all processes one by one corresponding to indexes in J as follows approach. By Lemma 10, for all $p_j \in P_{A_j}$ there must exist a simple path $SP(p_j, p_j^0)$ from p_j to p_j^0 . Then, for all $p_j \in \{p \in P_{A_j}, M'(p) > 0\}$, moving the process along $SP(p_j, p_j^0)$, the process will be either dead (since by Lemma 9, all the transitions in the postset of a choice place are disabled as long as one of them is disabled) or back to initial state, in which the tokens in p_j will flow back to p_j^0 . Repeat this for each process corresponding to indexes in J . The net reaches a marking $M'' \in R(N, M')$ verifying condition (ii) in the hypothesis, because, on the contrary, all process corresponding to indexes in J will get back to the initial state; that is, $M_0 \in R(N, M')$ which is in contradiction with Lemma 12. Let $M' := M''$; Lemma 13 holds. \square

From these lemmas, the main theorems can be obtained as follows.

Theorem 14. Let (N, M_0) be a well-marked OSC-S*PR net, $M \in R(N, M_0)$, and $t \in T$ is a dead transition for M . Then $\exists M' \in R(N, M)$, $\exists S$ is a siphon such that S is nonempty and non-max'-marked.

Proof. Consider the making M' given in Lemma 13. Let

$$\begin{aligned} S'_R &= \{r \in P_R \mid \exists t \in r^*, M'(r) < w(r, t), \\ &\quad p \in P_A \cap \bigcap t, M'(p) > 0\}; \\ S'_P &= \{p \in H(r) \mid r \in S_R, M'(p) = 0\}. \end{aligned} \quad (2)$$

S'_R is the set of resource places which have at least one disabled output transition and this transition is disabled only because of the lack of this resource. S'_P is the set of nonmarked places in $H(r)$. It is apparent that $S \neq \emptyset$ (otherwise $S'_R = \emptyset$, then for all $p \in P_A \cap \{p \mid M'(p) > 0\}$, it is true that for all $t' \in p^*$, $M'[t']$, which contradicts the condition that Ω is a set of dead transitions).

Now we prove that $S = S'_R \cup S'_P$ is a non-max'-marked siphon under M' .

First, we prove that S is a siphon; that is, for all $t' \in \cdot S$, $t' \in S^*$. Then, two cases should be considered.

Case 1. $t' \in \cdot r$, where $r \in S'_R$. By Definition 1, the subnet of an S*PR net is a state machine. Thus, $|\cdot t' \cap P_A| = 1$. Let $\cdot t' \cap P_A = \{p'\}$. Since $t' \in \cdot r$, it can be obtained that $p' \in H(r)$. Then, two subcases should be considered as follows.

Subcase 1.1. If $M'(p') = 0$, by the definition of S'_P , it can be obtained that $p' \in S'_P$. Therefore, $t' \in S'_P \subseteq S^*$.

Subcase 1.2. If $M'(p') \neq 0$, by the definition of Ω in Lemma 13, $t' \in \Omega$. By Lemma 13, t' is a dead transition for M' . Therefore, $\exists r' \in \cdot t' \cap P_R$ satisfies $M'(r') < W(r', t')$. By the definition of S'_R , it can be obtained that $r' \in S_R$. Thus, $t' \in S'_R \subseteq S^*$.

Case 2. $t' \in \cdot p$, where $p \in S'_P$. By the definition of S'_P , $\exists r \in S'_R$, such that $p \in H(r)$. If $t' \in r^*$, then $t' \in S'_R \subseteq S^*$. If $t' \notin r^*$, since $p \in H(r)$, there must exist $p' \in H(r) \cap P_A$, such that $t' \in (p')^*$. Similarly, the two following subcases should be considered.

Subcase 2.1. If $M(p') = 0$, by definition of S'_P , it can be obtained that $p' \in S'_P$. Therefore, $t' \in S'_P \subseteq S^*$.

Subcase 2.2. If $M(p') \neq 0$, similar to the proof in Subcase 1.2, $t' \in S'_R \subseteq S^*$ holds.

In conclusion, $\forall t' \in \cdot S, t' \in S^*$. Thus, S is a siphon.

Then, we prove siphon S is non-max'-marked under M' . $\forall p \in S'_P$, by definition of S'_P , $M'(p) = 0$. $\forall r \in S'_R$, by definition of S'_R , $\exists t \in r^*$ such that $M'(r) < w(r, t)$. Thus, by Definition 8, siphon S is non-max'-marked under M' .

In summary, $\exists M' \in R(N, M)$ and $\exists S$ is a siphon such that S is nonempty and non-max'-marked. \square

Theorem 15. Let (N, M_0) be a well-marked OSC-S*PR net. (N, M_0) is live if all siphons of (N, M_0) are max'-controlled.

Proof. Assume that each siphon in (N, M_0) is max'-controlled while (N, M_0) not live. Then, $\exists M \in R(N, M_0)$ and $\exists t \in T$ is a dead transition under M . By Theorem 14, $\exists M' \in R(N, M)$, $\exists S$ is a siphon which is non-max'-marked under M' . This contradicts the assumption that (N, M_0) is max'-controlled. Thus (N, M_0) is live. \square

It should be noticed that Theorem 15 is a sufficient liveness condition for OSC-S*PR nets. Thus, there may be some live OSC-S*PR nets which are not max'-controlled.

4. Liveness Test Algorithm for OSC-S*PR Nets

In this section, an MIP-based liveness test algorithm is proposed, which is based on Theorem 15 in Section 3. The operators set of an operation place is introduced here, which is similar to [14].

