On calculation of state space for linear system of simple sequential processes with resources

Liang Hong¹, XiaoHua Wang¹, Xun Li¹, ZeBin Su¹, Dan Zhang¹ and YuFeng Chen²

Abstract
This work focuses on the calculation of state space for linear system of simple sequential processes with resources (LS³PR). The method, different from reachability graphs rendering high computational complexity, finds reachable markings by combinatorics. First, the set of invariant markings can be obtained by combinatorics if the influence of deadlocks is ignored. Unfortunately, some of reachable states in the set of invariant markings are proved to be spurious if deadlocks exist. Second, we find the spurious markings by computing a set of minimal spurious markings, which can be calculated by a proposed algorithm based on strict minimal siphons. The obtained spurious markings are proved to cover all the spurious markings of the LS³PR. Removing the spurious markings from the set of invariant markings, the left ones constitute the state space of the LS³PR. The detailed method is shaped to an algorithm. The effectiveness of the algorithm is proved by example calculation and analysis. Finally, we analyze the computational complexity of the proposed method compared with reachability graphs.

Keywords
Flexible manufacturing system, Petri net, reachable marking, siphon, combinatorics

Date received: 6 October 2016; accepted: 3 March 2017

Academic Editor: Murat Uzam

Introduction
Different kinds of products can be made in a flexible manufacturing system (FMS) by computer control based on the allocation of a limited number of shared resources. Due to the existence of competition for resources, deadlocks are always unavoidable. Petri nets, as a tool with powerful modeling and analysis abilities, are widely used in the deadlock prevention problems of resource allocation systems.¹,² Petri nets–based deadlock prevention uses an offline computation mechanism to impose constraints on a system to prevent the system from reaching deadlock states.

Almost without exception, the existing literatures try to evaluate the performance of liveness-enforcing supervisors in terms of behavioral permissiveness, computational complexity, and structural complexity. Reachability graph (RG) analysis is an important technique for deadlock control, based on which an optimal or nearly optimal supervisor with high behavioral permissiveness can always be obtained. Depending on the calculation of state space, the theory of regions can definitely find an optimal liveness-enforcing supervisor if it exists.³ The theory of regions is used in previous

¹School of Electronics and Information, Xi’an Polytechnic University, Xi’an, China
²School of Electro-Mechanical Engineering, Xidian University, Xi’an, China

Corresponding author:
Liang Hong, School of Electronics and Information, Xi’an Polytechnic University, No. 19 Jinhua South Road, Xi’an 710048, China.
Email: chelseagoon@gmail.com

Creative Commons CC-BY: This article is distributed under the terms of the Creative Commons Attribution 3.0 License (http://www.creativecommons.org/licenses/by/3.0/) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (https://us.sagepub.com/en-us/nam/open-access-at-sage).
studies\textsuperscript{4–6} for an S\textsuperscript{3}PR\textsuperscript{7} to obtain a suboptimal liveness-enforcing supervisor.

However, reachability analysis\textsuperscript{8–10} needs a reachable marking enumeration, which requires a large number of consumptions of resources. In some cases, the analysis may stop due to the exhausted memory. It is shown in the work by Lipton\textsuperscript{11} that the complexity of the reachability problem of a Petri net is exponential. Currently, using the popular integrated net analyzer (INA)\textsuperscript{12} to obtain the state space of a large Petri net model is difficult, even impossible. Large sets of reachable markings with shared data structures can be represented by binary decision diagrams (BDDs). Besides, BDDs can implement efficient manipulation on the sets of reachable markings. In order to overcome the state explosion problem, BDD is used by Chen et al.\textsuperscript{13} to represent the state space. However, the method still requires a complete enumeration of reachable markings whose number increases exponentially with the size of a Petri net. Hence, state explosion problem is still a technique barrier of using RG analysis to prevent deadlocks in an FMS. Consequently, we figure out another way with LS\textsuperscript{3}PR by combinatorics. Then, excluding all spurious markings, one can obtain the state space of a large Petri net model is difficult, even impossible. Large sets of reachable markings which require a large number of consumptions of resources. In some cases, the analysis may stop due to the exhausted memory. It is shown in the work by Lipton\textsuperscript{11} that the complexity of the reachability problem of a Petri net is exponential.

Preliminaries

Petri nets

A Petri net is a four-tuple $N = (P, T, F, W)$, where $P$ and $T$ are finite, nonempty, and disjoint sets. $P$ is a set of places and $T$ is a set of transitions. $F \subseteq (P \times T) \cup (T \times P)$ is called a flow relation, represented by arcs with arrows from places to transitions or from transitions to places. $W : (P \times T) \cup (T \times P) \rightarrow \mathbb{N}$ is a mapping that assigns a weight to an arc: $W(x, y) > 0$ if $(x, y) \in F$, where $x, y \in P \cup T$ and $\mathbb{N} = \{0, 1, 2, \ldots\}$. If $W(x, y) \leq 1$, $\forall (x, y) \in F$, the net is called an ordinary Petri net, denoted by $N = (P, T, F)$; otherwise, it is called a generalized one. A subnet $N_s = (P_s, T_s, F_s, W_s)$ of $N$ is generated by $X = P_s \cup T_s$, where $P_s = P \cap X$, $T_s = T \cap X$, $F_s = F \cap (X \times X)$, and $\forall f \in F_s$, $W_s(f) = W(f)$.

