Information Sciences 409–410 (2017) 87–100

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Computation of strict minimal siphons in a class of Petri nets based on problem decomposition

Dan You[a], ShouGuang Wang[a,*], MengChu Zhou[b]

[a] School of Information & Electronic Engineering, Zhejiang Gongshang University, Hangzhou, 310018, China
[b] Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102, USA

A R T I C L E   I N F O

Article history:
Received 8 May 2016
Revised 19 April 2017
Accepted 7 May 2017
Available online 8 May 2017

Keywords:
Discrete event systems (DES)
Petri nets
Siphon computation
Deadlock
Problem decomposition
Polynomial complexity

A B S T R A C T

Efficient siphon computation plays an important role in deadlock control. This work, based on problem decomposition, develops a new method to compute all strict minimal siphons (SMS) in a class of Petri nets called Systems of Simple Sequential Processes with Resources (S^3PR). It is proved to be of polynomial complexity with respect to the number of SMSs. Therefore, it is readily applicable to an S^3PR with a large number of SMSs. Its superiority over the existing methods is validated via experimental results.

1. Introduction

Many discrete event systems (DES) such as flexible manufacturing systems, unmanned urban traffic systems, and network-communication systems in the real world face deadlock issues. In a deadlock situation, two or more competing processes are each waiting for the others to finish, and thus none can proceed. The occurrence of deadlocks often deteriorates the utilization of resources and may lead to catastrophic results. Therefore, it is critically important for both academic and industrial communities to solve the deadlock problems.

Petri nets are a modelling tool that is widely used to deal with deadlock problems in DES [3,5,6,8–10,12–14,18–21,24,26–30,35,37,41,43–46,48,49]. Siphons, as a well-known structural object of Petri nets, are closely related to deadlocks of a Petri net [6,22]. More specifically, once a siphon contains insufficient tokens, some transitions are permanently disabled, i.e., a partial or total deadlock arises in the net. Hence, many researchers study siphon-based deadlock resolution approaches [3,6,8,13,18,19,21,28–30,41,44,48,49], which guarantee that deadlocks never occur in Petri nets by controlling transition firings such that siphons are sufficiently marked.

The computational complexity of siphon-based deadlock resolution approaches mainly depends on that of siphon computation. Hence, the efficient computation of siphons is of great importance. Since the number of siphons in a Petri net in the worst case grows exponentially with respect to the net size, all siphon computation methods are theoretically of exponential complexity with respect to the net size. However, different methods differ greatly in their computational efficiency.

* Corresponding author.
E-mail addresses: youdan000@hotmail.com (D. You), wsg5000@hotmail.com (S. Wang), zhou@njit.edu (M. Zhou).

http://dx.doi.org/10.1016/j.ins.2017.05.011
0020-0255/© 2017 Published by Elsevier Inc.

Generally, siphon computation methods can be classified into two categories considering their scopes of applications. Some are applicable to general Petri nets, such as the INA-based method [31], sign matrix method [2], semi-tensor product of matrix method [11], graph theory [1], linear integer programming [10,28,29], and problem decomposition [7,25,40]. The other ones are applicable to some special classes of Petri nets only. Although application scopes are restricted, their computational efficiency is usually quite high in comparison with the former. Hence, many researchers study siphon computation for some special classes of Petri nets [4,6,8,13,16,17,19,20,27,30,33–35,38,42,46,47].

Especially, a class of Petri nets called Systems of Simple Sequential Processes with Resources (S^3PR) [8] receive much attention and many methods are proposed to compute their strict minimal siphons (SMS). Li et al. [20] propose a resource circuit-based method to compute SMS in S^3PR. They carry out an experiment to compare the computational efficiency among the INA-based method [31], sign matrix method [2], and the resource circuit-based method. Their results show that the resource circuit-based one offers much higher computational efficiency than the other two. However, some incorrect results about SMS computation in [20] are pointed out in [42]. Besides, the loop resource subset-based method proposed in [42] is shown to be superior to the resource circuit-based one. Xing et al. define the concept of maximal perfect resource-transition circuits (MPRT-circuits) [45] in S^3PR and a deadlock is characterized as a saturated MPRT-circuit. Essentially, a MPRT-circuit is a structural object equivalent to an SMS. The computation of MPRT-circuits is presented in [46,47].

Li and Zhou [19] divide SMS into elementary and dependent ones. They prove that under some conditions, a liveness-enforcing supervisor can be obtained by controlling elementary SMS only. It implies that only elementary SMS are required to be computed for supervisor synthesis. Hence, some efforts are made to improve the efficiency of computing elementary SMS [16,38].

