A Fast Method for Extracting all Minimal Siphons from Maximal Unmarked Siphons of a Petri Net

Qiaoli Zhuang
School of Information Science and Technology, Zhejiang Sci-Tech University, Hangzhou 310018, China
Email: chocolatezql@gmail.com
Wenzhan Dai
School of Information and Electronic Engineering, Zhejiang Gongshang University, Hangzhou 310018, China
Email: dwz@zjgsu.edu.cn

***Abstract*— In this paper, a fast method named algorithm 2 is proposed to extract all minimal siphons from maximal unmarked siphons obtained by the MIP-based deadlock detection method. Redundant computation is the major disadvantage of an existing method named algorithm 1 and it greatly decreases the computational efficiency of minimal siphons. In order to resolve this problem, the proposed method improves from three aspects. Firstly, no sink places and transitions exist in the subnet of the tree. Secondly, no equal non-null node exists in the tree. Thirdly, if the removal of one place from a subnet node leads to the removal of all places in this node, the same place of its son node is unnecessary to compute repeatedly. The applications of algorithm 2 are illustrated with FMS examples in the following sections and comparison of algorithm 1 with algorithm 2 is also presented. At the end, the result from experiment shows that the proposed method has higher efficiency.**

***Index Terms*— deadlock,flexible manufacturing system (FMS),Petri net**

I. Introduction

Flexible manufacturing system (FMS) [1], [2] is characterized with a high degree of resource sharing and concurrency. When various types of raw parts enter the system to compete limited resources such as robots and machines, deadlock may occur if there is no effective scheduling and control mechanism. Deadlock can not only lead to the stoppage of part of system or even the entire system, but also could have catastrophic consequences in highly automated systems such as semiconductor manufacturing and safety-critical distributed databases. Therefore, potential deadlocks must be carefully considered and effective control policy has to be made to ensure that deadlocks will never occur in FMS.

Compared with other formal tools, Petri nets are characterized of graphical presentation, solid theoretical foundation in mathematics and various analysis methods [3]–[5]. On base of the above three advantages, Petri nets have been widely used to model, analyze and control discrete event system (DES). FMS is one of the typical classes of DES. Thus, over the past ten years, Petri nets have become the most important tools to deal with deadlock problems in FMS [6]–[8].

Manuscript received April 6, 2013; accepted August 8, 2013

In a Petri net formalism, deadlocks are closely tied to a well-known structural object named siphon. A siphon is a subset of places such that every input transition is also an output transition of the subset. Once a siphon loses all of its tokens, it will be unmarked permanently. That is, if a siphon becomes unmarked at a reachable marking, some output transitions of it will be disabled permanently. In recent years, many deadlock control policies are proposed based on siphons [9]–[11]. However, as we all know, the complete siphon enumeration in a Petri net is normally NP-complete, which makes many approaches based on siphons suffer from the computational complexity.

In order to solve this problem, a fast deadlock detection approach is proposed by Chu and Xie [12], it is based on mixed integer programming (MIP) for structurally bounded nets whose deadlocks are tied to unmarked siphons. With this method, a maximal unmarked siphon can be found at a given marking. Since no explicit enumeration of siphons is required, the computational efficiency of siphon-based deadlock prevention policy is greatly improved. This makes it play an important role in the development of deadlock prevention policies based on siphons.

MIP method is firstly used to design a liveness-enforcing supervisor by Huang et al [11]. In this method, an algorithm is developed to derive a minimal siphon from a maximal unmarked siphon which is obtained by the MIP-based deadlock detection method. However, this algorithm is incorrect on details. First, it fails to indicate which places are included in the minimal siphon derived from a maximal unmarked siphon. Second, the algorithm falls into an endless loop. Furthermore, it is proved that a siphon derived from maximal unmark siphon by this method is not necessarily a minimal one. In spite of this, it provides an efficient way to deal with deadlock control problems in large-size Petri nets models. Many deadlock prevention policies for resource allocation systems are inspired by their preliminary work. Afterwards, a minimal siphons extraction algorithm is proposed by Li and Liu [13] to correct these problems, and the software package

doi:10.4304/jsw.9.3.582-588

to extract a minimal siphon from a maximal unmarked siphon is also provided [14]. However, a drawback of their methods is that only one minimal siphon can be extracted at a time.

As a result, for a structurally bounded ordinary net, a minimal siphon can be found by solving an MIP problem [15], [16]. Similar work on minimal siphon extraction using MIP method is developed by Chao [12] and Li [17], [18].