Definition 16. Let N be an S*PR net. For $p \in P_A$, $R_p = \{r \mid r \in P_R, p \in H(r)\}$ the resource places that are required by p are called the set of operators of p .

According to Theorems 14 and 15, the following theorem can be obtained, which is also a liveness test algorithm for testing liveness of OSC-S*PR nets.

Theorem 17 (Algorithm 1). *Let (N, M_0) be a well-marked S*PR net. Then, the minimal non-max'-marked siphon S under M is determined by $S = \{p \in P \mid s_p = 1\}$, where s_p and M are obtained by the following MIP formulation:*

$$\min \sum_{p \in P \setminus P^0} s_p \quad (3)$$

such that

$$|t^*| \sum_{p \in t^*} s_p \geq \sum_{p \in t^*} s_p, \quad t \in T, \quad p \in P \quad (4)$$

$$M = M_0 + [N] \cdot Y, \quad M \geq 0, \quad Y \geq 0 \quad (5)$$

$$\sum_{p \in P^0} s_p = 0 \quad (6)$$

$$\sum_{p \in P_A} s_p \geq 1 \quad (7)$$

$$\sum_{p \in P_R} s_p \geq 1 \quad (8)$$

$$p \in P_A, t \in P^*$$

$$e_t \geq \frac{M(p)}{B(p)} \quad (9)$$

$$M(p) \geq e_t \quad (10)$$

$$s_p + e_t \leq 1 \quad (11)$$

$$r \in P_R, t \in r^*, p \in P_A \cap t^*, r' \in R_p$$

$$(2s_r - 1) \cdot M(r)$$

$$\leq (2s_r - 1)$$

$$\cdot \left\{ \max_{t \in r^*} \left[s_r \cdot (1 - s_p) \cdot \left(\max_{r' \in R_p} s_{r'} \right) \cdot W(r, t) \right] - s_r \right\} \quad (12)$$

$$s_p, e_t \in \{0, 1\} \quad (13)$$

$$Y(p) \in \mathbf{N}, \quad (14)$$

where $M \in R(N, M_0)$, $B(p)$ is the structural bound of place p .

Then, the minimal non-max'-marked siphon is the set of places with $s_p = 1$.

Proof. Let S be a non-max'-marked siphon. By Definition 8, the following conclusions hold:

- (i) for all $p \in S_A$, $M(p) = 0$;
- (ii) for all $r \in S_R$, $M(r) < \max_{t \in r^* \cap [S]^*} W(r, t)$.

Then, let us explain the meaning of the constraints one by one.

Objective (3) ensures that the siphon is a minimal siphon.

Constraint (4) ensures that s_p is the characteristic vector of the siphon [22].

Constraint (5) ensures that $M \in R(N, M_0)$. This is trivial.

Constraints (6)–(8) ensure that the siphon contains no idle place and has one operation place and one resource place at least.

In constraints (9)–(11), $p \in P_A$, $t \in P^*$. e_t means whether transition t is disabled by the operation place p which is in the preset of it. If t is disabled by the operation place p , $e_t = 0$. Otherwise, $e_t = 1$. This comes from the following facts. If $M(p) \geq 1$, then transition t is enabled by the operation place p . From constraint (9) and definition of structural bound $B(p)$ in Section 2, it can be obtained that $e_t \geq M(p)/B(p) > 0$. Since $e_t \in \{0, 1\}$, we have $e_t = 1$, which follows the meaning of e_t ; if $M(p) = 0$, then transition t is disabled by the operation place p . From constraints (9) and (10), it can be obtained that $e_t \geq M(p)/B(p) = 0$ and $e_t \geq M(p) = 0$. Thus, $e_t = 1$ which follows the meaning of e_t .

Then, constraint (11) ensures that in the siphon S condition (i) holds. This comes from the following fact: for all $p \in S_A$, by the meaning of s_p , we have $s_p = 1$. By constraint (11), it can be obtained that $e_t \leq 1 - s_p = 0$. Since $s_p, e_t \in \{0, 1\}$, we have $e_t = 0$. By the conclusions above, it can be obtained that transition t is disabled by the operation place p and $M(p) = 0$. This follows condition (i).

Constraint (12) ensures that in siphon S condition (ii) will hold. Since $s_p \in \{0, 1\}$, the following two cases should be considered.

Case 1. If $s_r = 0$, that is, r is not an element of siphon S , then $\max_{t \in r^*} [s_r \cdot (1 - s_p) \cdot (\max_{r' \in R_p} s_{r'}) \cdot W(r, t)] - s_r = 0$. Constraint (10) is equivalent to $M(r) \geq 0$ which is trivial and will not add extra constraints to the formulations.

Case 2. If $s_r = 1$, that is, r is an element of siphon S , then constraint (10) is equivalent to

$$M(r) \leq \max_{t \in r^*} \left[(1 - s_p) \left(\max_{r' \in R_p} s_{r'} \right) \cdot W(r, t) \right] - 1. \quad (15)$$

Then we explain that constraint (15) indicates condition (ii). Two cases should be considered. If $p \in [s]$, that is, $t \in r^* \cap [S]^*$, then we can obtain that $p \notin S$ and $p \in H(r')$. Since $p \notin S$, $(1 - s_p) = 1$. By Definition 16 and $p \in H(r')$, it can be obtained that $(\max_{r' \in R_p} s_{r'}) = 1$. Thus, (15) can be transferred into $M(r) \leq \max_{t \in r^*} W(r, t) - 1$, which means $M(r) < \max_{t \in r^* \cap [S]^*} W(r, t)$. If $p \notin [s]$, that is, $t \notin r^* \cap [S]^*$, then we can obtain that $p \in S$ or $p \notin H(r')$. Similarly, it can be obtained that $(1 - s_p) = 0$ or $(\max_{r' \in R_p} s_{r'}) = 0$. Then, $(1 - s_p)(\max_{r' \in R_p} s_{r'}) \cdot W(r, t) = 0$, this kind of transitions will not be taken into comparison [14].