Some studies also focus on a class of Petri nets called S^4PR [32] when computing minimal siphons. Note that S^4PR is a more general class than S^3PR and thus siphon computation methods for S^4PR are also applicable to S^3PR. Cano et al. [4] propose a pruning-graph-based method, which can enumerate all minimal siphons in S^4PR. Tricas et al. [33,34] utilize a genetic algorithm to compute a set of minimal siphons. Besides, Tricas et al. [35] implement a deadlock prevention method by means of the solutions for a set of integer linear programs, which avoids the complete enumeration of siphons in S^4PR.

Chu and Xie [6] first propose mixed integer programming (MIP), through which a maximal unmarked siphon can be found at a given marking. Similar studies on the computation of a maximal unmarked siphon are reported in [27,30]. Later, Huang et al. [13] and Li and Liu [17] propose minimal siphon extraction algorithms based on MIP such that the complete siphon enumeration is successfully avoided.

Most of existing methods can hardly handle a large-size net. For example, when INA is used to find all minimal siphons in a net with 72 places and 64 transitions, the computation in a personal computer aborts due to memory overflow after several days [21]. Besides, most of existing complete siphon enumeration methods fail to obtain a result even when they are applied to a Petri net without a huge number of siphons. One reason behind it is that they are of exponential complexity not only with respect to the net size, but also with respect to the number of siphons to be computed. Hence arises a strong need for efficient siphon computation. To the best knowledge of the authors, no siphon computation method has been established such that it has only polynomial complexity with respect to the number of siphons in such special classes of Petri nets as S^3PR.

This work studies how to efficiently compute all SMSs in S^3PR. A novel algorithm based on problem decomposition is proposed. More importantly, it is proved to be of polynomial complexity with respect to the number of SMSs to be found, which has never been seen in any prior work.

The remainder of this work is organized as follows. Section 2 briefly presents the related notions of Petri nets. Section 3 introduces a novel SMS computation method for S^3PR. Section 4 presents its algorithm complexity and experimental results that validate the superiority of the proposed method over its peers. Section 5 concludes this work.

2. Preliminaries

2.1. Petri nets [23]

An *ordinary Petri net* is a 3-tuple $N=(P, T, F)$ where P and T are finite, nonempty, and disjoint sets. P is the set of *places*, and T is the set of *transitions*. The set $F\subseteq(P\times T)\cup(T\times P)$ is the *flow relation*. Given a net $N=(P, T, F)$ and a node $x\in P\cup T$, ${}^{\bullet}x=\{y\in P\cup T\ |(y, x)\in F\}$ is the *preset* of x, while $x^{\bullet}=\{y\in P\cup T\ |(x, y)\in F\}$ is the *post-set* of x. $\forall X\subseteq P\cup T$, ${}^{\bullet}X=\cup_{x\in X}{}^{\bullet}x$, and $X^{\bullet}=\cup_{x\in X}x^{\bullet}$. $\forall x\in P\cup T$, ${}^{\bullet\bullet}x={}^{\bullet}({}^{\bullet}x)$, and $x^{\bullet\bullet}=(x^{\bullet})^{\bullet}$. $N=(P, T, F)$ is called a *state machine* if $\forall t\in T$, $|{}^{\bullet}t|=|t^{\bullet}|=1$.

A marking M of N is a mapping from P to $\mathbf{N}$ where $\mathbf{N}=\{0, 1, 2, \ldots\}$. We use the multi-set notation $\sum_{p\in P}M(p)p$ to denote vector M, where $M(p)$ indicates the number of tokens in p at M. For example, $M=[2, 1, 0, 0]^T$ is denoted by $M=2p_1+p_2$. A place p is marked by M if $M(p)>0$.

The *incidence matrix* of N is a matrix $[N]$: $P\times T\rightarrow\mathbf{Z}$ indexed by P and T such that $[N](p, t)=-1$ if $p\in{}^{\bullet}t\backslash t^{\bullet}$; $[N](p, t)=1$ if $p\in t^{\bullet}\backslash{}^{\bullet}t$; otherwise $[N](p, t)=0$ for all $p\in P$ and $t\in T$. A *P-vector* is a column vector I: $P\rightarrow\mathbf{Z}$ indexed by P, where $\mathbf{Z}$ is the set of integers. I is a *P-invariant* if $I\neq\mathbf{0}$ and $I^T\bullet[N]=\mathbf{0}^T$ hold. A P-invariant I is a *semiflow* if every element of I is non-negative. $||I||=\{p\in P\ |I(p)\neq 0\}$ is called the *support* of I.

A nonempty set $S\subseteq P$ is a *siphon* if ${}^{\bullet}S\subseteq S^{\bullet}$. A siphon is *minimal* if there is no siphon contained in it as a proper subset. A minimal siphon that does not contain the support of any P-invariant is called a *strict minimal siphon* (SMS).