A marking $m$ is used to denote a marking or state of a Petri net $N$, which is a mapping from $P$ to $\mathbb{N}$. The number of tokens in place $p$ is denoted as $m(p)$. For economy of space, multisets are used to describe markings and vectors. Here, vector $m$ is denoted as $\sum_{p \in P} m(p) \cdot p$. $(N, M_0)$ is called a net system, where $M_0$ is an initial marking of $N$, $M_0 = \sum_{p \in P} M_0(p) \cdot p$ is used to denote a marking of a subnet of $N$ and $S \subseteq P$ is a mapping from $S$ to $\mathbb{N}$ and $M(S)$ is used to denote the sum of tokens in $S$, $M_0^S$ is called an initial marking of $S$ such that $\forall p \in S$, $M_0(p) = M_0(p)$. $x = \{y \in P \cup T \mid (y, x) \in F\}$ and $x^* = \{y \in P \cup T \mid (x, y) \in F\}$ are used to denote the preset and postset of $x$, respectively. Similarly, given $X \subseteq P \cup T$, $x = \cup_{x \in x} x$, and $x^* = \cup_{x \in x} x^*$. A nonempty set $S \subseteq P$ is called a SMS if there is no siphon contained in it and $S \subseteq S^*$. A transition $t \in T$ is enabled at a marking $m$ if $\forall p \in x^t, m(p) \geq W(p, t)$. This fact is denoted as $M[t]$. Firing it yields a new marking $M' = M(p) - W(p, t) + W(t, p)$ as denoted by $M[t]$. $M'$ is called an immediately reachable marking from $M$ and $M$ is called a closely previous reachable marking of $M'$. A marking $M''$ is said to be reachable from $M$ if there exists a sequence of transitions $\sigma = t_0t_1\ldots t_n$ and markings $M_1, M_2, \ldots, M_n$ such that $M[t_0]M_1[t_1]M_2 \cdots M_n[t_n]M''$ holds. The set of markings reachable from $M$ in $N$ is called the reachability set of Petri net $(N, M)$ and denoted as $R(N, M)$. $\|R(N, M)\|$ is used to denote the number of reachable markings in $R(N, M)$. $I$ is called a $P$-invariant if $\forall M \in \mathbb{N}[P], I^T \cdot M = I^T \cdot M_0$. $I_0(N, M_0) = \{M \in \mathbb{N}[P] \mid I^T \cdot M = I^T \cdot M_0\}$ is called the set of IMs. A $P$-invariant $I$ is called minimal $P$-semiflow if it is non-negative and its support $\|I(p)\| = \{p \mid I(p) \neq 0\}$ is not a superset of the support of any other $P$-invariants. Here, $\sum_{p \in \|I(p)\|} I(p) \cdot p$ is used to denote vector $I$. 

The article is organized as follows. Section “Preliminaries” presents the preliminaries used throughout this article. The detailed approach of computing the state space of an LS\textsuperscript{3}PR is introduced and a well-known example of LS\textsuperscript{3}PR is adopted to support the approach in section “Main approach.” We perform an analysis on computational complexity of the proposed method compared with RG in section “Computational complexity analysis and comparison.” Finally, the conclusion of the article is presented in section “Conclusion.”
Let $I$ be a minimal $P$-semiflow. $(N[I], M_0[I])$ is a marked subnet generated by $X = I || I || \cup^* I || I || \cup I || I^*|| M = I_2^T M_0[I]$ and $M_0[I]$ is a mapping from $P_I$ to $N$ with $\sum_{p \in I} M_0(p) \cdot p$ as its multiset form. A $P$-vector of $N[I]$ is a column vector $I_2 : P_I \rightarrow Z$ indexed by $P_I$, where $P_I$ denotes the set of places in $I[I]$. $I_2(N[I], M_0[I]) = \{ M \in \mathbb{N}^{|P_I|} | I_2^T M = I_2^T M_0[I]\}$ denotes the set of IMs of $(N[I], M_0[I])$.

Let $I_i$ be a minimal $P$-semiflow of $N$ and $I = \sum_{i=1}^n I_i$ be a $P$-semiflow of $N$. $I_\Delta$ is a minimal $P$-semiflow of $N[I]$ and $I_\Delta = \sum_{i=1}^n I_\Delta_i$ is a $P$-semiflow of $N[I]$. $I_\Delta$ and $I_\Delta_i$, $i \in \{1, 2, \ldots, n\}$, are column vectors: $P_l \rightarrow Z$ indexed by $P_l$. Let $X_l$ be a matrix where each column is a minimal $P$-semiflow of $N[I]$. $I_\Delta(N[I], M_0[I]) = \{ M \in \mathbb{N}^{|P_l|} | X_l^T M = X_l^T M_0[I]\}$ denotes the set of IMs of $(N[I], M_0[I])$.

Let $X$ be a matrix where each column is a minimal $P$-semiflow of $N$ and denote the set of IMs of $(N, M_0)$ as $I_X(N, M_0) = \{ M \in \mathbb{N}^{|P|} | X^T M = X^T M_0[I]\}$. It is found that $I_X(N[I], M_0[I]) = I_X(N[I], M_0[I])$, if $n = 1$, and $I_X(N[I], M_0[I]) = I_X(N, M_0)$ if $n$ denotes the number of all the minimal $P$-semiflows in $N$. The number of markings in $I_X(N[I], M_0[I])$ is denoted as $[I_X(N[I], M_0[I])]$.

**Definition 3.** A linear $S^3PR$, called LS$^3PR$ for short, is an ordinary Petri net $N = (I^0 \cup P_A \cup P_H, T, F)$ such that:

1. $\forall i \in \{1, \ldots, l\}$, the subnet $N_i$ generated by $\{p_i^0\} \cup P_A \cup T_i$ is a strongly connected state machine, such that every cycle contains place $p_i^0$ and $\forall p \in P_A, \{p\} = 1$.  
2. $\forall i \in \{1, \ldots, l\}, \forall p \in P_A, *p \cap P_R = p^* \cap P_R$ and $**p \cap P_R = 1$.  
3. $N$ is strongly connected.