To extract the set of all minimal siphons from the unmarked maximal siphon obtained by the MIP-based deadlock detection method, an algorithm developed by Wang et al [19] bases on a tree structure, named subnet tree, contains nodes (subnets) and edges (places). This algorithm provides a simple, direct, and convenient graphical representation of the minimal siphon extraction process. However, many nodes in the tree generated by this method are computed repeatedly and it will greatly decrease the computational efficiency of minimal siphons.

In addition, minimal siphons can also be computed by resource circuits [20], but it can only be used in a subclass of Petri nets called Systems of Simple Sequential Processes with Resources (S^3PR) [8]. Elia and Carlos [21] present a method about computation of the minimal siphons from a generating family of siphons, but it can be used in a subclass of Petri nets S^4PR called Systems of Sequential System with Shared Resources as well.

In this paper, based on the algorithm 1 [19], an algorithm 2 of extracting all minimal siphons from unmarked maximal siphon is proposed to improve the computational efficiency of minimal siphons. To this end, some measures are designed to avoid redundant computation of nodes in the subnet tree proposed by Wang et al [19].

The rest of this paper is organized as follows. Section II gives preliminary definitions of Petri nets as well as MIP-based deadlock detection method proposed by Chu and Xie [12]. Section III introduces the algorithm proposed by Wang et al [19] for extracting all minimal siphons from maximal unmarked siphon. The improved minimal siphon extraction algorithm is proposed in Section IV. Section V presents some examples to demonstrate the algorithm. Finally, conclusions are made in section VI.

II. PRELIMINARIES

A. Basic definitions

A Petri net [22] is a four-tuple $N = (P, T, F, W)$ where P and T are finite, nonempty, and disjoint sets. P is the set of places, and T is the set of transitions. $F \subseteq (P \times T) \cup (T \times P)$ is the incidence relation between P and T. The set $W : F \rightarrow \mathbb{N}^+$ is a mapping that assigns a weight to an arc in F, where $\mathbb{N}^+$={1, 2, 3,...}. Let $x \in P \cup T$ is a node of net $N = (P, T, F, W)$, ${}^\bullet x = \{y \in P \cup T | (y, x) \in F\}$ is called the preset of x, and $x^\bullet = \{y \in P \cup T | (x, y) \in F\}$ is called the postset of x.

The relative change of tokens for every place can be represented by the incidence matrix $[N]$ when a transition fires, where $[N]$ is a $|P| \times |T|$ integer matrix with $[N](p, t) = W(t, p) - W(p, t)$.

Given a Petri net $N = (P, T, F, W)$, a marking M of net N is a mapping from P to $\mathbb{N}$, where $\mathbb{N} = \{0, 1, 2, ...\}$. $M(p)$ denotes the number of tokens in place p. A place p is marked by a marking M iff $M(p) > 0$. A subset $S \subseteq P$ is marked by M iff at least one place in S is marked by M. The sum of tokens of all places in S is denoted by $M(S)$, where $M(S) = \Sigma_{p \in S} M(p)$.

Let S is a non-empty subset of P, S is a siphon iff ${}^\bullet S \subseteq S^\bullet$, S is a trap iff $S^\bullet \subseteq^\bullet S$. A siphon is minimal iff there is no other siphons contained in it as a proper subset. A minimal siphon S is said to be strict if it does not contain a marked trap.

Let $t \in T$ is a transition of Petri net N, t is a source transition iff ${}^\bullet t = \varnothing$ and t is a sink transition iff $t^\bullet = \varnothing$. Let $x \in P$ is a place of Petri net N, x is a source place iff ${}^\bullet x \subseteq \varnothing$ and x is a sink place iff $x^\bullet \subseteq \varnothing$.

B. MIP-based deadlock detection method

An MIP-based deadlock detection method is firstly proposed by Chu and Xie [12]. They point that the algorithm of finding an unmarked siphon corresponds with an MIP problem. Therefore, they introduce two indicators:

$$v_p = 1\{p \notin S\} \ and \ z_t = 1\{t \notin S^\bullet\} \tag{1}$$

where $S \in P$ is a maximal unmarked siphon of Petri net N .

