Discrete Event Dyn Syst (2013) 23:3–26
DOI 10.1007/s10626-011-0124-9

An iterative method for synthesizing non-blocking supervisors for a class of generalized Petri nets using mathematical programming

Mi Zhao · Yifan Hou

Received: 3 May 2011 / Accepted: 18 November 2011 / Published online: 13 December 2011

Abstract This paper presents a novel and computational deadlock prevention policy for a class of generalized Petri nets, namely G-systems, which allows multiple resource acquisitions and flexible routings with machining, assembly and disassembly operations. In this research, a mixed integer programming (MIP)-based deadlock detection technique is used to find an insufficiently marked minimal siphon from a maximal deadly marked siphon for generalized Petri nets. In addition, two-stage control method is employed for deadlock prevention in Petri net model. Such proposed method is an iterative approach consisting of two stages. The first one is called *siphons control*, which adds a control place to the original net for each insufficiently marked minimal siphon. The objective is to prevent minimal siphons from being insufficiently marked. The second one, called *control-induced siphons control*, is to add a control place to the augmented net with its output arcs connecting to source transitions, which assures that there is no new insufficiently marked siphon generated due to the addition of the monitors. Compared with the existing approaches, the proposed deadlock prevention policy can usually lead to a non-blocking supervisor with more permissive behavior and high computational efficiency for a sizeable plant model due to avoiding complete siphon enumeration. Finally, a practical flexible manufacturing system (FMS) example is utilized to illustrate the proposed method.

Keywords Petri nets · Deadlock prevention · Mixed integer programming · *G*-systems

M. Zhao (✉)
Machinery and Electricity College, Shihezi University, Xinjiang 832003, People's Republic of China
e-mail: zhaomi530@163.com

Y. Hou
School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, People's Republic of China
e-mail: houyifan.mm@gmail.com

1 Introduction

Flexible manufacturing systems (FMS) is made up of a collection of resources such as computer numerical control machine, robots, AGV, buffers and fixtures. It usually exhibits a high degree of resource sharing in order to increase flexibility such that it can quickly respond to market changes. The competition for shared resources can cause deadlocks that are a highly undesirable situation in an FMS. Their occurrence means the stoppage of the whole or partial system operation. In an FMS, deadlocks and related blocking phenomena can often cause unnecessary costs such as long downtime and low utilization of some critical and expensive resources, and may lead to catastrophic results in some highly automated systems. One way of dealing with deadlock problems is to model an FMS with Petri nets. The major strategies using Petri net techniques to cope with deadlocks in FMS are mainly implemented by preventing some necessary condition or detect and resolve a deadlock when it occurs.

Generally, three approaches have been proposed to cope with the deadlock problems for resource allocation systems in the literature. The first one is deadlock detection and recovery (Fanti and Zhou 2004), which is an optimistic strategy that grants a resource to a request as long as the resource is available. A deadlock detection algorithm is used to detect the occurrence of deadlocks. Once a deadlock is detected, a recovery mechanism is initialized by aborting one or more processes involved in the deadlock and the resources held by the aborted processes are relinquished. A deadlock detection and recovery strategy is often used in the case where deadlocks are infrequent and resulting consequence is not serious or does not cause much damage. As for the second one, deadlock avoidance (Barkaoui and Abdallah 1994; Ezpeleta and Recalde 2004; Hsien and Chang 1994), at each system state an on-line resource allocation mechanism is used to determine the correct system evolutions among the feasible ones. Although this approach usually leads to the better use of resources and throughput, but do not totally eliminate all deadlocks. Moreover, conservative methods eliminate all unsafe states and deadlocks, and even some good states, thereby degrading the system performance. Deadlock prevention is considered to be a well-defined problem in discrete event systems. It is usually achieved by establishing a static resource allocation policy such that deadlocks never occur in the system (Zouari and Barkaoui 2003; Ezpeleta et al. 1995; Li et al. 2007a; Li and Zhou 2004a, 2006b; Liu et al. 2009a, b, 2010a; Tricas 2003; Uzam 2002; Chao 2008; Jeng et al. 2004).

More recently, monitors (control places) and related arcs are added to a plant net model to realize the off-line computational mechanism (Iordache and Antsaklis 2006; Moody and Antsaklis 2001; Wei and Li 2008; Xing et al. 2009). The monitors are used to prevent the presence of insufficiently marked siphons that are the direct cause of deadlocks in such a generalized Petri net. Unfortunately, a liveness-enforcing monitor-based supervisor derived from siphons has found its high structural complexity when the number of siphons is large. By fully utilizing the structural information in a Petri net, the work by Li and Zhou proposed the concept of elementary siphons (Li and Zhou 2004a, 2006a). In theory, the size of a supervisor designed using elementary siphons is bounded by the smaller of place count and transition count. Although the concept of elementary siphons in a generalized Petri net is developed in our previous work (Li et al. 2007b; Li and Zhao 2008), which plays an important role in the design of structurally simple liveness-enforcing net supervisors. However, it does not lower the computational complexity because it still needs complete siphon enumeration whose computation is very time-consuming when dealing with a large-size plant net model. As we known, such enumeration is expensive since the number of siphons in a net grows exponentially with respect to the net size. Hence, the shortcoming of the existing methods is that complete siphon enumeration is needed. To avoid the disadvantage, Huang et al. (2001) proposed a two-stage deadlock prevention policy for S^3PR nets and performs the synthesis of a supervisor in an iterative way. At the first stage, a maximal empty siphon is obtained by a Mixed Integer Programming (MIP) deadlock detection technique, then a monitor is added for each minimal siphon that is derived from the maximal empty one, control-induced siphons can possible be produced due to the addition of monitors. At the second stage, monitors are added to make control-induced siphons (if they exist) controlled without generating new problematic siphons.

A more generalized class of Petri nets, namely *G*-systems, was firstly proposed by Barkaoui et al. (1997), which generalizes the well-known manufacturing-oriented net models in the literature and can describe some realistic manufacturing features, including process flexibility, assembly(synchronization) and disassembly(splitting) operations, assignment flexibility, and permutation flexibility. A supervisor for G-systems aims to ensure the relevant property of non-blockingness. In this paper, we focus on the establishment of an MIP-based control policy for deadlock prevention in a *G*-system. We developed a fast deadlock detection approach based on MIP for general structurally bounded nets whose deadlocks tie to deadly marked siphons. The algorithm is an MIP-based iterative approach, at each iteration, a maximal deadly marked siphon can be computed by the MIP technique. The algorithm consists of two main stages. The first stage, called *siphons control*, adds monitors to the plant model by the enforcement that all the siphons in the plant are *P*-invarient max-controlled. In fact, the control of a siphon in this stage is implemented by enforcing a generalized mutual exclusion constraint (GMEC). Consider that adding control places may generate new insufficient marked siphons. The second stage, called *control-induced siphons control*, aims to assure the controllability of the newly generated siphons, which result from the addition of the monitors in the first stage. To assure that there are no new insufficient marked siphons generated, any output arc of the monitors added in the second stage points to the source transitions of the plant model. Since the monitors added in the second stage cannot lead to problematic siphons any more, the second stage can converge rapidly with respect to the elimination of deadlock states. Note that a siphon is said to be problematic if its insufficient markedness in generalized Petri nets is tied to a deadlock. The proposed method is computationally efficient with a small number of additional monitors and makes the resultant liveness-enforcing Petri net supervisor have more permissive behavior.