**Property 1.** Let $N = \bigcap_{i=1}^n N_i = (I^0 \cup P_A \cup P_H, T, F)$ be an LS$^3PR$ consisting of $n$ simple sequential processes and $S$ be a siphon in $N$:

1. Any $p \in P_A$ with $I_p$ as its associated minimal $P$-semiflow, $\| I_p \| = P_d \cup \{p_1^0\}$.  
2. Any resource $r \in P_H$ with $I_r$ as its associated minimal $P$-semiflow, $\| I_r \| = \{r\} \cup H(r)$.
3. $\forall p \in S$, $\exists r \in S_p$, $p \in H(r)$ and $\forall r' \in P_H \backslash \{r\}$, $p \notin H(r')$.
4. $[S] \cup S$ is the support of a $P$-semiflow of $N$.  
5. $[S] = \cup_{i=1}^n [S_i]$, where $[S_i] = [S] \cap P_A$.

**Definition 4.** A Petri net $(N, M_0)$ is quasi-safe if $\forall M \in R(N, M_0), \forall p \in P_A \cup P_R, M(p) \leq 1$.

In this article, quasi-safe LS$^3PR$s are the subclass of Petri nets we focus on enumerating the reachable markings.

**Main approach**

This section aims to elaborate on how to completely enumerate the reachable markings of an LS$^3PR$. First, for clarity, we use a simple example to exhibit the rough calculation of reachable markings by the proposed method. Second, the set of IMs of an LS$^3PR$, obtained in the case of ignoring deadlocks, can be calculated by combinatorics. In the following subsection, we exclude all spurious markings from the set and obtain the reachable markings of the LS$^3PR$. Finally, we give an algorithm as well as a supporting example to shape the detailed approach.

**A simple example**

For the simple LS$^3PR$ in Figure 1, there are two $P$-invariants, $I_1 = p_2 + p_6 + p_7$ and $I_2 = p_1 + p_5 + p_6$, containing resources. Supposing that there exists no deadlock, according to the definition of $P$-invariant, the tokens stayed in resource place initially can flow freely in the support of the $P$-invariant. For instance, we can consider that the token in $p_7$ can flow among $p_2$, $p_6$, and $p_7$ freely. Without considering deadlocks, by
combinatorics, we can list the IM of the LS\textsuperscript{3}PR, as shown in Table 1.

We know that the markings in Table 1 are the reachable markings of the LS\textsuperscript{3}PR if there exists no deadlock. Unfortunately, by reachability analysis, $M_6$ is proved spurious and should be removed. Hence, the reachable markings of the LS\textsuperscript{3}PR in Figure 1 are obtained and shown in Table 2.

Thus, in this article, we place an emphasis on how to steer clear of RG to find the state space of an LS\textsuperscript{3}PR.

**Table 1.** Invariant markings of the LS\textsuperscript{3}PR in Figure 1.

| $p_1$ | $p_2$ | $p_3$ | $p_4$ | $p_5$ | $p_6$ | $p_7$ | $p_8$ |
|-------|-------|-------|-------|-------|-------|-------|-------|
| $M_1$ | 2     | 0     | 0     | 2     | 0     | 0     | 1     |
| $M_2$ | 1     | 1     | 0     | 2     | 0     | 0     | 0     |
| $M_3$ | 2     | 0     | 0     | 0     | 1     | 0     | 1     |
| $M_4$ | 1     | 0     | 1     | 2     | 0     | 0     | 0     |
| $M_5$ | 0     | 1     | 1     | 2     | 0     | 0     | 0     |
| $M_6$ | 1     | 0     | 1     | 1     | 0     | 1     | 0     |
| $M_7$ | 0     | 0     | 0     | 1     | 1     | 0     | 0     |
| $M_8$ | 2     | 0     | 0     | 0     | 1     | 1     | 0     |
| $M_9$ | 2     | 0     | 0     | 0     | 0     | 1     | 0     | 0     |

**Table 2.** Reachable markings of the LS\textsuperscript{3}PR in Figure 1.

| $p_1$ | $p_2$ | $p_3$ | $p_4$ | $p_5$ | $p_6$ | $p_7$ | $p_8$ |
|-------|-------|-------|-------|-------|-------|-------|-------|
| $M_1$ | 2     | 0     | 0     | 2     | 0     | 0     | 1     |
| $M_2$ | 1     | 1     | 0     | 2     | 0     | 0     | 0     |
| $M_3$ | 2     | 0     | 0     | 0     | 1     | 0     | 1     |
| $M_4$ | 1     | 0     | 1     | 2     | 0     | 0     | 0     |
| $M_5$ | 0     | 1     | 1     | 2     | 0     | 0     | 0     |
| $M_6$ | 1     | 0     | 1     | 1     | 0     | 1     | 0     |
| $M_7$ | 0     | 0     | 0     | 1     | 1     | 0     | 0     |
| $M_8$ | 2     | 0     | 0     | 0     | 1     | 1     | 0     |
| $M_9$ | 2     | 0     | 0     | 0     | 0     | 1     | 0     |

Thus, in this article, we place an emphasis on how to steer clear of RG to find the state space of an LS\textsuperscript{3}PR.

**Theorem 1.** The number of combinations of placing $m$ elements into $n$ distinguishable containers is $C(n + m - 1, m) = (n + m - 1)!/[m!(n - 1)!]$.

**Corollary 1.** Let $(N, M_0)$ be an LS\textsuperscript{3}PR, $I_r$ be a minimal $P_r$-semiflow with $\| I_r \| = \{ r \} \cup H(r)$, and $(N|_r, M_0|_r)$ be a marked subnet of $(N, M_0)$ supported by $I_r$. Supposing that there are $m$ places in $(N|_r, M_0|_r)$, the number of IMs of $(N|_r, M_0|_r)$ is $|I_r(N|_r, M_0|_r)| = C(m + 1 - 1, 1) = m$.