Therefore, constraint (12) ensures that in the siphon S condition (ii) holds.

In conclusion, constraints (11) and (12) ensure that the obtained siphon satisfies conditions (i) and (ii), that is, is a non-max'-marked siphon.

In summary, if the MIP formulation has a feasible solution, the obtained siphon is the minimal non-max'-marked siphon. \square

Thus, Theorem 17 can find a non-max'-controlled siphon at a bad marking. It utilizes some MIP tools to find non-max'-controlled siphons rather than traverse the state space of the net. Following Theorem 17, Corollary 18 can be obtained.

Corollary 18. *Let (N, M_0) be a well-marked OSC-S*PR net. It is live if Algorithm 1 has no feasible solution.*

Proof. For a well-marked OSC-S*PR net, Algorithm 1 having no feasible solution means that all siphons of this net are max'-controlled. By Theorem 15, it is live. \square

Example 19. Applying Algorithm 1 to the OSC-S*PR net in Figure 1 and solving by Lingo [23] no feasible solution can be found. By Corollary 18, it is live, which follows the fact that it is a live net.

5. Liveness Test for S*PR Nets

In order to test liveness of the S*PR nets, an indirect method is given in this section, which comprises two steps as follows: first, for a given S*PR net, constructing an OSC-S*PR net to ensure that if the latter is live then the former must be live; second, testing liveness of the constructed OSC-S*PR net by the aforementioned MIP-based algorithm.

Remark 20. In the following sections, we assume that for all $p_{0i} \in P_0$, $|p_{0i}^*| = 1$.

Remark 20 restricts that each process in the net can only have one start action.

In the following, the constructing algorithm is proposed.

Insertion 1. Let (N, M_0) be a well-marked S*PR net. For all $p \in P$, $t \in p^*$, a pair of virtual place and virtual transition $\{p_v, t_v\}$ is inserted into (p, t) as follows:

- (i) $F = (F \cup (p, t_v) \cup (t_v, p_v) \cup (p_v, t)) / (p, t)$;
- (ii) $P_A = P_A \cup \{p_v\}$, $T = T \cup \{t_v\}$;
- (iii) $w(p, t_v) = 1$, $w(t_v, p_v) = 1$, $w(p_v, t) = 1$;
- (iv) $M_0(p_v) = 0$.

Then, for a given S*PR net, an OSC-S*PR net can be constructed by the following algorithm.

Algorithm 2

Input. S*PR net (N, M_0) .

Output. A constructed net $(\widehat{N}, \widehat{M}_0)$.

Step 1. $\Omega := P_C$, $P_v := \phi$, $T_v := \phi$, $(N', M'_0) := (N, M_0)$;

Step 2. while $\Omega \neq \phi$, for the net (N', M'_0) do the following:

Step 2.1. Post $:= p^*$:

Step 2.2. while $Post \neq \phi$, do the following:

Step 2.1.1. for all $t \in Post$, insert $\{p_{vi}, t_{vi}\}$ to (p, t) by Insertion 1 and obtain (N'', M''_0) ;

Step 2.1.2. $Post = Post / \{t\}$, $P_v = P_v \cup \{p_{vi}\}$, $T_v = T_v \cup \{t_{vi}\}$, $i+ = 1$, $(N', M'_0) := (N'', M''_0)$;

endwhile

Step 2.3. $\Omega := \Omega / \{p\}$;

endwhile

Step 3. $(\widehat{N}, \widehat{M}_0) := (N', M'_0)$;

Step 4. Output the constructed net $(\widehat{N}, \widehat{M}_0)$.

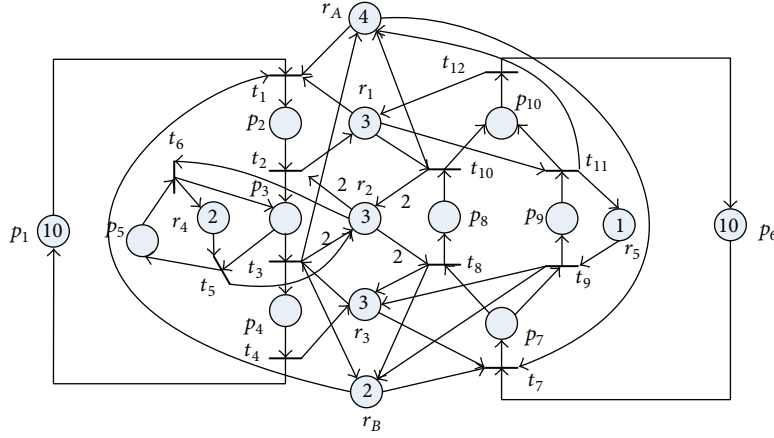
Since P_C and p^* are finite sets, it can be obtained that Algorithm 2 will end in finite steps. What is more, $(\widehat{N}, \widehat{M}_0) = (P \cup P_v, T \cup T_v, \widehat{F}, \widehat{W}, \widehat{M}_0)$ is called the constructed net of (N, M_0) , and (N, M_0) is called original S*PR net of $(\widehat{N}, \widehat{M}_0)$.