Since $\forall t \in p^\bullet$, $v_p = 0 (p \in S) \Rightarrow z_t = 0 (t \in S^\bullet)$ and $\forall p \in t^\bullet$, $z_t = 1 (t \notin S^\bullet) \Rightarrow v_p = 1 (p \notin S)$, this leads to the following formulas:

$$z_t \geq \sum_{p \in {}^\bullet t} v_p - |{}^\bullet t| + 1, \forall t \in T \tag{2}$$

$$v_p \geq z_t, \forall (t, p) \in F \tag{3}$$

$$v_p, z_t \in \{0, 1\} \tag{4}$$

For a structurally bounded net, we have

$$v_p \geq M(p)/SB(p), \forall p \in P \tag{5}$$

where $SB(p) = max\{M(p) | M = M_0 + CY, M \geq 0, Y \geq 0\}$

An immediate implication of this property is that the maximal siphon S unmarked at a given marking M can be determined by the following integer programming problem and there exist empty siphons at M iff $G(M) < |P|$, where $G(M) = Minimize \sum_{p \in P} v_p$, s.t. constraints(1-5) and

$$M = M_0 + [N] \cdot Y, M \geq 0, Y \geq 0 \tag{6}$$

III. ALL MINIMAL SIPHONS EXTRACTION ALGORITHM

Definition 1 [19]: A tree is a two-tuple $\Gamma = (V, E)$ where V and E are two finite and disjoint sets. V is the set of vertices and E is the set of edges with $E \subseteq \{(u, v) | u, v \in V\}$ and each edge is assigned a label. Given a tree-structure graph $\Gamma = (V, E)$ and two nodes $u, v \in$

V, $u(v)$ is called a father (son) node of $v(u)$ if there exists an edge from u to v, i.e, $(u, v) \in E$.

Definition 2 [19]: Let $N = (P, T, F)$ be a Petri net with $P_x \subset P$ and $T_x \subset T$. $N_x = (P_x, T_x, F_x)$ is called a subnet generated by (P_x, T_x) if $F_x = F \cap [(P_x \times T_x) \cup (T_x \times P_x)]$.

Based on the definition 1 and 2, a new approach to extract the set of all minimal siphons from an unmarked maximal siphon is proposed by Wang et al [19]. The algorithm bases on a subnet tree contained nodes (subnets) and edges (places). It makes the minimal siphon extraction process a simple, direct, and convenient graphical representation. The algorithm is represented as follows, denoted algorithm 1

Algorithm 1: Extract the set of all minimal siphons from an unmarked maximal siphon based on trees.

Input: a maximal unmarked siphon P_x.

Output: the set of all minimal siphons Π derived from P_x.

1) Let $N_x = (P_x, T_x, F_x)$ denote the subnet generated by (P_x, T_x) where $T_x = {P_x}^{\bullet} \cup^{\bullet} P_x$
2) **while** there exist a sink transition or a sink place p in N_x **do**
3) $T_x := T_x \setminus \{t\}$; or $P_x := P_x \setminus \{p\}$;
4) **end while**
5) $\Pi := \varnothing$;
6) **while** there exists a source place p in N_x **do**
7) $P_x := P_x \setminus \{p\}$;
8) $\Pi := \cup\{p\}$;
9) **while** there exists a source transition t in N_x **do**
10) $T_x := T_x \setminus \{t\}$;
11) $P_x := P_x \setminus t^{\bullet}$;
12) **end while**
13) **end while**
14) Let $N_x^0 = (P_x, T_x, F_x)$ denote the subset generated by (P_x, T_x)
15) Let N_x^0 be the root node of this tree, and N_x^0 be a new node
16) $\Xi := \{N_x^0\}$; /*Ξ is a set of nodes*/
17) **while** there exist a new node in Ξ **do**
18) Let $x := N_x^0$ be an old node
19) **for** each $p \in P_x$ **do**
20) $P_x := P_x \setminus \{p\}$;
21) **while** there exist a source transition t in (P_x, T_x, F_x) **do**
22) $T_x := T_x \setminus \{t\}$;
23) $P_x := P_x \setminus t^{\bullet}$;
24) **end while**
25) Let $N_x = (P_x, T_x, F_x)$ be a new node
26) Add an arc p from x to N_x
27) **if** $P_x = \varnothing$ **then**
28) Let N_x be a null node
29) **else**
30) Let N_x be a new node
31) $\Xi := \Xi \cup N_x$;
32) **end if**
33) **end for**
34) **end while**
35) **for** each $N_x \in \Xi$ **do**
36) **if** each son node of N_x is a null node **then**
37) $\Pi := \Pi \cup P_x$;
38) **end if**
39) **end for**
40) Output:Π
41) End

It has been proved in [19] that algorithm 1 can extract the set of all minimal siphons from an unmarked maximal siphon. However, many of nodes in a subnet tree created by it are computed repeatedly, it will decrease the computational efficiency seriously.