By Corollary 1, we find the number of IMs of $(N|_r, M_0|_r)$ and each IM in $I_r(N|_r, M_0|_r)$ can be generated by combinatorics. In an LS\textsuperscript{3}PR, $\| I_r \| = \{ r \} \cup H(r)$ and $M(\| I_r \|) = 1$. The token in $\| I_r \|$ can flow freely among the places in $\| I_r \|$ if we do not consider deadlocks.

$I_{r_1} = [0, 1, 0, 0, 1, 0, 1, 0]^T$ and $I_{r_2} = [1, 1, 1]^T$. The net $(N|_{r_1}, M_0|_{r_1})$ shown in Figure 2, supported by $I_{r_1}$, is a marked subnet of the LS\textsuperscript{3}PR in Figure 1. $I_{r_1} = p_2 + p_6 + p_7$ and $M_0|_{r_1} = p_7$. $p_7$ has two holders $p_2$ and $p_6$. The token in $p_7$ can flow freely among $p_2$, $p_6$, and $p_7$. By Corollary 1, we find $|I_{r_1}(N|_{r_1}, M_0|_{r_1})| = 3$, and the specific markings are generated and shown in Table 3.

**Table 3.** Markings in $l_{r_1}(N|_{r_1}, M_0|_{r_1})$.

| $\| I_{r_1} \|$ | $p_1$ | $p_2$ | $p_3$ |
|----------------|-------|-------|-------|
| $M_1$          | 1     | 0     | 0     |
| $M_2$          | 0     | 1     | 0     |
| $M_3$          | 0     | 0     | 1     |

**Definition 5.** Let $\| I_{r_1} \|$, $\| I_{r_2} \|$, ..., and $\| I_{r_8} \|$ be supports of minimal $P_r$-semiflows in $(N, M_0)$ and $M_0|_{r_1}$, $M_0|_{r_2}$, ..., and $M_0|_{r_8}$ be the initial states of $\| I_{r_1} \|$, $\| I_{r_2} \|$, ..., and $\| I_{r_8} \|$, respectively. A set $G = \bigcup_{i=1}^{m} I_{r_i}$ is called an $m$-configuration, where $m = |P_R \cap G|$ and $m \leq n$. $M_0|_{r_1} = \sum_{i=1}^{m} M_0|_{r_i}$ is the initial state of $G$. 

![Figure 1. A simple LS\textsuperscript{3}PR.](image1)

![Figure 2. A subnet of an LS\textsuperscript{3}PR.](image2)
Table 4. States in \( R(N^{[6]}_6, M^G_{06}) \) and \( R(N^{[6]}_6, M^G_{06}) \).

| \( l_0 \) | \( P_1 \) | \( P_2 \) | \( P_3 \) | \( l_8 \) | \( P_5 \) | \( P_6 \) | \( P_7 \) | \( P_8 \) |
|---|---|---|---|---|---|---|---|---|
| \( M_1 \) | 1 | 0 | 0 | \( M_1 \) | 1 | 0 | 0 |
| \( M_2 \) | 0 | 1 | 0 | \( M_2 \) | 0 | 1 | 0 |
| \( M_3 \) | 0 | 0 | 1 | \( M_3 \) | 0 | 0 | 1 |

Table 5. Reachable states of theory permissive of \( (N^G, M^G_9) \).

| \( G \) | \( P_1 \) | \( P_2 \) | \( P_3 \) | \( P_4 \) | \( P_5 \) | \( P_6 \) | \( P_7 \) | \( P_8 \) |
|---|---|---|---|---|---|---|---|---|
| \( M_1 \) | 0 | 1 | 1 | 2 | 0 | 0 | 0 | 0 |
| \( M_2 \) | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| \( M_3 \) | 1 | 1 | 0 | 2 | 0 | 0 | 0 | 1 |
| \( M_4 \) | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| \( M_5 \) | 2 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| \( M_6 \) | 2 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| \( M_7 \) | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| \( M_8 \) | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

Table 6. Reachable states of theory permissive of \( (N, M_6) \) in Figure 1.

| \( N \) | \( P_1 \) | \( P_2 \) | \( P_3 \) | \( P_4 \) | \( P_5 \) | \( P_6 \) | \( P_7 \) |
|---|---|---|---|---|---|---|---|
| \( M_1 \) | 0 | 1 | 1 | 2 | 0 | 0 | 0 |
| \( M_2 \) | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| \( M_3 \) | 1 | 1 | 0 | 2 | 0 | 0 | 0 |
| \( M_4 \) | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| \( M_5 \) | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

Definition 6. Let \( G \) be an \( m \)-configuration of an \( LS^3 \)PR and \( G = \{ l_i_1 \} \cup \{ l_i_2 \} \cup \cdots \cup \{ l_i_n \} \). Supposing that there exists an SMS \( S \) that satisfies \( G = S \cup [S] \), then \( G \) is denoted an \( m \)-configuration. Sometimes, \( m \)-configuration is denoted as \( \ast \)-configuration for short.

Theorem 2. Let \( G = \{ l_i_1 \} \cup \{ l_i_2 \} \cup \cdots \cup \{ l_i_n \} \) be an \( m \)-configuration of an \( LS^3 \)PR \( (N, M_6) \). Suppose that \( p_1 \in [l_i_1] \cap [P_A], p_2 \in [l_i_2] \cap [P_A], \ldots, p_m \in [l_i_m] \cap [P_B] \). If \( M^G = p_1 + p_2 + \cdots + p_m \) is an unreachable state of \( (N^G, M^G_9) \) if \( \forall p \in \{ p_1, p_2, \ldots, p_m \}, \forall p_j \in \ast, p \cap P_A, p_j \in G \).