The paper proceeds as follows. Section 2 introduces mathematical programming approaches for generalized Petri nets. The definitions and properties of *G*-systems and its deadlock control policy are developed in Section 3. In Section 4, a *G*-system example is given to demonstrate the proposed method, comparison of deadlock control among the different control methods is shown accordingly, and a problem of the proposed method is discussed finally. While Section 5 concludes this paper. Some basic definitions and properties of Petri nets are presented in Appendix.

2 Mathematical programming approaches for generalized Petri nets

As is well known, an empty siphon S causes that no transitions in $S^\bullet$ is enabled in an ordinary net. The classical algorithm for determining a maximal unmarked siphon S in Chu and Xie (1997) is only used for ordinary Petri nets. As for a generalized Petri net, deadlock occurs due to insufficiently marked siphons. Insufficient marked siphon implies a dead transition in a net. This section presents conditions for checking whether there exist a maximal deadly marked siphons at a given marking M in (N, M) or not in a generalized Petri net. Similarly, the maximal deadly marked siphon at a given marking M can be determined by an MIP problem.

Definition 1 (Ohta and Tsuji 2003) Let (N, M_0) be a Petri net and M be a reachable marking of N. Insufficient marked siphon is a pair (y, k) of $|P|$-dimensional rational number vector $y \geq 0$ and rational number $k > 0$ satisfying the following condition, i.e., $\forall j \in J$, $(y^T[N]_j \leq 0) \vee (y^T[N]_j^- \geq k) \wedge (y^T M < k)$, where $J = \{1, 2, \ldots, |T|\}$ is the set of indices of the transitions.

Definition 2 (Park and Reveliotis 2001) Let (N, M_0) be a Petri net, where $N = (P, T, F, W)$. S is a maximal deadly marked siphon at M, if $\forall p \notin S$, $M(p) > W(p, t) - 1$.

Definition 3 Let (N, M_0) be a structurally bounded net, and $B(p)$ be the structural bound of place p in N. Let $v_p \geq (M(p) - W(p, t) + 1)/B(p)$, where $B(p) = \max\{M(p) \mid M = M_0 + [N]Y, M \geq 0, Y \geq 0\}$.

Note that structural bound can be determined by using any LP software. As shown in Chu and Xie (1997), finding a siphon S in N is the solution of a mixed integer programming problem after introducing two binary variables v_p and z_t, which are defined as follows:

$v_p = 1\{p \notin S\} \Longleftrightarrow$ place p is removed during the algorithm execution, $\forall p \in P$, $\forall W(p, t) > 0$.
$z_t = 1\{t \notin S^\bullet\} \Longleftrightarrow$ transition t is removed during the algorithm execution, $\forall t \in T$.

Now we present the MIP detection method for generalized Petri nets as follows.

Theorem 1 *Given a marking $M \in R(N, M_0)$ of a structurally bounded Petri net (N, M_0), the maximal deadly marked siphon S contained in M is determined by the following IP formulation. The maximal deadly marked siphon is obtained with $S = \{p_i | p_i \in P \wedge v_{p_i} = 0, i = 1, 2, \cdots, |P|\}$.*

$$G^{\mathrm{MIP}} = \min \sum_{p \in P} v_p$$

s.t.

$$\begin{cases} z_t \geq \sum_{p \in {}^\bullet t} v_p - |{}^\bullet t| + 1, \forall t \in T. & (1) \\ v_p \geq z_t, \forall (t, p) \in F. & (2) \\ v_p, z_t \in \{0, 1\}. & (3) \\ v_p \geq (M(p) - W(p, t) + 1)/B(p), \forall p \in P, \forall W(p, t) > 0. & (4) \\ M = M_0 + [N]Y, M \geq 0, Y \geq 0. & (5) \end{cases}$$

Proof In order to understand the formulation of this theorem, notice that Eqs. 1 and 4 imply that any fireable transition t at M has $z_t = 1$. Furthermore, Eq. 2 implies that any place $p \in t^\bullet$ for some t with $z_t = 1$ has $v_p = 1$. Finally, the result that no additional place p (resp., transition t) has $v_p = 1$ (resp., $z_t = 1$), it is guaranteed by the specification of the objective function in the above formulation. Conversely, for any solution of the linear system, let $S = \{p \in P | v_p = 0\}$. For any $t \in {}^\bullet S$, from Eq. 2, we have $z_t = 0$. From Eq. 1, we have $\sum_{p \in {}^\bullet t} v_p \leq |{}^\bullet t| - 1$ which implies that $v_p = 0$, i.e., $p \in S$, at least one place $p \in {}^\bullet t$. As a result, there exists a maximal deadly siphon S iff $G^{\mathrm{MIP}} < |P|$. □

Theorem 2 *Let* (N, M_0) *with* $N = (P, T, F, W)$ *be a net system.* (N, M_0) *contains no maximal deadly marked siphon if* $G^{\mathrm{MIP}} = |P|$.

Proof Based on the definition of the maximal deadly marked siphon, from Theorem 1, we conclude that (N, M_0) contains no maximal deadly siphon if $G^{\mathrm{MIP}} = |P|$. □

The MIP-based method is a fast deadlock detection approach for structurally bounded nets whose deadlocks are tied to deadly marked siphons in generalized nets. The advantage of the MIP-based deadlock detection method is its high efficiency. That is to say, the solution to MIP does not need the complete siphon enumeration, leading to better computational efficiency. It is able to find a maximal siphon deadly marked at a reachable marking. Based on above results, we can conclude that there is no maximal deadly siphon in a net system (N, M), if $G^{\mathrm{MIP}}(M) = |P|$ is true.

Consider the system in Fig. 1. Based on Theorem 1, Lindo (2011), a commercial mathematical programming software package, gives a feasible solution with a maximal deadly marked siphon $S = \{p_3, p_4, p_6, p_7, p_8, p_9, p_{11} - p_{17}\}$ at some marking M. We can derive a minimal siphon from it by Algorithm 1 as below.