Remark 21. By Algorithm 2 and Insertion 1, it is trivial that for all $p \in P_v$, $\widehat{M}_0(p) = 0$ and for all $p \in P$, $\widehat{M}_0(p) = M_0(p)$; that is, $\widehat{M}_0(p) = \sum_{p \in P} M_0(p)p$.

Lemma 22. *Let (N, M_0) be a well-marked S*PR net and $(\widehat{N}, \widehat{M}_0)$ is the constructed net obtained by Algorithm 2. Then, $(\widehat{N}, \widehat{M}_0)$ is a well-marked OSC-S*PR net.*

Proof. First, we prove that $(\widehat{N}, \widehat{M}_0)$ is a well-marked S*PR net. By Insertion 1, the virtual places are considered as operation places. Thus, it is apparent that $(\widehat{N}, \widehat{M}_0)$ satisfies conditions (i), (ii), (iii), (iv), and (v) in Definition 1. Then, we prove that it satisfies conditions (vi) and (vii). Without loss of generality, assume that $p_1 \in P_C$ and $t_1 \in p_1^*$ in (N, M_0) . Then by Insertion 1, p_v and t_v will be inserted to (p_1, t_1) . Since $t_v = \{p_1\}$, $t_v^* = \{p_v\}$ and $p_v^* = \{t_1\}$, it can be obtained that no self-loop is inserted to $(\widehat{N}, \widehat{M}_0)$ by Insertion 1. This is valid for each insertion in Algorithm 2. Therefore, it can be concluded that no self-loop will be inserted to $(\widehat{N}, \widehat{M}_0)$; that is, $(\widehat{N}, \widehat{M}_0)$ will still be a pure net. What is more, p_1 and t_1 are still connected by the path p_1, t_v, p_v, t_1 . This is also valid for all the insertions in Algorithms. Hence, $(\widehat{N}, \widehat{M}_0)$ will still be strongly connected; that is, $(\widehat{N}, \widehat{M}_0)$ satisfy condition (vii) in Definition 1. Since t_v do not attach to any resources, p_v will use the resources that $p \in P_C \cap t_v^*$ has used. Therefore, $(\widehat{N}, \widehat{M}_0)$ satisfies condition (vi) of Definition 1. Henceforth, $(\widehat{N}, \widehat{M}_0)$ is an S*PR net. What is more, by Remark 21, the initial marking of virtual place is zero and the initial marking of other places stays the same. Thus, regarding virtual places as members of operation places, $(\widehat{N}, \widehat{M}_0)$ satisfies Definition 2 as well; that is, $(\widehat{N}, \widehat{M}_0)$ is a well-marked S*PR net.

Second, we prove that $(\widehat{N}, \widehat{M}_0)$ is an OSC-S*PR net. Considering Step 2.1 in Algorithm 2 with Remark 20, it is guaranteed that for all $p \in P_A$ if $|p^*| \geq 2$, then for all $t \in p^*$, $t = \{p\}$ and $t \in T_v$. Thus $(\widehat{N}, \widehat{M}_0)$ satisfies Definition 5; that is, $(\widehat{N}, \widehat{M}_0)$ is an OSC-S*PR net.

FIGURE 2: An S^*PR net.

In summary, $(\widehat{N}, \widehat{M}_0)$ is a well-marked OSC- S^*PR net. \square

Example 23. The Petri net plant in Figure 2 is an S^*PR . Since $p_3^* = \{t_3, t_5\}$, $t_3 \neq t_5$ and $p_7^* = \{t_8, t_9\}$, $t_8 \neq t_9$, it does not satisfy the OSC-Condition. By Algorithm 2, the constructed net is shown as Figure 1, and it is an OSC- S^*PR net.

Then, we will prove that the original S^*PR net must be live if its constructed net is live.

Theorem 24. Let (N, M_0) be an S^*PR net and let $(\widehat{N}, \widehat{M}_0)$ be its constructed net. If $(\widehat{N}, \widehat{M}_0)$ is live, (N, M_0) is live.

Proof. The liveness of Petri nets is determined by the firing sequence of transitions and the number of tokens consumed and produced by transitions. If the each firing sequence of (N, M_0) has a corresponding firing sequences of $(\widehat{N}, \widehat{M}_0)$ and the number of tokens consumed and produced by those firing sequences is the same, then it can be obtained that if $(\widehat{N}, \widehat{M}_0)$ is live, (N, M_0) is live. In other words, (N, M_0) must retain the number or direction of flow of tokens in $(\widehat{N}, \widehat{M}_0)$. What is more, by Remark 21, the initial making of (N, M_0) and $(\widehat{N}, \widehat{M}_0)$ satisfies $\widehat{M}_0(p) = \sum_{p \in P} M_0(p)p + \sum_{p_v \in P_v} 0 * p_v = \sum_{p \in P} M_0(p)p$. Thus, this theorem could be proved by checking whether each firing sequence of (N, M_0) has a corresponding firing sequence in $(\widehat{N}, \widehat{M}_0)$ which consumes and produces the same number of tokens in the same places.