Proof. We prove it by contradiction. If \( M^G = p_1 + p_2 + \cdots + p_m \) is reachable, let \( M^G = \{ M | M = \psi u, p_i + p_i + \cdots + p_i + r_i, p_i \in \{ p_1, p_2, \ldots, p_m \} \} \cdot 0 \in \{ p_1, p_2, \ldots, p_m \} \cdot \geq 1 \} \) be the set of closely previous states of \( M^G \) in \( R(N^G, M^G_9) \), where \( p_j, p \in \ast, p \cap P_A \). Supposing that \( p_j \in [l_i_j] \cap [P_A] \), it is found that, \( M \in M^G \), there exists \( p \) such that \( p \neq p_j, p \in [l_i_j] \cap [P_A] \cdot M(p) = 1 \). At this moment, there are two tokens in \( [l_i_j] \cap [P_A] \) since \( M(p) = 1 \) and \( M(p_j) = 1 \). It contradicts with the fact that \( l_i_j \) is a \( P \)-semiflow. Hence, \( M^G \) is an unreachable state of \( (N^G, M^G_9) \).

Definition 7. Let \( G \) be an \( m \)-configuration of an \( LS^3 \)PR \( (N, M_6) \) and \( M^G = p_1 + p_2 + \cdots + p_m \) be an

Unreachable states exclusion

Without deadlocks, the tokens in the \( P \)-semiflows of an \( LS^3 \)PR can flow freely among the places in the support of the \( P \)-semiflow. The existence of deadlocks makes part of states unreachable. It is proved in the work by Ezepeleta et al.\(^7\) that an SMS means at least a deadlock. Hence, we concentrate on an SMS-based method to determine the unreachable states from the upper bound.

Let \( R_0(N, M_6) \) and \( R(N, M_6) \) denote the set of reachable states of theory permissive and the set of reachable states of an \( LS^3 \)PR \( (N, M_6) \), respectively. Here, \( R_0(N, M_6) = \{ M | M \in R(N, M_6) \} \) is used to denote the set of unreachable states.
unreachable state of \((N^G, M^G)\). \(M^G\) is called a minimal unreachable state (MUS) of \((N, M^G)\) if \(\exists p_1 \in \{p_1, p_2, \ldots, p_m\}, M = M^G - p_1\) is not an unreachable state of \((N^G, M^G)\), where \(G' = \{p | p \in G, p \not\in \{I_r\}\}\) and \(I_r\) denotes the minimal \(P_r\)-semiflow where \(p_1\) stays.

**Property 4.** Let \(G = || I_r || \cup || I_1 || \cup \ldots \cup || I_n || \cup \ldots \cup || I_m ||\) be an \(m\)-configuration of an LS\(^3\)PR \((N, M_0)\). \(G\) is at an unreachable state if \(G' = \{p | p \in G, p \not\in \{I_r\}\}\) is at an unreachable state.

The net \(N_0\) shown in Figure 3(a) is a subnet of the LS\(^3\)PR in Figure 1. \(||I_r||\) and \(||I_r||\) make up a two-configuration, denoted as \(G = ||I_r|| \cup ||I_p|| = \{p_2, p_3, p_5, p_6, p_7, p_8\}\). There exists an SMS \(S = \{p_3, p_6, p_7, p_8\}\) in \(G\). \(M^G_0 = p_3 + p_6\) is the initial state of \(N_0\). By Theorem 2, it is concluded that \(M^G_u = p_3 + p_6\) is an unreachable state of \((N^G, M^G_0)\).

As known to all, the unreachable states of an LS\(^3\)PR are generated due to the existence of deadlocks. Precisely, deadlocks render part of states unreachable. By Property 4, we can find that the unreachable states of an LS\(^3\)PR \((N, M_0)\) can be derived from the MUSs of the \(*\)-configurations in \((N, M_0)\). If we can find all the MUSs, we can obtain all the unreachable states of \((N, M_0)\) correspondingly.

**Lemma 1.** Let \(G = || I_1 || \cup || I_2 || \cup \ldots \cup || I_n || \cup \ldots \cup || I_m ||\) be an \(m\)-configuration of an LS\(^3\)PR \((N, M_0)\). Suppose that \(p_1 \in || I_r || \cup || P_A ||, p_2 \in || I_r || \cup || P_A ||, \ldots, p_1 \in || I_2 || \cup || P_A ||, \ldots, p_m \in || I_m || \cup || P_A ||. M^G = p_1 + p_2 + \ldots + p_1 + \ldots + p_m\) is a reachable state of \((N^G, M^G)\) if \(M^G - p_1\) is a reachable state of \((N^G, M^G)\) and \(\exists p \in || P_A || \cup || P_A ||, p \not\in G\), where \(G' = \{p' | p' \in G, p' \not\in || I_r ||\}\) and \(I_r\) denotes the minimal \(P_r\)-semiflow where \(p_1\) stays.

**Proof.** An \(m\)-configuration, as an independent part of \(N\), and its reachable states are independent of that of other parts. If \(M^G - p_1\) is a reachable state of \((N^G, M^G)\), we just need to check whether \(p_1\) can be marked. If \(p_1\) can be marked, then we can affirm that \(M^G\) is reachable. We find there exists \(p \in R^G (p) = M(p)\) implying that whether \(p\) can be marked does not depend on \(G\). Hence, \(p_1\) can be marked and \(M^G\) is a reachable state of \((N^G, M^G)\).