Consider the example in Fig. 1. Firstly, a maximal deadly marked siphon $S = \{p_3, p_4, p_6, p_7, p_8, p_9, p_{11} - p_{17}\}$ is obtained by the MIP method. If p_3 is chosen, we have $S_m = \{p_4, p_6, p_7, p_8, p_9, p_{11} - p_{17}\}$. The subnet due to $(S_m, {}^\bullet S_m \cup S_m^\bullet)$ has a source transition t_4 since ${}^\bullet t_4 = \emptyset$. Note that $t_4{}^\bullet = \{p_4, p_{17}\}$. As a result, p_4 and p_{17} can be removed from S_m due to Algorithm 1. Their removal leads to source transitions t_3

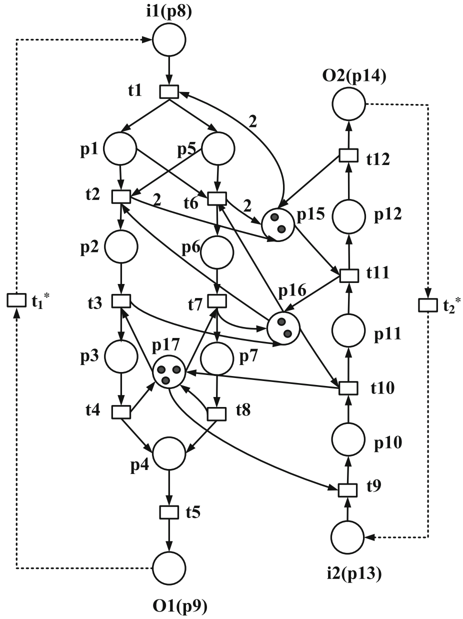


Fig. 1 A small G-system net model.

and t_5 in the resultant subnet. Accordingly, p_9 and p_{16} can be removed, which leads to source transitions t_6, t_{10} and t_1^*, then p_6, p_{11}, p_{15} and p_8 can be removed, which leads to source transitions t_7 and t_{11}, after that p_7 and p_{12}, which can be removed which leads us to have an empty set S_m. That is to say, the removal of p_3 leads to removing all places in $(S, {}^\bullet S \cup S^\bullet)$. Hence, we can say that p_3 is in any minimal siphon that can be derived from S. Likewise, p_7, p_{16} and p_{17} are necessarily contained

Algorithm 1 Finding a minimal siphon from a maximal deadly marked siphon S

Input: a maximal deadly marked siphon S
Output: a minimal siphon $S_{\min}$

$S_{\min} := \emptyset$.
while $(S_{\min} \neq S)$ **do**
    Choose a place p^* from $S \backslash S_{\min}$
    $S_m := S \backslash \{p^*\}$
    while ($\exists$ source transition without input arcs t in $(S_m, {}^\bullet S_m \cup S_m{}^\bullet)$) **do**
        $S_m := S_m \backslash \{p \mid p \in t^\bullet\}$
    end while
    if $S_m \neq \emptyset$ **then**
        $S := S_m$
    else
        $S_{\min} := S_{\min} \cup \{p^*\}$
    end if
end while
Output $S_{\min}$

in any minimal siphon that can be derived from S. Otherwise expressed, we have $S_i \cap S_j = \{p_3, p_7, p_{16}, p_{17}\}$, where S_i and S_j are siphons that can be derived from S with $i \neq j$.

Suppose that in a scenario we have already had $S_{\min} = \{p_3, p_7, p_{16}, p_{17}\}$. Let $p^* = p_{11}$, its removal cannot result in a source transition. Let $p^* = p_{15}$. Its removal can lead to a source transition t_{11}. It is easy to verify that the removal of p_{15} can remove all places. As a result, p_{15} is put into $S_{\min}$. Likewise, p_{12} is put into $S_{\min}$. Therefore, the minimal siphon that can be derived is $S_{\min} = \{p_3, p_7, p_{12}, p_{15}, p_{16}, p_{17}\}$.

Suppose that in the same scenario we have already had $S_{\min} = \{p_3, p_7, p_{16}, p_{17}\}$. Let $p^* = p_{15}$, its removal cannot result in a source transition. However, the removal of p_{11} can lead to the removal of all places. As a result, p_{11} is an element of the minimal siphon $S_{\min}$ derived in this case, where $S_{\min} = \{p_3, p_7, p_{11}, p_{16}, p_{17}\}$. In a word, the algorithm can derive a minimal siphon from a maximal deadly marked one, depending on the order of choosing the places in set $S \setminus S_{\min}$. That is to say, if the removing sequence of places in net S_m is different, we can obtain different minimal siphons.

3 G-systems and deadlock control policy

3.1 *G*-systems

In this subsection, we introduce a class of resource allocation systems (RAS), namely *G*-systems (Barkaoui et al. 1997) that can be viewed as a Petri net describing a general problem arising in many contemporary application domains such as FMS and workflow management systems.

Definition 4 A *G*-task is a net $GT = (N, M_0, M_F)$, where

(1) $N = (P, T, F, W)$ is a circuit-free net with two special places i and o. Place i is a source place ($^\bullet i = \emptyset$) and o is a sink place ($o^\bullet = \emptyset$).
(2) The augmented net N^* is obtained from N by adding a transition t^* such that $\{t^*\} = o^\bullet$ and $\{t^*\} = {}^\bullet i$ with $W(o, t^*) = W(t^*, i) = n$ is strongly connected.
(3) $M_0 = n.i$, which denotes an initial marking, and $M_F = n.o$, which denotes a finite set of final markings.
(4) $(N, n.i)$ is quasi-live.

Definition 5 A *G*-task $GT = (N, n.i)$ is sound if (1) $\forall M \in R(N, n.i), n.o \in R(N, n.i)$ and (2) $\forall M \in R(N, n.i)$, $M(o) \geq n \Rightarrow M = n.o$.

Definition 6 A *G*-task system with resources GTR is a marked net $(N, M_{\mu 0}, M_{\mu F})$, where

(1) $N = (P \cup P_R, T, F \cup F_R, W \cup W_R)$.
(2) P_R is the set of resources with $P_R \neq \emptyset$ and $P \cap P_R = \emptyset$.
(3) $F_R \subseteq (P_R \times T) \cup (T \times P_R)$ is the flow relation for resources.
(4) $\forall u \in F_R$, $W_R(u) \geq 1$.
(5) $\forall r \in P_R, \exists I_r \neq \mathbf{0}$, $I_r^T[N] = \mathbf{0}^T$ and $||I_r|| \cap P_R = \{r\}$.