IV. Improved Algorithm

In order to erase nodes which are computed repeatedly in the subnet tree, in this section, we propose an improved algorithm 2, which can quickly exact the set of all minimal siphons from an unmarked maximal siphon obtained by the MIP-based deadlock detection method.

Algorithm 2: Extract the set of all minimal siphons from an unmarked maximal siphon based on the subnet tree.

Input: a maximal unmarked siphon P_x.

Output: the set of all minimal siphons Π derived from P_x.

1) Let $N_x = (P_x, T_x, F_x)$ denote the subnet generated by (P_x, T_x) where $T_x = {P_x}^{\bullet} \cup^{\bullet} P_x$;
2) **while** there exist a sink transition t or a sink place p in N_x **do**
3) $T_x := T_x \setminus \{t\}$; or $P_x := P_x \setminus \{p\}$;
4) **end while**
5) $\Pi := \varnothing$;
6) **while** there exists a source place p in N_x **do**
7) $P_x := P_x \backslash \{p\}$;
8) $\Pi := \Pi \cup \{p\}$;
9) **while** there exists a source transition t in N_x **do**
10) $T_x := T_x \setminus \{t\}$;
11) $P_x := P_x \setminus t^{\bullet}$;
12) **end while**
13) **end while**
14) Let $N_x^* = (P_x^*, T_x^*, P_x^*)$ denote the subnet generated by the decomposition of the subnet N_x
15) Let Θ_x^* denote the subset P_x^* during decomposition of N_x^*, every Θ_x^* is associated with N_x^*, removal of a place $p \in \Theta_x^*$ will trigger removal of all places of subnet N_x^*
16) Let $N_x^0 = (P_x^0, T_x^0, F_x^0)$ denote the original subnet generated by (P_x, T_x)
17) Let N_x^0 be the root node of this tree, and N_x^0 be a new node
18) $\Xi := \{N_x^0\}$ /*Ξ is a set of nodes*/
19) $\Theta_x^0 = \varnothing$
20) **while** there exist a new node $N_x^* = (P_x^*, T_x^*, F_x^*)$ in Ξ **do**
21) Let $x := N_x^*$ be an old node
22) Let $\Phi := \varnothing$; /*Φ is the subset of P_x^*, which include places that have been dealt with*/
23) Let Θ_x denote the set of Θ_x^* which

associated with the father node of N_x^*
24) $P_x := P_x^*$;$T_x := T_x^*$;$F_x := F_x^*$
25) **for** each $p \in P_x$ **do**
26) **if** $p \in \Theta_x$ **then**
27) Add arc p from x to null node
28) **else**
29) $P_x := P_x \setminus \{p\}$;
30) **while** there exist a source transition t in (P_x, T_x, F_x) **do**
31) $T_x := T_x \setminus \{t\}$;
32) $P_x := P_x \setminus t^\bullet$;
33) **end while**
34) **while** there exist a sink transition t or a sink place p in N_x **do**
35) $T_x := T_x \setminus \{t\}$; or $P_x := P_x \setminus \{p\}$;
36) **end while**
37) **if** $\Phi \subseteq P_x$ **then**
38) Let $N_x = (P_x, T_x, F_x)$ be a new node
39) Add an arc p from x to N_x
40) **if** $P_x = \varnothing$ **then**
41) Let N_x be a null node;
42) $\Theta_x^* := \Theta_x^* \cup \{p\}$;
43) **else**
44) Let N_x be a new node;
45) $\Xi := \Xi \cup N_x$;
46) **end if**
47) **else**
48) Add arc p from x to null node
49) **end if**
50) **end if**
51) $\Phi := \Phi \cup \{p\}$;
52) **end for**
53) $\Phi := \varnothing$;
54) **end while**
55) **for** each $N_x \in \Xi$ **do**
56) **if** each son node of N_x is a null node **then**
57) $\Pi := \Pi \cup P_x$;
58) **end if**
59) **end for**
60) Output:Π
61) End

Definition 3: The non-null node $N_x = (P_x, T_x, F_x)$ of a subnet tree is equal to the node $N_y = (P_y, T_y, F_y)$ where $P_x = P_y$.