**Theorem 3.** Algorithm 1 completely enumerates the MUSs of the \(*\)-configurations in an LS\(^3\)PR \((N, M_0)\).

**Proof.** Without considering deadlocks, we can find a set of all the reachable states of theory permissive, denoted as \(R_G\). Actually, \(R_G\) contains two parts: set of reachable states \(R_R\) and set of unreachable states \(R_U\). By the theory of resource circuit, operation places in \(G\) can be divided into two parts: \(P_L = \{p | p \in G \cap P_A, \forall p \in \llbracket P_k \cap P_A, p' \in G \rrbracket\}\) and \(P_U = \{p | p \in G - P_A, \exists p' \in \llbracket P_k \cap P_A, p' \not\in G \rrbracket\}\). For \(\forall m \in R_G, M = p_1 + p_2 + \ldots + p_1 + \ldots + p_m\) and \(K = \{p_1, p_2, \ldots, p_m\}\). Supposing that \(\forall p \in K, p \in P_L\), by Theorem 2, \(M\) is confirmed to be unreachable. Suppose that there is one place belonging to \(P_O\) and the other places \(P_D\) is the set of the other places) belonging to \(P_L\) in \(K\). If \(M' = \sum_{p \in P_L} p\) is an unreachable state, by Property 4, \(M\) is confirmed to be unreachable; else by Lemma 1, \(M\) is confirmed to be reachable. Suppose that there are two place belonging to \(P_O\) and the other places \(P_D\) is the set of the other places) belonging to \(P_L\) in \(K\). If \(M' = \sum_{p \in P_L} p\) is an unreachable state, by Property 4, \(M\) is confirmed to be unreachable; else by Lemma 1, \(M\) is confirmed to be reachable. By parity of reasoning, we can classify all the states in \(R_G\) into \(R_R\) and \(R_U\). Thus, it is noticed that the state in \(R_G\) either satisfies the condition in Theorem 2 and Property 4 or Lemma 1. Furthermore, Property 4 is based on Theorem 2. Hence, in this sense, we can find all the unreachable state of \((N^G, M^G)\) by Theorem 2. However, by Property 4, it is found that not all the unreachable states of a \(*\)-configuration \(G\) obtained by Theorem 2 are a minimal one of \((N^G, M^G)\). Algorithm 1 reserves the unreachable states that are minimal. Hence, Algorithm 1 completely enumerates the MUSs of the \(*\)-configurations in \((N, M_0)\).

**Definition 8.** Let \(G_1\) be a \(*\)-configuration of an LS\(^3\)PR \((N, M_0)\), \(M^G_{u_1}\) be one of MUSs of \((G_j, M^G_{u_1})\), and \(U_{\text{min}}\) be the set of all the MUSs of the \(*\)-configurations in \((N, M_0)\). The unreachable state in \(R_u(N, M_0)\) satisfies \(\forall p \in G, M^G_{u_1}(p) = M(p)\) is called the unreachable state of \((N, M_0)\) derived from \(M^G_{u_1}\). The set of the unreachable states is denoted as \(R^G_{\text{m}}(N, M_0)\). \(R^G_{\text{m}}(N, M_0)\) denotes the set of all the unreachable states of \((N, M_0)\). As previously mentioned, \(R_G(N, M_0)\) is the set
Algorithm 1. Computing the set of MUS of the $^\ast$-configurations in $(N, M_0)$.

**Input:** An LS $^3$PR $(N, M_0)$ and II. \(\forall \Pi \) denotes the set of all SMSs $^\ast$.

**Output:** $U_{\text{min}}$. \(\forall U_{\text{min}} \) denotes the set of all MUSs $^\ast$.

- $U_{\text{min}} := \emptyset$, $i := 2$.
- while $\Pi \neq \emptyset$
  - find the set of all SMSs with $i$ resources from II, denoted as $S_r$.
  - foreach $S \in S_r$
    - $S_r := S_r \setminus \Pi$.
    - II := $\Pi \setminus S$.
    - construct an $i$-configuration $G = \bigcup_{f=1}^{i} \| f \|$, II.
  - find the set of all unreachable states of $(N^G, M_0^G)$ by the method in Theorem 2, denoted as $U^G$.
  - foreach $M^G_i \in U^G$
    - if $\emptyset \subseteq \exists M \in U_{\text{min}}$ such that $\forall p \in M$
      - $M(p) = M^G_i(p)$.
      - $U_{\text{min}} := U_{\text{min}} \cup \{ M^G_i \}$.
    - end
  - end
  - $i := i + 1$.
- end

output $U_{\text{min}}$.

of reachable states of $(N, M_0)$ obtained without considering deadlocks. Hence, we just need to exclude the states in $R_U(N, M_0)$ from $R_B(N, M_0)$ and the left states are the desired ones. That is to say, $R(N, M_0) = \{ M | M \in R_B(N, M_0) \land M \notin R_U(N, M_0) \}$ is the set of the reachable states of $(N, M_0)$.

**State estimate algorithm and example**

We shape the detailed approach of calculating the state space of an LS$^3$PR $(N, M_0)$ to an algorithm and support it with an example in this section. First, a set of all the reachable states of theory permissive is generated by combinatorics. Then, we exclude the unreachable states by finding all the MUSs of the $^\ast$-configurations in $(N, M_0)$ if there exist deadlocks. Finally, the left states are the reachable states of $(N, M_0)$.

Here, the LS$^3$PR $(N, M_0)$ in Figure 4, quoted from the literature, is taken as an example. According to the combinatorics and Property 2, we can find the set of reachable states of theory permissive $R_B(N, M_0)$. The states are listed in Table 7 (for economy of space, only a part of states is listed).

There are four SMSs, II = \{ $S_1, S_2, S_3, S_4$ \}, shown as follows

- $S_1$ = \{ $p_4, p_{11}, p_{14}, p_{15}$ \}
- $S_2$ = \{ $p_4, p_{12}, p_{13}, p_{14}, p_{15}$ \}
- $S_3$ = \{ $p_5, p_{11}, p_{14}, p_{15}, p_{16}$ \}
- $S_4$ = \{ $p_5, p_{12}, p_{13}, p_{14}, p_{15}, p_{16}$ \}

Algorithm 2. Computing the reachability set of an LS$^3$PR.