The only difference between an S^*PR net and its constructed net is the virtual transitions and virtual places that are inserted by Algorithm 2. Without loss of generality, we could assume that in (N, M_0) , $p \in P_C$, $t \in p^*$ and in $(\widehat{N}, \widehat{M}_0)$, $t_v = \{p\}$, $t_v^* = \{p_v\}$ and $p_v^* = \{t\}$. In the firing sequence of transitions, the insertion means a replacement of the firing of t by a subfiring sequence $t_v t$. From the viewpoint of token number, the subfiring sequence produces tokens to each $p' \in t^*$ by the weight of $w(t, p')$ and it demands tokens from $r \in t \cap P_R$ and p by the weight of $w(r, t)$ and $w(p, t_v)$ (the token in p_v remains zero). Since in $(\widehat{N}, \widehat{M}_0)$, $w(p, t_v) = 1$ which is equal to $w(p, t) = 1$ in (N, M_0) , the number of tokens demand and produced by the subfiring sequence is

equal to that number of transition t . Therefore, the number of tokens in each place changed by firing $t_v t$ is the same as that of firing t . This is valid for each $t \in P_C^*$. Therefore, for any firing sequence σ of (N, M_0) , each transition t_i (which satisfies $t_i \in (P_C)^*$) in σ is replaced by the subfiring sequence $t_v t_i$, a new firing sequence $\widehat{\sigma}$ will be obtained for $(\widehat{N}, \widehat{M}_0)$, and the token in each place changed by firing $\widehat{\sigma}$ must be the same to that of firing σ (the token in p_v remains zero). Thus, it can be obtained that if $(\widehat{N}, \widehat{M}_0)$ is live, then (N, M_0) is live. \square

So, the liveness of an S^*PR net (N, M_0) can be tested the algorithm shown as follows.

Algorithm 3

Input. S^*PR net (N, M_0) .

Output. a non-max'-marked siphon S .

Step 1. Construct an OSC- S^*PR $(\widehat{N}, \widehat{M}_0)$ from (N, M_0) by Algorithm 2;

Step 2. Obtain a non-max'-marked siphon S of $(\widehat{N}, \widehat{M}_0)$ by Algorithm 1;

Step 3. Output S of $(\widehat{N}, \widehat{M}_0)$.

Theorem 25. Let (N, M_0) be a well-marked S^*PR net. If Algorithm 3 has no feasible solution, then (N, M_0) is live.

Proof. By Algorithm 3, (N, M_0) will be transformed to its constructed $(\widehat{N}, \widehat{M}_0)$. By Lemma 22, $(\widehat{N}, \widehat{M}_0)$ is an OSC- S^*PR net. By Step 2 and Step 3, Algorithm 3 has no feasible solution which means that applying Algorithm 1 to $(\widehat{N}, \widehat{M}_0)$ there is no feasible solution obtained. Then, by Corollary 18, $(\widehat{N}, \widehat{M}_0)$ is live. Therefore, by Theorem 24, it can be obtained that (N, M_0) is live. \square

Example 26. The Petri net model in Figure 2 is an S^*PR net, but not an OSC- S^*PR net. The choice places p_3 and p_7 dissatisfy the OSC-Condition. By Algorithm 2, $\{t_{v1}, p_{v1}\}$

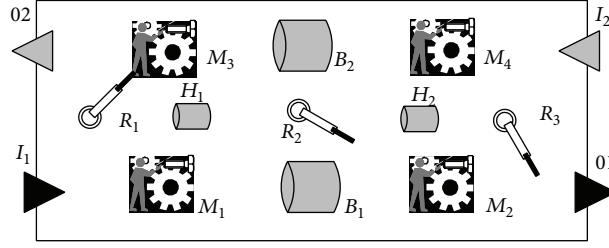
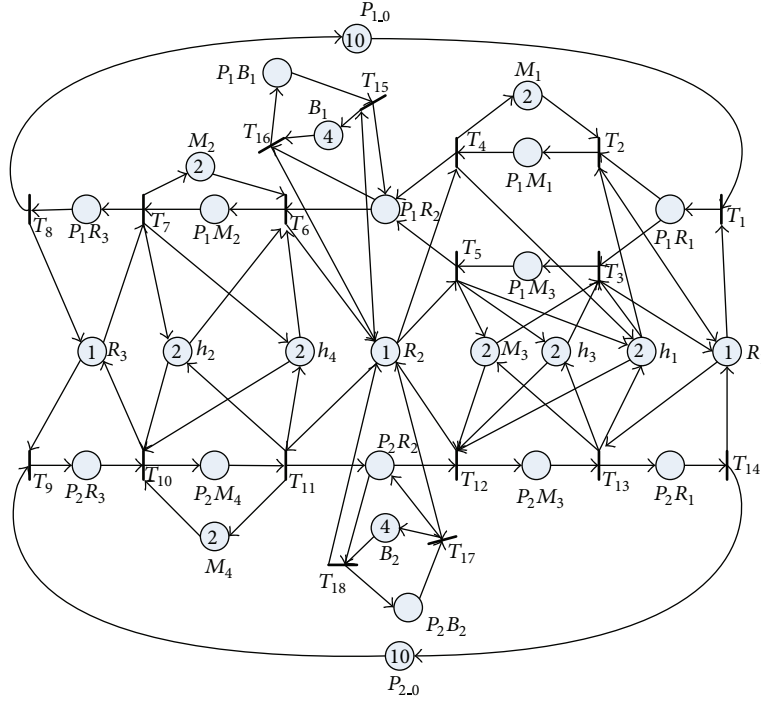


FIGURE 3: Layout of a manufacturing cell.

FIGURE 4: An S^*PR net modeling the system in Figure 3.

and $\{t_{v2}, p_{v2}\}$ are inserted at the rear of p_3 , and $\{t_{v3}, p_{v3}\}$ and $\{t_{v4}, p_{v4}\}$ are inserted at the rear of p_7 . The constructed net is obtained as Figure 1. Then, by Algorithm, it can be obtained that the constructed net is live. By Theorem 24, the original in Figure 2 is live.