(6) $M_{\mu 0} = n.i + \sum h_j.r_j$, $M_{\mu F} = n.o + \sum h_j.r_j$, $o, i \in P$, $r_j \in P_R$, $j \in \{1, \cdots, |P_R|\}$.
(7) The subsystem $G = (N, M_0, M_F)$ is a consistent G-task, where $N = (P, T, F, W)$, and M_0 and M_F are the restrictions on $M_{\mu 0}$ and $M_{\mu F}$ respectively with respect to P.

A GTR system is basically a consistent G-task plus a set of places (P_R) modeling the resources shared by its processes. We require a G-task net with resources to be (externally) sound with respect to resource use, i.e., a requested resource is eventually released and a released resource has previously been requested. This resource preservation property is expressed in terms of invariants in the system. Due to the structure of G-task subsystems, subnets induced by these invariants are not necessarily state machines. In a GTR, it is allowed that several resources can be requested or released simultaneously.

Then we can compose several GTR nets into a system, where they share resources. This is obtained by utilizing the places representing the resources shared with different GTR systems.

Definition 7 A G-system GS is recursively defined as follows:

(1) A GTR is a G-system.
(2) Let $GS_i = (NS_i, MS_{0i}, MS_{Fi})$ ($i \in \{1, 2\}$) be two G-systems such that $P_1 \cap P_2 = T_1 \cap T_2 = \emptyset$. We denote the set of shared resources by $P_{R1}P_{R2} = P_{R1} \cap P_{R2}$.
(3) System GS is a G-system resulting from the fusion of two systems GS_1 and GS_2 over the resource set $P_{R1}P_{R2}$, which is denoted as $GS = GS_1 \circ GS_2$.

Definition 8 Let S be a strict minimal siphon in $N = (P \cup P_R, T, F \cup F_R, W \cup W_R)$. $\forall r \in P_R$, I_r is a minimal P-semiflow. Let $\Omega_S = \sum_{r \in S \setminus (S \cap P_A)} I_r$, $\Omega'_S = \sum_{p \in S} \Omega_S(p)p$, where $P_A = P \setminus (\cup_{i=1}^{k} \{i_i, o_i\})$ is called the set of operation places. We define $Th_S = \Omega_S - \Omega'_S$.

It is easy to see that $Th_S \subseteq P_A$ is true. Let $\sum_{p \in Th_S} h_S(p)p$ denote Th_S, where $h_S(p)$ indicates that siphon S loses $h_S(p)$ tokens if the number of tokens in p increases by unit.

A relevant property of any system with shared resources is non-blocking, i.e, from any reachable state, it is always possible to reach a desirable (or final) state. A GS (N_S, M_0) can be augmented by adding a transition t^* to its each G-task. We denote the resultant net system by (N_S^*, M_0).

Theorem 3 (Barkaoui et al. 1997) *An augmented GS is live iff* (N_S^*, M_0) *is bounded and satisfies max cs-property.*

The net system shown in Fig. 1 is a G-system with $i_1 = p_8$, $o_1 = p_9$, $i_2 = p_{13}$, $o_2 = p_{14}$, and the augmented net system is obtained by adding transitions with ${}^\bullet i_1 = o_1{}^\bullet = t_1^*$, ${}^\bullet i_2 = o_2{}^\bullet = t_2^*$. Moreover, we have $P_{A_1} = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_9\}$, $P_{A_2} = \{p_{10}, p_{11}, p_{12}, p_{14}\}$, $P_{R_1} = \{p_{15}, p_{16}, p_{17}\}$, and $P_{R_2} = \{p_{15}, p_{16}, p_{17}\}$. It has three strict minimal siphons: $S_1 = \{p_3, p_7, p_{12}, p_{15}, p_{16}, p_{17}\}$, $S_2 = \{p_2, p_6, p_{12}, p_{15}, p_{16}\}$, and $S_3 = \{p_3, p_7, p_{11}, p_{16}, p_{17}\}$. We have $I_{p_{15}} = p_1 + p_5 + p_{12} + p_{15}$, $I_{p_{16}} = p_2 + p_6 +$

$p_{11} + p_{16}$ and $I_{p_{17}} = p_3 + p_7 + p_{10} + p_{17}$, then $Th_{S_3} = (I_{p_{16}} + I_{p_{17}}) - (p_3 + p_7 + p_{11} + p_{16} + p_{17}) = p_2 + p_6 + p_{10}$. It is easy to see that one token is removed from S_3 if the number of tokens in p_2 increases by unit.

3.2 Iteration deadlock control policy

A siphon-based methodology plays an important role in the development of liveness-enforcing Petri net supervisors for resource allocation systems (Reveliotis 2003). For a particular class of Petri nets, G-systems, we develop a two-stage deadlock prevention method based on the MIP approach above discussions.

Definition 9 A string $x_1, x_2, \cdots, x_n$ $(x_i \in P \cup T)$ is called a path of N if $\forall i \in \{1, 2, \cdots, n-1\}$, $x_{i+1} \in x_i^\bullet$. An elementary path from x_1 to x_n is a path whose nodes are all different (except, perhaps, $x_1 = x_n$), which is denoted by $EP(x_1, x_n)$. A path $x_1, x_2, \cdots, x_n$ is called a circuit if it is an elementary path and $x_1 = x_n$.

C is defined as a circuit of a net N, and $C(x)$ is a circuit including node x. The support of a path $EP(x_1, x_n)$ and a circuit C is defined as the set of its node, denoted as $\|EP(x_1, x_n)\|$ and $\|C\|$, respectively.

Lemma 1 (Barkaoui and Pradat-Peyre 1996) *Let* (N, M_0) *be a marked net and* S *be a siphon of* N. S *is max-controlled if there exists a P-invariant* I *such that* $\forall p \in (||I||^- \cap S)$, $max_{p^\bullet} = 1$, $||I||^+ \subseteq S$, $\sum_{p \in P} I(p) M_0(p) > \sum_{p \in S} I(p)(max_{p^\bullet} - 1)$.