**Input:** An LS$^3$PR $(N, M_0)$.

**Output:** $R(N, M_0)$

- find $R_d(N, M_0)$ by Property 2.
- compute the set of all the SMSs in N, denoted as II.
- if $II = \emptyset$
  - $R(N, M_0) := R_B(N, M_0)$.
- else
  - apply Algorithm 1 to II to find $U_{\text{min}}$.
  - $\forall U_{\text{min}}$ denotes the set of all the MUSs of the $^\ast$-configurations in $(N, M_0)$.
  - foreach $M^G_i(N, M_0) \in U_{\text{min}}$
    - compute $R^G_i(N, M_0)$.
    - $R(U(N, M_0)) := R(U(N, M_0)) \cup R^G_i(N, M_0)$.
  - end
- $R(N, M_0) := R_B(N, M_0)$.
- end

Based on the SMSs, four $^\ast$-configurations are constructed by Definition 6 and shown as follows

- $G_1 = \{ p_3, p_4, p_8, p_9, p_{10}, p_{11}, p_{14}, p_{15} \}$
- $G_2 = \{ p_2, p_3, p_4, p_7, p_9, p_{10}, p_{11}, p_{12}, p_{13}, p_{14}, p_{15} \}$
- $G_3 = \{ p_5, p_3, p_4, p_7, p_8, p_9, p_{10}, p_{11}, p_{14}, p_{15}, p_{16} \}$
- $G_4 = \{ p_2, p_3, p_4, p_5, p_7, p_8, p_9, p_{10}, p_{11}, p_{12}, p_{13}, p_{14}, p_{15}, p_{16} \}$

By Theorem 2, we can find unreachable states of each $^\ast$-configuration. However, some of these states may be contained in the other $^\ast$-configurations that have less resources. Hence, it is necessary to find MUSs. By Definition 7, it is found that $M^G_{p_4} = p_4 + p_{10}$, $M^G_{p_3} = p_3 + p_{10}$, and $M^G_{p_1} = p_1 + p_{11}$ are the MUSs of $M^G_{^\ast}$ since $M = p_1$, $i \in \{4, 10, 11\}$ are not the MUSs of $M^G_{^\ast}$. There is no MUS in $N_{^\ast}$ since the unreachable states of $N_{^\ast}$ by Theorem 2 are not minimal. That is to say, the unreachable states derived from $M^G_{u_1}$ are included in the other unreachable states derived from $M^G_{u_2}$, $M^G_{u_3}$, or $M^G_{u_4}$. For example, it is found that $M^G_{u_1} = p_3 + p_4 + p_5 + p_{12}$ is an unreachable state but not MUS of $N_{^\ast}$ since $M^G_{u_1} = p_1 + p_4 + p_{12}$ is an MUS. By applying Algorithm 1 to II, $U_{\text{min}} = \{ M^G_{u_1}, M^G_{u_2}, M^G_{u_3}, M^G_{u_4}, M^G_{^\ast} \}$ is obtained and shown in Table 8.

In the following, we need to compute the unreachable states of $(N, M_0)$ derived from the MUS in $U_{\text{min}}$. Taking $M^G_{u_1}$ as an example, we denote the unreachable states of $(N, M_0)$ derived from $M^G_{u_1}$ as $R^G_{u_1}(N, M_0)$. The elements of $R^G_{u_1}(N, M_0)$ are generated by Definition 8 and listed in Table 9.

The rest unreachable states can be deduced by analogy. Finally, we find $R_U(N, M_0) = \bigcup_{i=1}^{3} \bigcup_{f=1}^{3} \bigcup_{j=1}^{3} R^G_{u_j}(N, M_0)$.
of \((N, M_0)\), we just need to exclude the elements of \(R_U(N, M_0)\) from \(R_b(N, M_0)\). The reachable states of \((N, M_0)\) are finally listed in Table 10.

### Computational complexity analysis and comparison

In this section, we compare the computational complexity of computing the state space of an LS³PR by the proposed method with that by RG. First, if RG is applied to find reachable states, supposing that \(k\) and \(t\) denote the number of reachable states and the number of transitions, respectively, for any state \(M\) in reachability set, one needs to compute subsequent states, which requires \(t\) steps. Thus, its computational amount is \(O(t)\). Overall, considering all \(k\) reachable states in reachability set, the computational complexity of using RG to find reachable states is \(O(tk)\).

Second, if we use the proposed method to compute reachable states, it is found that its computational complexity contains three parts. Supposing that there are \(m\) operation places and \(n\) resource places, computing the reachable states of theory permissive needs at most \((m = n)\) steps. It costs \(2^n\) computational amount when using the policy in the work by Wang et al.\(^{19}\) to find all SMSs and computing the unreachable states derived from each SMS requiring at most \((m/n)\) steps. Hence, in this part, the computational complexity is \(O(2^n \cdot (m/n))\). And the third part, we need at most \((m/n)!\) steps to exclude all unreachable states from the set of reachable states of theory permissive. Consequently, putting them together, the computational complexity of using the proposed method is \(O((m/n)^n + 2^n \cdot (m/n - 2)^n + (m/n)!)\). Clearly, it greatly reduces the computational complexity although it is still an exponential complexity method.