The algorithm 2 improved algorithm 1 [19] from three aspects. First, for each node N_x^* in the subnet tree, it should be made to ensure that there is no sink element included in N_x, it is guaranteed by adding step 35 to 37 after step 34. Second, to ensure that there is no equal non-null node generated in the subnet tree, we define a generation rule for a new node , that is the new non-null node N_x generated by removing a place $p \in P_x^*$ must include places which have been dealt with during the course of decomposition of N_x^*, N_x^* is denoted the father node of N_x . This generation rule is realized by defining the place set Φ as shown in step 23 and 52. Third, it is obvious that if removal of place $p \in P_x$ can lead to the removal of all places of N_x, then removal of place

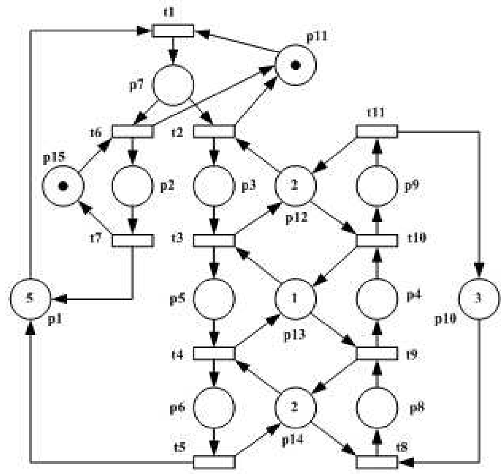

Figure 1. A S^3PR net

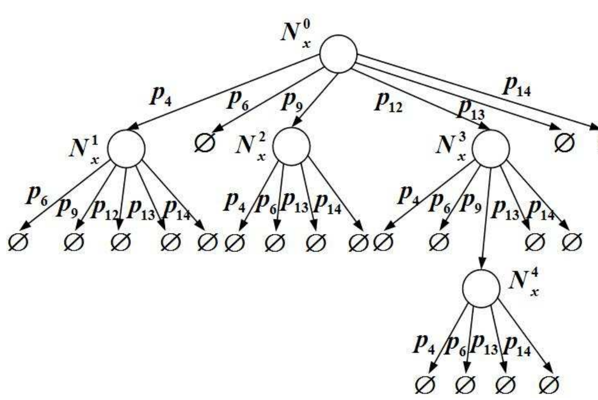

Figure 2. A subnet tree

$p \in P_x$ can also lead to the removal of all places of the subnet of N_x. Therefore, it is unnecessary to remove the place p repeatedly for son nodes of N_x. For this purpose, we add an additional place set Θ_x^* for each node of N_x^* the subnet tree to record places whose removal can lead to the removal of all places of N_x^* . It is shown from step 27 to 28 and step 43.

By improvement mentioned above, the number of non-null nodes in the subnet tree is greatly reduced. The computational efficiency of minimal siphons based on algorithm 1 is further improved.

V. Examples

In this section, we present some examples to show the application of algorithm 2.

Fig.1 and Fig.2 are the example system and the generated subnet tree shown in [19]. A decomposed maximal unmarked siphon is $P_x = \{p_4, p_6, p_9, p_{12}, p_{13}, p_{14}\}$. As shown in Fig.2, all minimal siphons which are extracted from P_x are $S_1 = \{p_6, p_9, p_{12}, p_{13}, p_{14}\}$ and $S_2 = \{p_4, p_6, p_{13}, p_{14}\}$, S_1 is derived from the node N_x^1 and S_2 is derived from N_x^2 and N_x^4. Obviously, N_x^4 is the redundant node which is equal to N_x^2.

Fig.3 is the subnet tree generated by our method. As shown in Fig. 3, redundant nodes are avoided efficiently.