Definition 10 Let S be a strict minimal siphon in a G-system plant net $(N_{\mu 0}, M_{\mu 0})$, where $N_{\mu 0} = (P_0 \cup P_A \cup P_R, T_0, F_0, W_0)$, P_0 is the set of sink and source places, P_A is the set of operation places, and P_R is the set of resource places. A monitor V_S is added to $(N_{\mu 0}, M_{\mu 0})$, leading to an extended net system $(N_{\mu 1}, M_{\mu 1})$ with $N_{\mu 1} = (P_0 \cup P_A \cup P_R \cup \{V_S\}, T_1, F_1, W_1)$. Let $\{\alpha, \beta, \cdots, \gamma\} \subseteq \{1, 2, \cdots, k\}$ such that $\forall i \in \{\alpha, \beta, \cdots, \gamma\}$, $Th(S) \cap P_{A_i} \neq \emptyset$ and $\forall j \in \{1, 2, \cdots, k\} \backslash \{\alpha, \beta, \ldots, \gamma\}$, $Th(S) \cap P_{A_j} = \emptyset$. For a siphon S, $\forall p \in P_A \cup P_0$, a non-negative integer $k_S(p)$ is defined by the following steps.

- Step 1 $\forall p \in P_A \cup P_0, k_S(p) := 0$;
- Step 2 $\forall p \in Th(S), k_S(p) := h_S(p)$, where $Th(S) = \sum_{p \in Th(S)} h_S(p) p$;
- Step 3 $\forall i \in \{\alpha, \beta, \cdots \gamma\}$, let $p_s \in Th(S) \cap P_{A_i}$ be such a place that $\forall p_t \in EP(p_u, p_{0i})$, $p_u \in p_s^{\bullet\bullet}$, $p_t \notin Th(S)$. We assume that there are m such places, $p_s^1, p_s^2, \cdots$, and p_s^m. Certainly, we have $\{p_s^i | i = 1, 2, \cdots, m\} \subseteq Th(S) \cap P_{A_i}$. $\forall p_s^i$, let p_v be a place in $EP(p_{0i}, p_s^i)$ such that $h_S(p_v) \geq h_S(p_w)$, $\forall p_w \in EP(p_{0i}, p_s^i)$. $\forall p_v, \forall p_x \in EP(p_{0i}, p_v), k_S(p_x) := h_S(p_v)$. $\forall p_y \in \cap_{i=1}^m EP(p_{0i}, p_s^i), k_S(p_y) := h_S(p_z^i)$, where $p_z^i \in Th(S) \cap P_{A_i}$, $\nexists p \in Th(S) \cap P_i, h_S(p) > h_S(p_z^i)$;
- Step 4 Let $K_S = \{p | k_S(p) \neq 0,\ p \notin Th(S)\}$ be the set of places, which is corresponding to k_S.

In the net shown in Fig. 1, let $P_{A_1} = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_9\}$ and $P_{A_2} = \{p_{10}, p_{11}, p_{12}, p_{14}\}$. For unmarked minimal siphon $S = \{p_3, p_7, p_{11}, p_{16}, p_{17}\}$, we have $Th(S) = p_2 + p_6 + p_{10}$. For P_{A_1}, we have $p_s^1 = p_2$, $p_s^2 = p_6$, $p_v = p_1$, $p_z^1 =$

p_2, and $p_y = p_1$. While for P_{A_2}, we have $p_s = p_{10}$ and $p_z^2 = p_{10}$. As a result, we have $k_{S_1}(p_1) = 1$, $k_{S_1}(p_2) = 1$, $k_{S_1}(p_6) = 1$, $k_{S_1}(p_{10}) = 1$. $\forall p \in P_A \backslash \{p_1, p_2, p_6, p_{10}\}$, $k_{S_1}(p) = 0$, then we have $K_{S_1} = \{p_1\}$.

Proposition 1 (Barkaoui et al. 1997; Li and Zhao 2008) *Let S be a strict minimal siphon in a marked G-system $(N_{\mu 0}, M_{\mu 0})$, where $N_{\mu 0} = (P_0 \cup P_A \cup P_R, T_0, F_0, W_0)$. Construct k_S for S due to Definition 10. A monitor V_S is added to $(N_{\mu 0}, M_{\mu 0})$ by the enforcement that $g_S = k_S + V_S$ is a P-invariant of the resultant net system $(N_{\mu 1}, M_{\mu 1})$, where $N_{\mu 1} = (P_0 \cup P_A \cup P_R \cup \{V_S\}, T_1, F_1, W_1)$, $\forall p \in P_0 \cup P_A \cup P_R$, $M_{\mu 1}(p) = M_{\mu 0}(p)$. Let $h_S = \sum_{r \in S_R} I_r - g_S$ and $M_{\mu 1}(V_S) = M_{\mu 0}(S) - \xi_S$ ($\xi_S \in \mathbf{N}^+$). Then S is max-controlled if $\xi_S > \sum_{p \in S} h_S(p)(max_{p^\bullet} - 1)$.*

As a design parameter, ξ_S is called the control depth variable of siphon S. Proposition 1 indicates that a strict minimal siphon can be max-controlled by adding a monitor for it and by properly supervising its initial tokens. In order to have more permissive behavior in a liveness-enforcing supervisor, every control depth variable should be minimized on the condition that all siphons satisfy the max cs-property.

Let us consider the minimal siphon $S = \{p_3, p_7, p_{11}, p_{16}, p_{17}\}$ derived by Algorithm 1. From Definition 10, we have $k_S = p_1 + p_2 + p_6 + p_{10}$, as a result, $g_S = k_S + V_S = p_1 + p_2 + p_6 + p_{10} + V_S$. Noticing that $\sum_{r \in S_R} I_r = I_{p_{16}} + I_{p_{17}} = p_2 + p_6 + p_{11} + p_{16} + p_3 + p_7 + p_{10} + p_{17}$, we have $h_S = p_{11} + p_{16} + p_3 + p_7 + p_{17} - p_1 - V_S$. According to Proposition 1, V_S is added such that h_S is a P-invariant of N_μ. Let $\xi_S = 1$. Clearly, we have $\xi_S > \sum_{p \in S} h_S(p)(max_{p^\bullet} - 1) = 0$. S is hence max-controlled by adding monitor V_S.