6. Examples

In this section, a typical example (shown as Figure 3) is quoted to illustrate Algorithm 2, which is a classic FMS cell with 2 buffers, 4 machines, and 3 robots [15]. And it can be modeled as Figure 4.

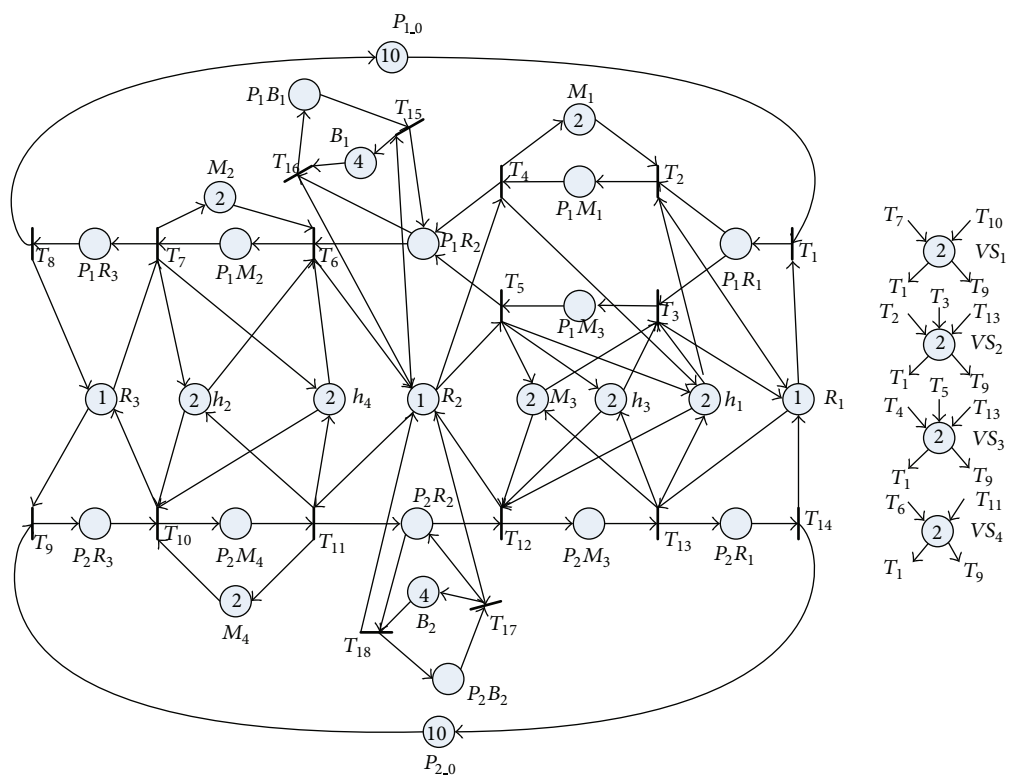
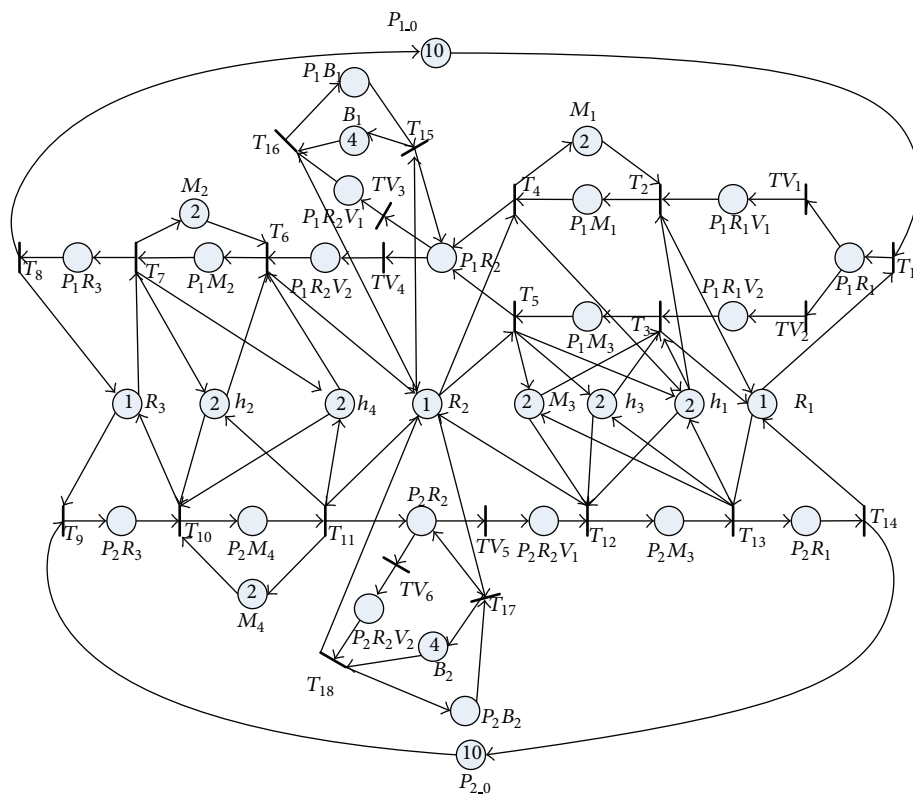
Since the choice operation places P_1R_1 , P_1R_2 , and P_2R_2 dissatisfy the OSC-Condition, the net in Figure 4 is not an OSC- S^*PR net. By the Step 2.1 of Algorithm 2, pairs of virtual transitions and places should be inserted at the rear of each branch of choice places; in other words, $\{TV_1, P_1R_1V_1\}$, $\{TV_2, P_1R_1V_2\}$, $\{TV_3, P_1R_2V_1\}$, $\{TV_4, P_1R_2V_2\}$, $\{TV_5, P_2R_2V_1\}$, and $\{TV_6, P_2R_2V_2\}$ are inserted. Then, the constructed net is shown as Figure 5.