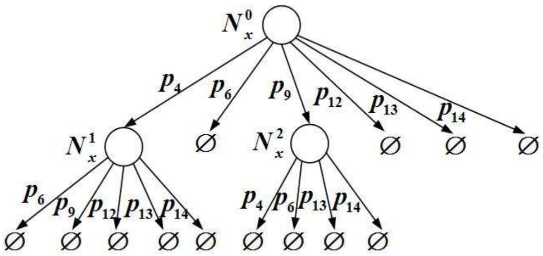

Figure 3. An improved subnet tree

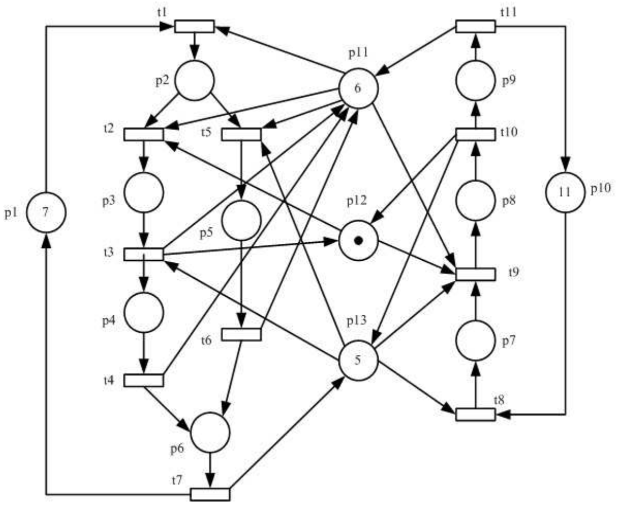

Figure 4. An ordinary petri net

To illustrate the generality and details of algorithm 2, another example is presented.

Fig.4 is an ordinary petri net. Based on the MIP-based deadlock detection method [12], a maximal unmarked siphon $P_x = \{p_3, p_4, p_5, p_6, p_8, p_9, p_{11}, p_{13}\}$ can be obtained by Lindo [23], a commercial mathematical programming software package.

According to the algorithm 2, $T_x = P_x^{\bullet} \cup {}^{\bullet}P_x = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}, t_{11}\}$. Then, after removal of sink transitions and places, the subnet $N_x^0 = \{P_x^0, T_x^0, F_x^0\}$ can be generated by (P_x, T_x), which is shown in Fig.5, where $P_x^0 = \{p_3, p_4, p_5, p_6, p_8, p_9, p_{11}, p_{13}\}$ and $T_x^0 = \{t_2, t_3, t_4, t_5, t_6, t_7, t_9, t_{10}, t_{11}\}$.

Now, the subnet tree is computed. Let N_x^0 be a new node and the root of the tree. The removal of p_3 leads to generate a new subnet node N_x^1 where exist a sink transition t_2, after removing all sink transitions and places, N_x^1 is re-presented as shown in Fig.6, and place p_3 is added to the set Φ, where $\Phi = \{p_3\}$. Because no source transition exists in Fig.6, N_x^1 become a new node and arc p_3 is added from N_x^0 to N_x^1 as shown in Fig.7. However, the removal of p_4 can lead to the removal of all places. As a result, an arc p_4 is added from N_x^0 to a null node and p_4 is added to the set Φ and Θ_x^0, where $\Phi = \{p_3, p_4\}$ and $\Theta_x^0 = \{p_4\}$.

Suppose that the same steps have been performed repeatedly on places p_5, p_6 and p_8, new arcs and new nodes are added in the tree as shown in Fig.7. As a