Proposition 2 *The siphon control method in Proposition 1 does not generate new non-max-controlled siphons in $(N_{\mu 1}, M_{\mu 1})$.*

Proof As is well known, an empty strictly minimal siphon was main incentive of deadlock in an ordinary net. Similarly, insufficiently marked siphon can lead to deadlock state for a generalized net. In order to ensure all strictly minimal siphons satisfying cs-property, we should add monitors for insufficiently marked siphons. Suppose that monitor V_S is added to make S max-controlled by P-invariant $h_S = \sum_{r \in S_R} I_r - g_S$. By Proposition 1, $g_S = k_S + V_S = \sum_{p \in Th(S)} k_S(p)p + \sum_{p \in K_S} k_S(p)p + V_S$. By the construction of k_S, if $V_S^\bullet \cap T_i \neq \emptyset$, $t_i^{s\bullet} \subset ||g_S||$, where $t_i^s \in i_j{}^\bullet$, and i_j is a source place of N_i. As a result, any output arc of V_S points to source transitions of N_i. A strictly minimal siphon is composed of operation places and resource places, that is to say, it does not include idle places(source places in a G-system). Based on the definition of siphon($^\bullet S \subseteq S^\bullet$), if all output arcs of monitors point to source transition, we have $^\bullet T_i$ must include idle places, it is contradictory to the definition of siphon. Hence, the system can not generate new additional siphons if all output arcs of monitors point to source transitions. Moreover, noticing that a source transition represents the entry of a raw product into a system and an idle place does not need any resource, new circular wait, one of the four necessary conditions of deadlocks, due to the addition of V_S is impossible. That is to say, the addition of V_S does not contribute additional deadlock states in $(N_{\mu 1}, M_{\mu 1})$. Considering that the resultant net with additional monitor V_S is still a G-system and that deadlock states are led by the existence of strict minimal siphons, we can say that no new non-max-controlled siphon is produced due to V_S. □

Proposition 2 implies that if all siphons in a G-system plant model are max-controlled by adding monitors as stated in Proposition 1, the resultant net is non-blocking. Alternatively, liveness can be enforced in the augmented net system of a G-system.

Definition 11 Let U be a T-semiflow of $N_{\mu 1}$. C_U is called a induced-circuit of T-semiflow U if exists an elementary circuit C_U such that $\|U\| = \|C_U\| \cap T$ holds. if p is a node of C_U, an idle reachable path of p is defined as $EP_{C_U}(p, p_i^0)$.

As the net shown in Fig. 1 for example, there are three T-semiflows in the net, which are $U_1 = t_1 + t_2 + t_3 + t_4 + t_5 + t_{13}$, $U_2 = t_1 + t_5 + t_6 + t_7 + t_8 + t_{13}$ and $U_3 = t_9 + t_{10} + t_{11} + t_{12} + t_{14}$. Each T-semiflow can derive an elementary circuit, that is $C_{U1} = p_8 t_1 p_1 t_2 p_2 t_3 p_3 t_4 p_4 t_5 p_9 t_1^* p_8$, $C_{U2} = p_8 t_1 p_1 t_6 p_6 t_7 p_7 t_8 p_4 t_5 p_9 t_1^* p_8$ and $C_{U3} = p_{13} t_9 p_{10} t_{10} p_{11} t_{11} p_{12} t_{12} p_{14} t_2^* p_{13}$. Moreover, p_1 is a node of C_{U1}, then $EP_{C_{U1}}(p_1, p_8) = p_1 t_2 p_2 t_3 p_3 t_4 p_4 t_5 p_9 t_{13} p_8$.

Definition 12 Let S be a siphon in $(N_{\mu 1}, M_{\mu 1})$, where $p \in P_{A_i}, i \in \mathbf{N}^+$. Maximal resource requirements of p in S is denoted by $Q_S(p)$, where $Q_S(p) = \max_{q \in EP_{C_U}(p, p_i^0) \cap P_{A_i}} Th_S(q)$.

To illustrate the computation of $Q_S(p)$ for p, we take the net shown in Fig. 1 for example. The idle places are p_8 and p_{13}, $P_{A1} = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7\}$, $P_{A2} = \{p_{10}, p_{11}, p_{12}, p_{14}\}$. In the net $S_3 = \{p_3, p_7, p_{11}, p_{16}, p_{17}\}$ is a siphon with $Th_{S3} = p_2 + p_6 + p_{10}$. Firstly, we have $EP_{C_{U1}}(p_2, p_8) = p_2 t_3 p_3 t_4 p_4 t_5 p_9 t_1^* p_8$, $EP_{C_{U2}}(p_6, p_8) = p_6 t_7 p_7 t_8 p_4 t_5 p_9 t_1^* p_8$, $EP_{C_{U3}}(p_{10}, p_{13}) = p_{10} t_{10} p_{11} t_{11} p_{12} t_{12} p_{14} t_2^* p_{13}$. Based on Definition 12, $Q_S(p_1) = Th_{S3}(p_2) = 1$, $Q_S(p_2) = 1$, $Q_S(p_5) = Th_{S3}(p_6) = 1$, $Q_S(p_6) = 1$, $Q_S(p_3) = Q_S(q_7) = 0$, $Q_S(p_{10}) = 1$, $Q_S(p_{11}) = 0$, $Q_S(p_{12}) = 0$.

Next we introduce an iterative method to design a liveness-enforcing supervisor for G-systems. The method consists of two main stages: the siphons control stage adds control places to the original net such that all the siphons are controlled. The control-induced siphons control stage adds control places to the net with its output arcs to source transitions of the resultant net after the resource places are removed. Our proposed algorithm is thus presented as follows:

Theorem 4 *Algorithm* 2 *leads to a non-blocking Petri net supervisor* $(N_{\mu 1}, M_{\mu 1})$ *for a G-system plant net model* $(N_{\mu 0}, M_{\mu 0})$.

Proof This result follows from the fact that all siphons are max-controlled, i.e., it satisfies max cs-property. By Theorem 3, $(N_{\mu 1}, M_{\mu 1})$ is non-blocking. □

To prevent deadlocks, we adopt Algorithm 2 to the net shown in Fig. 1. First, a maximal deadly siphon $S^1 = \{p_3, p_4, p_6, p_7, p_8, p_9, p_{11} - p_{17}\}$ can be obtained, then two insufficiently marked minimal siphons $S_1 = \{p_3, p_7, p_{11}, p_{16}, p_{17}\}$ and $S_1' =$

Algorithm 2 Deadlock prevention policy

Input: Given a G-system net $(N_{\mu 0}, M_{\mu 0})$, $N_{\mu 0} = (P^0 \cup P_A \cup P_R, T_0, F_0, W_0)$.
Output: A non-blocking supervisor $(N_{\mu 1}, M_{\mu 1})$, $N_{\mu 1} = (P^0 \cup P_A \cup P_R \cup \{V_S\} \cup \{V_S^*\}, T_1, F_1, W_1)$.