After applying the MIP formulation in Theorem 17 to Figure 5 and solving it with LINGO [23] we obtain a minimal non-max'-marked siphon $S = P_1R_3 + P_2M_4 + H_4 + R_3$, which follows the fact that the net is nonlive.

Generally, some structural controllers can be added to enforce the liveness of the net (referring to [9, 10, 15, 21]). Here, four controllers VS_1 , VS_2 , VS_3 , and VS_4 are added into the net in Figure 5, and the resulted S^*PR net is shown as Figure 6. Then, the constructed net in Figure 7 can be obtained by Algorithm 2. It can be verified that the net in Figure 7 is live. By Algorithm 3, the net in Figure 6 must be live.

7. Conclusions

It is well-known that S^3PR nets and S^4PR nets can model sequential FMS, while full routine flexibilities can be modeled by S^*PR nets. As a subclass of S^*PR nets, OSC- S^*PR nets have some modeling abilities about routine flexibilities with the restriction of each choice operation using the same type



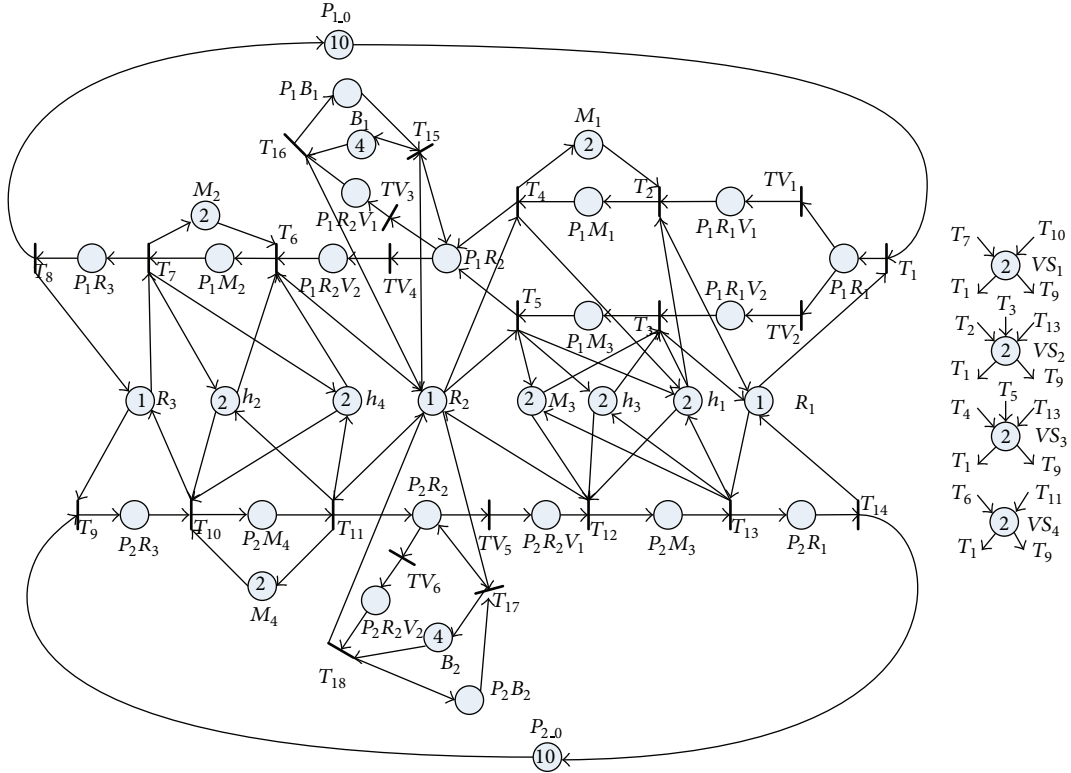
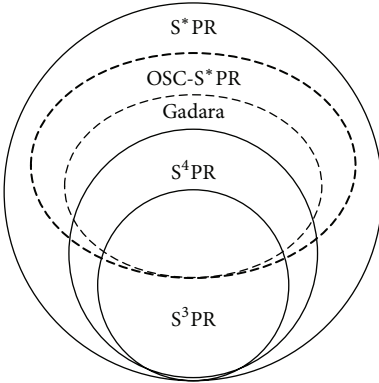


FIGURE 7: The constructed net of Figure 6.

FIGURE 8: Modeling capabilities of some subclasses of S^* PR nets.

and quantity of resources. Obviously, Gadara nets [17] are a special case of OSC- S^* PR nets satisfying no resources needed in all choice operations. Henceforth, their modeling capabilities can be illustrated as Figure 8 approximately.

It is worth to say that Theorem 15 is just a sufficient condition, and some live OSC- S^* PR nets may contain non-max'-controlled siphons. In other words, sometimes Theorem 15 is a little strict though, so there maybe exist ways to loosen the sufficient condition. Moreover, the liveness property of S^* PR nets still cannot be tested directly by using standardized mixed integer programming (MIP) tools in this paper. And we should make great efforts to develop more efficient algorithms for testing liveness of S^* PR nets.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors would like to thank the National Nature Science Foundation of China under Grant no. 61071062, the Provincial Nature Science Foundation of Zhejiang under Grant no. Y12F02030, and the School Funding of Hangzhou Normal University under Grant no. 2013JGJ001.