---

**Table 7.** States in \(R_b(N, M_0)\).

| \(p_1\) | \(p_2\) | \(p_3\) | \(p_4\) | \(p_5\) | \(p_6\) | \(p_7\) | \(p_8\) | \(p_9\) | \(p_{10}\) | \(p_{11}\) | \(p_{12}\) | \(p_{13}\) | \(p_{14}\) | \(p_{15}\) | \(p_{16}\) |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| \(M_1\) | 4 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| \(M_2\) | 3 | 1 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| \(M_3\) | 4 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |
| \(M_4\) | 3 | 0 | 1 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| \(\vdots\) | \(\vdots\) | \(\vdots\) | \(\vdots\) | \(\vdots\) | \(\vdots\) | \(\vdots\) | \(\vdots\) | \(\vdots\) | \(\vdots\) | \(\vdots\) | \(\vdots\) | \(\vdots\) | \(\vdots\) | \(\vdots\) | \(\vdots\) |
| \(M_{144}\) | 4 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |

**Table 8.** MUS of *configurations.

| \(G_i\) | MUS |
|-------|-----|
| \(G_1\) | \(M_{G_1}^{u_1} = p_4 + p_{10}; M_{G_1}^{u_2} = p_9 + p_{10}; M_{G_1}^{u_3} = p_{10} + p_{11}\) |
| \(G_2\) | \(M_{G_2}^{u_1} = p_3 + p_4 + p_{12}; M_{G_2}^{u_2} = p_3 + p_9 + p_{12}; M_{G_2}^{u_3} = p_3 + p_11 + p_{12}\) |
| \(G_3\) | \(M_{G_3}^{u_1} = p_4 + p_5 + p_{8}; M_{G_3}^{u_2} = p_4 + p_8 + p_9; M_{G_3}^{u_3} = p_5 + p_8 + p_{11}\) |
| \(G_4\) | \(\emptyset\) |

MUS: minimal unreachable state.
Conclusion

This study presents a theory instead of RG to calculate the state space of an LS³PR. In deadlock prevention problems, we are committed to synthesizing optimal supervisors for plant nets. Hence, it is necessary to find all reachable states to synthesize an optimal or nearly optimal supervisor. The existing methods, such as RG, are difficult, even impossible, to find state space for a large net system due to its high computational complexity. This article pioneers the very first study to deal with a subclass of S³PR (LS³PR) by combinatorics. Excluding all unreachable states from the set of reachable states of theory permissive calculated by combinatorics, the left states are the reachable states of the considered LS³PR. Analysis shows its computational complexity is far lower than RG.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported in part by the National Natural Science Foundation of China under Grant No. 16JK1342, the National Natural Science Foundation of China under Grant No. 51607133, the Natural Science Basic Research Plan in Shaanxi Province of China under Grant No. 2016JQ5106, the Science and Technology Project of Shaanxi Province under Grant No. 2016GY-136, the Doctoral Research Startup Funds of Xi’an Polytechnic University under Grant No. BS1335, and the Innovation and Entrepreneurship Training Program of Xi’an Polytechnic University under Grant No. 2016125.

References

1. Hou YF, Zhao M and Liu D. Deadlock control for a class of generalized Petri nets based on proper resource allocation. Asia J Control 2016; 18: 206–223.
2. Li ZW. Deadlock analysis and control in resource allocation systems. Inform Sciences 2016; 363: 174–177.
3. Ghaffari A, Rezg N and Xie XL. Design of a live and maximally permissive Petri net controller using the theory of regions. IEEE T Robotic Autom 2003; 19: 137–142.
4. Li ZW, Zhou MC and Jeng MD. A maximally permissive deadlock prevention policy for FMS based on Petri net siphon control and the theory of regions. IEEE T Autom Sci Eng 2008; 5: 182–188.
5. Uzam M and Zhou MC. An improved iterative synthesis approach for liveness enforcing supervisors of flexible manufacturing systems. Int J Prod Res 2006; 44: 1987–2030.
6. Uzam M and Zhou MC. An iterative synthesis approach to Petri net-based deadlock prevention policy for flexible manufacturing systems. IEEE T Syst Man Cy A 2007; 37: 362–371.
7. Ezpeleta J, Colom JM and Martinez J. A Petri net based deadlock prevention policy for flexible manufacturing systems. *IEEE T Robotic Autom* 1995; 11: 173–184.
8. Esparza J. Reachability in live and safe free choice Petri nets is NP-complete. *Theor Comput Sci* 1998; 198: 211–224.
9. Ferrarini L. On the reachability and reversibility problems in a class of Petri nets. *IEEE T Syst Man Cyb* 1994; 24: 1474–1482.
10. Kostin AE. Reachability analysis in T-invariant-less Petri nets. *IEEE T Automat Contr* 2003; 48: 1019–1024.
11. Lipton RJ. *The reachability problem requires exponential space*. New Haven, CT: Yale University, 1976, p.62.
12. Starke PH. *INA: integrated net analyzer*. Berlin: Handbuch, 1992.
13. Chen YF, Li ZW, Khalgui M, et al. Design of a maximally permissive liveness-enforcing Petri net supervisor for flexible manufacturing systems. *IEEE T Autom Sci Eng* 2011; 8: 374–393.
14. Hong L and Chao DY. Enumeration of reachable states for arbitrary marked graphs. *IET Control Theory A* 2012; 6: 1536–1543.
15. Hong L, Wang AR, Jing JF, et al. Combinatorics and resource circuit based enumeration of reachable states for S^PR. *Adv Mech Eng* 2016; 8: 1–11.
16. Hong L, Jing JF, Hou YF, et al. Combinatorics-based estimation of reachable states for the system of simple sequential process with resources. *IEEE Trans Electron Inf Syst* 2016; 11: 134–141.
17. Roberts FS and Tesman B. *Applied combinatorics*. Oxford: Taylor & Francis, 2009.
18. Li ZW and Zhou MC. *Modeling, analysis, and deadlock control of automated manufacturing systems*. Beijing, China: Science Press, 2009 (in Chinese).
19. Wang AR, Li ZW, Jia JY, et al. An effective algorithm to find elementary siphons in a class of Petri nets. *IEEE T Syst Man Cy A* 2009; 39: 912–923.