Siphons control stage

$V_S = \emptyset, i = 1$.
Apply MIP-based deadlock detection method on $(N_{\mu 0}, M_{\mu 0})$ to obtain a maximal siphon S^i.
while $(S^i \neq \emptyset)$ **do**
  Apply Algorithm 1 to obtain a minimal siphon S_i.
  Add a control place V_{S_i} to $(N_{\mu 0}, M_{\mu 0})$ such that $V_{S_i} + \sum_{p \in S_i} p$ is a P-semiflow of $N_{\mu 0}$.
  Add an arc from V_{S_i} to $t, \forall t \in \{t \mid t \in {}^\bullet p, p \in \| Th_{S_i} \|, {}^\bullet t \cap \| Th_{S_i} \| = \emptyset\}$.
  Add an arc from t to $V_{S_i}, \forall t \in \{t \mid t \in p^\bullet, p \in \| Th_{S_i} \|, t^\bullet \cap \| Th_{S_i} \| = \emptyset\}$.
  $\forall p \in S_i$, $M_{\mu 0}(V_{S_i}) = \sum_{r \in S \cap P_R} M_{\mu 0}(r) - \max_{p^\bullet}$.
  Add a new constraint to the MIP problem of $(N_{\mu 0}, M_{\mu 0})$, that is $\sum K_S M_{\mu 0}(p) \leq M_{\mu 0}(V_{S_i}), \forall p \in Th(S_i)$.
  $V_S = V_S \cup \{V_{S_i}\}, i = i + 1$.
  Obtain G^{MIP} of $(N_{\mu 0}, M_{\mu 0})$ and solve G^{MIP} to obtain a maximal siphon S^i.
end while
Obtain $(N_{\mu 1}, M_{\mu 1})$.
$j = i + 1, V_S^* = \emptyset$.
Apply MIP to $(N_{\mu 1}, M_{\mu 1})$ to obtain a maximal siphon S_j^*.
while $(S_j^* \neq \emptyset)$ **do**
  Apply Algorithm 1 to obtain a minimal siphon S_j.

Control-induced siphons control stage

  Add a control place $V_{S_j}^*$ to $(N_{\mu 1}, M_{\mu 1})$ by Proposition 1, $V_S^* = V_S^* \cup \{V_{S_j}^*\}$.
  $\forall p \in P^0 \cup P_A \cup P_R \cup \{V_S\}$, $\forall t \in T$, $W_{\mu 1}^*(p, t) = W_{\mu 1}(p, t)$, $W_{\mu 1}^*(t, p) = W_{\mu 1}(t, p)$, $M_{\mu 1}^*(p) = M_{\mu 1}(p)$.
  $\forall t \in P^{0\bullet}$, $W_{\mu 1}^*(V_{S_j}^*, t) = Q_S(p), p \in t^\bullet \cap P_A$.
  $\forall t \in T \backslash P^{0\bullet}$, $W_{\mu 1}^*(t, V_{S_j}^*) = Q_S(p) - Q_S(p'), p \in {}^\bullet t \cap P_A, p' \in t^\bullet \cap P_A$.
  $\forall p \in S_j$, $M_{\mu 1}(V_{S_j}^*) = M_{\mu 1}(S_j) - \max_{p^\bullet}$.
  Obtain $(N_{\mu 1}, M_{\mu 1}) = (N_{\mu 1}, M_{\mu 1}) \cup V_S^*$, $j = j + 1$.
  Obtain G^{MIP} of $(N_{\mu 1}, M_{\mu 1})$, and solve G^{MIP} to obtain a maximal siphon S_j^*.
end while
Output $(N_{\mu 1}, M_{\mu 1})$.

$\{p_3, p_7, p_{12}, p_{15}, p_{16}, p_{17}\}$ can be derived from S^1 by Algorithm 1. If S_1 is chosen, we have $Th_{S_1} = p_2 + p_6 + p_{10}$ and $M_0(V_{S1}) = M_0(p_{16}) + M_0(p_{17}) - 1 = 4$. Based on Algorithm 2, a constraint is added to original net $(N_{\mu 0}, M_{\mu 0})$, that is $M(p_2) + M(p_6) + M(p_{10}) \leq 4$. The details in other iterations concerning the siphons control are given as follows: $S^2 = \{p_2, p_3, p_4, p_6, p_7, p_9, p_{12} - p_{16}\}$, $S_2 = \{p_2, p_6, p_{12}, p_{15}, p_{16}\}$; $Th_{S_2} = 2p_1 + p_{11}$, s.t.2: $2M(p_1) + M(p_{11}) \leq 2$.

After two iterations in the *siphons control* stage, the original linear programming problem is:

$$G^{\mathrm{MIP}} = \min \sum\nolimits_{i=1}^{17} v_p$$

s.t.

$$\begin{cases} z_t \geq \sum\nolimits_{p \in {}^{\bullet}t} v_p - |{}^{\bullet}t| + 1, \forall t \in T. \\ v_p \geq z_t, \forall (t, p) \in F. \\ v_p, z_t \in \{0, 1\}. \\ v_p \geq (M(p) - W(p, t) + 1)/B(p), \forall p \in P, \forall W(p, t) > 0. \\ M = M_0 + [N]Y, M \geq 0, Y \geq 0. \\ M(p_2) + M(p_6) + M(p_{10}) \leq 4 \\ 2M(p_1) + M(p_{11}) \leq 2 \end{cases}$$

Lindo (2011), a famous commercial mathematical programming software package, gives $G^{\mathrm{MIP}} = 17$, two control places V_{S1} and V_{S2} with the associated arcs added to the net such that no maximal deadly marked siphon can be obtained in $(N_{\mu0}, M_{\mu0})$, as shown in Fig. 2, the augmented net is denoted as $(N_{\mu1}, M_{\mu1})$. After the *siphons control* stage, a new maximal deadly marked siphon $S_3^* = \{p_2 - p_4, p_7, p_9, p_{11} - p_{15}, p_{17} - p_{19}\}$ is obtained in $(N_{\mu1}, M_{\mu1})$, so we have $S_3 = \{p_3, p_7, p_{11}, p_{17} - p_{19}\}$,