References

- [1] G. Liu, D. Y. Chao, and M. Uzam, "Maximally permissive deadlock prevention via an invariant controlled method," *International Journal of Production Research*, vol. 51, no. 15, pp. 4431–4442, 2013.
- [2] J. Ezpeleta, F. García-Vallés, and J. M. Colom, "A class of well structured Petri nets for flexible manufacturing systems," in *Application and Theory of Petri Nets*, pp. 64–83, Springer, Berlin, Germany, 1998.
- [3] M. D. Byrne and P. Chutima, "Real-time operational control of an FMS with full routing flexibility," *International Journal of Production Economics*, vol. 51, no. 1-2, pp. 109–113, 1997.
- [4] T. Murata, "Petri nets: properties, analysis and applications," *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541–580, 1989.
- [5] F. Chu and X.-L. Xie, "Deadlock analysis of Petri nets using siphons and mathematical programming," *IEEE Transactions*

- on *Robotics and Automation*, vol. 13, no. 6, pp. 793–804, 1997.
- [6] F. Tricas, F. Garcia-Valles, J. M. Colom, and J. Ezpeleta, “A structural approach to the problem of deadlock prevention in process with resources,” in *Proceedings of the 4th International Workshop on Discrete Event Systems (WODES '98)*, pp. 273–278, Cagliari, Italy, August 1998.
 - [7] J. Ezpeleta, J. M. Colom, and J. Martinez, “A petri net based deadlock prevention policy for flexible manufacturing systems,” *IEEE Transactions on Robotics and Automation*, vol. 11, no. 2, pp. 173–184, 1995.
 - [8] J. Ezpeleta, F. García-Vallés, and M. Colom J, “A class of well structured Petri nets for flexible manufacturing systems,” in *Application and Theory of Petri Nets*, pp. 64–83, Springer, Berlin, Germany, 1998.
 - [9] Z. Li and M. Shpitalni, “Smart deadlock prevention policy for flexible manufacturing systems using Petri nets,” *IET Control Theory and Applications*, vol. 3, no. 3, pp. 362–374, 2009.
 - [10] Y. Huang, M. Jeng, X. Xie, and S. Chung, “Deadlock prevention policy based on Petri nets and siphons,” *International Journal of Production Research*, vol. 39, no. 2, pp. 283–305, 2001.
 - [11] A. Giua and C. Seatzu, “Liveness enforcing supervisors for railway networks using ES^2 PR Petri nets,” in *Proceedings of the 6th IEEE International Workshop on Discrete Event Systems*, pp. 55–60, 2002.
 - [12] J. Park and S. A. Reveliotis, “Deadlock avoidance in sequential resource allocation systems with multiple resource acquisitions and flexible routings,” *IEEE Transactions on Automatic Control*, vol. 46, no. 10, pp. 1572–1583, 2001.
 - [13] Y.-Y. Shih, D. Y. Chao, and C.-Y. Chiu, “A new MIP test for S^3 PGR²,” in *Global Perspective for Competitive Enterprise, Economy and Ecology*, Advanced Concurrent Engineering, pp. 41–52, Springer, London, UK, 2009.
 - [14] G. Liu and Z. Li, “General mixed integer programming-based liveness test for system of sequential systems with shared resources nets,” *IET Control Theory & Applications*, vol. 4, no. 12, pp. 2867–2878, 2010.
 - [15] J. Ezpeleta, F. Tricas, F. García-Vallés, and J. M. Colom, “A banker’s solution for deadlock avoidance in FMS with flexible routing and multiresource states,” *IEEE Transactions on Robotics and Automation*, vol. 18, no. 4, pp. 621–625, 2002.
 - [16] Z. Zhang and W. Wu, “Sequence control of essential siphons for deadlock prevention in petri nets,” *Transactions on Embedded Computing Systems*, vol. 12, article 8, no. 1, 2013.
 - [17] Y. Wang, H. Liao, S. Reveliotis, T. Kelly, S. Mahlke, and S. Lafor-tune, “Gadara nets: modeling and analyzing lock allocation for deadlock avoidance in multithreaded software,” in *Proceedings of the 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference (CDC/CCC '09)*, pp. 4971–4976, Shanghai, China, December 2009.
 - [18] H. Liao, Y. Wang, H. K. Cho et al., “Concurrency bugs in multithreaded software: modeling and analysis using Petri nets,” *Discrete Event Dynamic Systems: Theory and Applications*, vol. 23, no. 2, pp. 157–195, 2013.
 - [19] Z. W. Li and M. C. Zhou, *Deadlock Resolution in Automated Manufacturing Systems: A Novel Petri Net Approach*, Springer, London, UK, 2009.
 - [20] D. Y. Chao, “Max'-controlled siphons for liveness of S^3 PGR²,” *IET Control Theory and Applications*, vol. 1, no. 4, pp. 933–936, 2007.
 - [21] C. Zhong and Z. Li, “Self-liveness of a class of Petri net models for flexible manufacturing systems,” *IET Control Theory & Applications*, vol. 4, no. 3, pp. 403–410, 2010.
 - [22] F. Basile, P. Chiacchio, A. Giua, and C. Seatzu, “Deadlock recovery of Petri net models controlled using observers,” in *Proceedings of the 8th International Conference on Emerging Technologies and Factory Automation (ETFA '01)*, vol. 2, pp. 441–449, Antibes-Juan les Pins, France, October 2001.
 - [23] LINGO, “Premier optimization modeling tools,” <http://www.lingo.com/>.