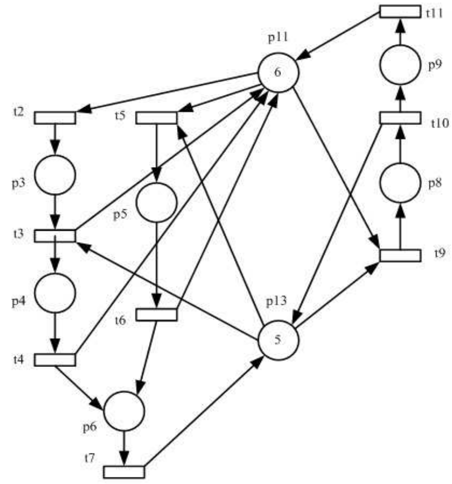

Figure 5. A subnet N_x^0

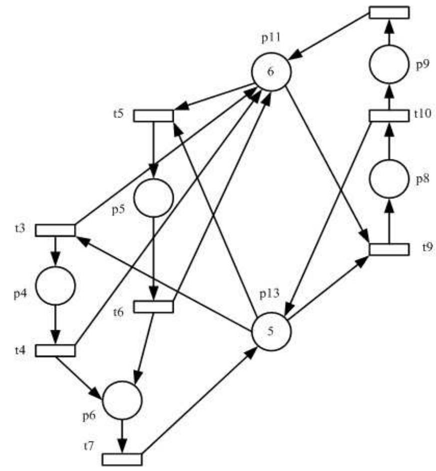

Figure 6. A subnet N_x^1

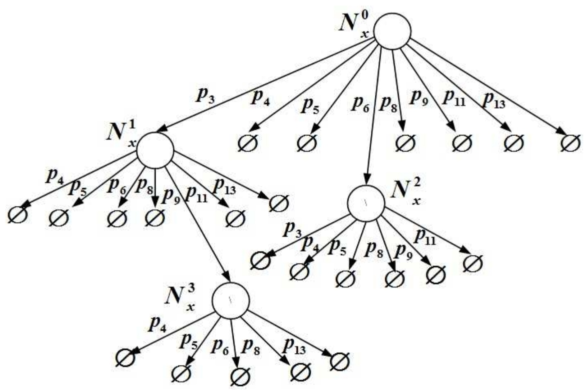

Figure 7. A generated subnet tree

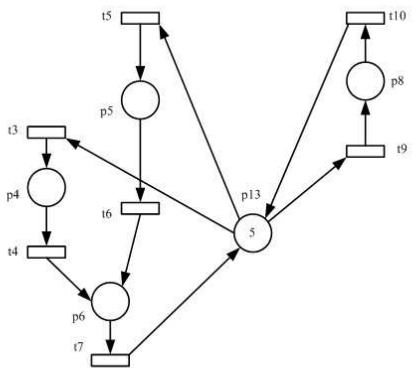

Figure 8. A subnet N_x^*

result, $\Phi = \{p_3, p_4, p_5, p_6, p_8\}$ and $\Theta_x^0 = \{p_4, p_5, p_8\}$. Now, the removal of p_9 leads to generate a new subnet node N_x^* as shown in Fig.8 and no source transition exists in N_x^*. However, places of N_x^* do not contain all places in Φ, according to the algorithm 2, an arc p_9 is added from N_x^0 to a null node and Φ is updated by $\Phi = \{p_3, p_4, p_5, p_6, p_8, p_9\}$.

Suppose that all places of N_x^0 have been dealt with, then the place set Θ_x^0 is obtained where $\Theta_x^0 = \{p_4, p_5, p_8\}$ and the place set Φ is reset to be empty. Now we remove p_4 from N_x^1. Because N_x^0 is the father node of N_x^1 and p_4 is contained in Θ_x^0, it is unnecessary to remove p_4 repeatedly, in consequence an arc p_4 is directly added from N_x^1 to a null node as shown in Fig.7, and Φ is updated by $\Phi = \{p_4\}$.

By repeating these steps, the tree can be generated as shown in Fig.7. Because each son node of N_x^2 and N_x^3 is a null node, two minimal siphons can be obtained from the maximal unmarked siphon, they are $S_1 = \{p_3, p_4, p_5, p_8, p_9, p_{13}\}$ and $S_2 = \{p_4, p_5, p_6, p_8, p_{11}\}$.

VI. Conclusions

Based on the algorithm proposed by Wang et al, this paper introduces a fast method to extract the set of all minimal siphons from unmarked maximal siphon obtained by the MIP-based deadlock detection method. Redundant computation is the major drawback of algorithm 2, it will deeply influence the computational efficiency of minimal siphons.

The major contribution of this paper is to avoid redundant computation from three aspects. Firstly, no sink places and transitions exist in the subnet of the tree. Secondly, no equal non-null node exists in the tree. Thirdly, since removal of a place from one subnet node can lead to the removal of all places in it, the same place of its son node is unnecessary to be computed repeatedly. The result from experiment has shown that this method can avoid redundant computation effectively and computational complexity of all minimal siphons is greatly simplified.

Nevertheless, all minimal siphons are still obtained from an unmarked maximal siphon based on MIP method. As we all know, MIP problem is theoretically NP-hard, and the unmarked maximal siphon obtained by the MIP-based deadlock detection method is not necessarily the real existing siphon with fake reachable markings. Therefore, the future work should focus on computing all minimal siphons directly for a given Petri net.

References

[1] C. Basnet and J. H. Mize, "Scheduling and control of flexible manufacturing systems: A critical review," *Int. J. Comput. Integr. Manuf.*, vol. 7, no. 6, pp. 340–355, 1994.

[2] S. Wang, D. Wang, Z. Tan, X. Guan, and C. Wang, "A fast convergence deadlock control strategy of petri nets based on mip algorithm," *IJACT: International Journal of Advancements in Computing Technology*, vol. 4, no. 17, pp. 589–598, 2012.

[3] C. Xia, "Structural property analysis of petri net," *Journal of Computers*, vol. 7, no. 1, pp. 292–300, 2012.

[4] X. X. Yang Xu, "Modeling and analysis of security protocols using colored petri nets," *Journal of Computers*, vol. 6, no. 1, pp. 19–27, 2011.

[5] C. Xia, "Analysis and application of petri subnet reduction," *Journal of Computers*, vol. 6, no. 8, pp. 1662–1669, 2011.

[6] M. P. Fanti and M. C. Zhou, "Deadlock control methods in automated manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A: Syst. Humans*, vol. 34, no. 1, pp. 5–22, 2004.

[7] Y. S. Huang, Y. L. Pan, and M. C. Zhou, "Computationally improved optimal deadlock control policy for flexible manufacturing systems," *IEEE Transactions on Man and Cybernetics, Part A: Systems and Humans*, vol. 42, no. 3, pp. 404–415, 2012.

[8] J. Ezpeleta, J. M. Colom, and J.Martinez, "A petri net based deadlock prevention policy for flexible manufacturing systems," *IEEE Trans. Robot. Autom.*, vol. 11, no. 2, pp. 173–184, 1995.

[9] X. Guan, Y. Li, J. Xu, C. Wang, and S. Wang, "A literature review of deadlock prevention policy based on petri nets for automated manufacturing systems," *JDCTA: International Journal of Digital Content Technology and its Applications*, vol. 6, no. 21, pp. 426–433, 2012.

[10] S. G. Wang, C. Y. Wang, and Y. P. Yu, "Comments on siphon-based deadlock prevention policy for flexible manufacturing systems," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 41, no. 2, pp. 338– 340, 2011.

[11] Y. S. Huang, M. D. Jeng, X. L. Xie, and S. L. Chung, "Deadlock prevention policy based on petri nets and siphons," in *Int. J. Prod. Res.*, 2001, pp. 283–305.

[12] F. Chu and X.L.Xie, "Deadlock analysis of petri nets using siphons and mathematical programming," *IEEE Trans. Robot. Autom.*, vol. 13, no. 6, pp. 793–804, 1997.

[13] Z. W. L.i and D. Liu, "A correct minimal siphons extraction algorithm from a maximal unmarked siphon of a petri net," *Int. J. Prod. Res.*, vol. 45, no. 9, pp. 2161–2165, 2007.

[14] D. Liu, X. Li, and Z. W, "Fanm: Extraction of a minimal siphon from a maxmial one [online]," 2010, http//www.google.com/zhiwulixidian.html.

[15] A. Giua and C. Seatzu, "Liveness enforcing supervisors for railway networks using es^2pr petri nets," in *Workshop Discrete Event System in Proc. 6th Int.*, 2002, pp. 55–60.

[16] A.Giua and C.Seatzu, "Modeling and supervisory control of railway networks using petri nets," *IEEE Trans. Autom. Sci. Eng.*, vol. 5, no. 3, pp. 431–445, 2008.

[17] S. Y. Li and Z. W. Li, "Solving siphons with the minimal cardinality in petri nets and its application to deadlock control," *Int. J. Prod. Res.*, vol. 50, no. 22, pp. 6203–6218, 2012.

[18] S. Y. Li, Z. Li, and H. S. Hu, “Siphon extraction for deadlock control in flexible manufacturing systems by using petri nets,” *Int. J. Comput. Integr. Manuf*, vol. 24, no. 8, pp. 710–725, 2011.

[19] S. Wang, M. Zhou, and C. Wang, “Extracting all minimal siphons from maximal unmarked siphons in manufacturing-oriented petri nets,” in *IEEE CASE 2011*, 2011, pp. 2681–2685.

[20] Z. W. Li and M. C. Zhou, “On siphon computation for deadlock control in a class of petri nets,” *IEEE Trans. Syst., Man, Cybern., A, Syst., Humans*, vol. 38, no. 3, pp. 667–679, 2008.

[21] C. E.E., R. C.A., and C. J., “On the computation of the minimal siphons of s^1pr nets from a generating family of siphons,” in *IEEE ETFA 2010*, 2010, pp. 1–8.

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

[23] Lindo, Premier Optimization Modeling Tools, http://www.lindo.com/.

Wenzhan Dai received the M.S. degree in electronic engineering from Nanjing University of Science and Technology, Nanjing, China, in 1985 and 1988. He is a Professor at Zhejiang Sci-Tech University and Zhejiang Gongshang University. His research in- terest covers system modeling and in- telligent control. Corresponding author of this paper.

Qiaoli Zhuang received the B.S. degree in computer science and technology from Shaanxi Normal University, Xi’an, China, in 2001 and M.S. degree in computer application technology from Hangzhou Dianzi University, Hangzhou, China, in 2007, respectively. Currently, she is a PhD candidate of mechanical design and theory in School of Machinery and Automatic control, Zhe- jiang Sci-Tech University. Her research interests include supervisory control of DES and Petri nets.