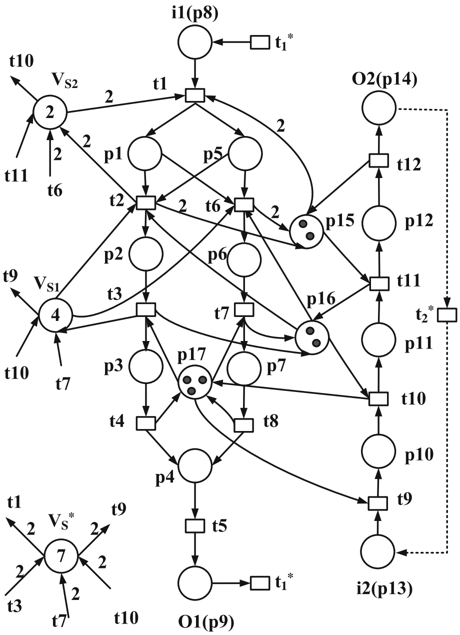


Fig. 2 The non-blocking *G*-system

$Th_{S_3} = 2p_1 + p_2 + p_6 + 2p_{10}$, and s.t.3: $2M(p_1) + M(p_2) + M(p_6) + 2M(p_{10}) \leq 7$. From Definition 12, we have $Q_{S_3^*}(p_1) = Q_{S_3^*}(p_2) = 2$, $Q_{S_3^*}(p_6) = 2$ and $Q_{S_3^*}(p_{10}) = 2$. V_S^* is added to $(N_{\mu 1}, M_{\mu 1})$ with ${}^\bullet V_S^* = \{2t_3, 2t_7, 2t_{10}\}$ and $V_S^{*\bullet} = \{2t_1, 2t_9\}$, $M_{\mu 1}(V_S^*) = M_{\mu 1}(S_3) - \max_{p^\bullet} = 9 - 2 = 7$. Add the control place V_S^* to the net system $(N_{\mu 1}, M_{\mu 1})$, we have $G^{\mathrm{MIP}} = 19$, which implies that there is no deadly marked siphon in the augmented net $(N_{\mu 1}, M_{\mu 1})$. After the *Control-induced siphons control* stage, the net is live, denoted as $(N_{\mu 1}^*, M_{\mu 1}^*)$ shown in Fig. 2.

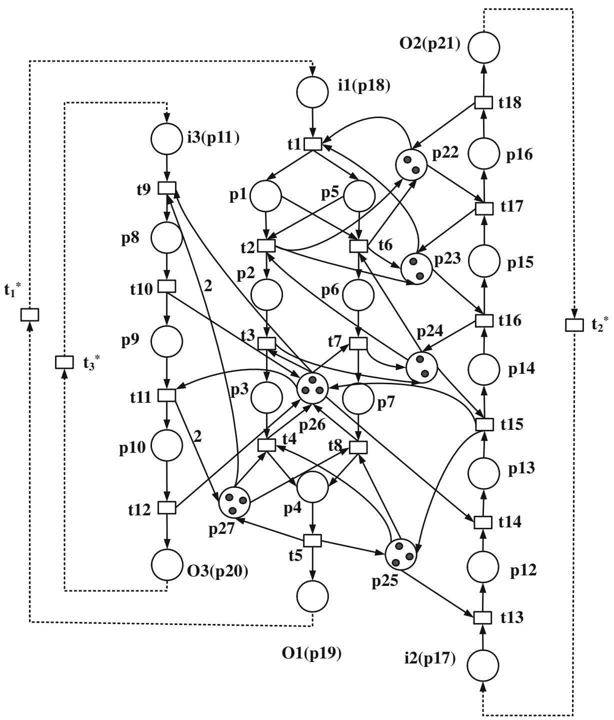
4 An example

In this section, we first consider a typical example to illustrate the definitions, theorems and algorithms in above sections. Next comparisons are made between the method in Li and Zhao (2008) and the proposed method by using the same example. Finally, a problem of the proposed method is discussed.

4.1 Application of deadlock prevention policy

Now we apply the results in this paper to the G-system shown in Fig. 3, which uses a multi-set of resources at each processing step. The net has three source places i_1

Fig. 3 A G-system net $(N_{\mu 0}, M_{\mu 0})$



(p_{18}), i_2 (p_{17}), and i_3 (p_{11}); three sink places o_1 (p_{19}), o_2 (p_{21}), and o_3 (p_{20}); and six resource places $P_R = \{p_{22}, p_{23}, p_{24}, p_{25}, p_{26}, p_{27}\}$, others are operation places. The augmented net is strongly connected after adding three transitions t_1^*, t_2^* and t_3^*, respectively.

Firstly, six resources exist in this system leading to eight minimal P-semiflows concerning resources:

$$I_{p22} = p_1 + p_{16} + p_{22},$$
$$I'_{p22} = p_5 + p_{16} + p_{22}, \quad M_{\mu 0}(p_{22}) = 2;$$
$$I_{p23} = p_1 + p_{15} + p_{23},$$
$$I'_{p23} = p_5 + p_{15} + p_{23}, \quad M_{\mu 0}(p_{23}) = 2;$$
$$I_{p24} = p_2 + p_6 + p_{14} + p_{24}, \quad M_{\mu 0}(p_{24}) = 2;$$
$$I_{p25} = p_4 + p_{12} + p_{13} + p_{25}, \quad M_{\mu 0}(p_{25}) = 3;$$
$$I_{p26} = p_3 + p_7 + p_8 + p_{10} + p_{13} + p_{26}, \quad M_{\mu 0}(p_{26}) = 3;$$
$$I_{p27} = p_4 + 2p_8 + 2p_9 + p_{27}, \quad M_{\mu 0}(p_{27}) = 3.$$

To prevent deadlocks, we adopt Algorithm 2 to the net shown in Fig. 3. First, we can obtain a maximal deadly siphon $S^1 = \{p_1, p_2, p_4 - p_{10}, p_{13} - p_{21}, p_{25}, p_{26}\}$, from this siphon, an insufficient marked minimal siphon $S_1 = \{p_4, p_8, p_{10}, p_{13}, p_{25}, p_{26}\}$ can be obtained by Algorithm 1, then we have $Th_{S_1} = p_3 + p_7 + p_{12}$ and $M_0(V_{S1}) = M_0(p_{25}) + M_0(p_{26}) - 1 = 5$, so a constraint is added to the original linear programming of the net $(N_{\mu 0}, M_{\mu 0})$, that is $M(p_3) + M(p_7) + M(p_{12}) \leq 5$, so we have $G^{\text{MIP}} = 8$. That is to say, there are maximal deadly marked siphons in $(N_{\mu 0}, M_{\mu 0})$, other monitors are needed to control insufficient marked minimal siphons. The details in other iterations concerning the siphons control are given as follows:

1. $G^{\text{MIP}} = 8$;
   $S^2 = \{p_2, p_3, p_4, p_7 - p_{10}, p_{12}, p_{14} - p_{21}, p_{24} - p_{26}\}$;
   $S_2 = \{p_4, p_8, p_{10}, p_{14}, p_{24}, p_{25}, p_{26}\}$;
   $Th_{S_2} = p_2 + p_3 + p_6 + p_7 + p_{12} + 2p_{13}$;
   $M_0(V_{S2}) = M_0(p_{24}) + M_0(p_{25}) + M_0(p_{26}) - 1 = 7$;
   s.t. 2: $M(p_2) + M(p_3) + M(p_6) + M(p_7) + M(p_{12}) + 2M(p_{13}) \leq 7$
2. $G^{\text{MIP}} = 15$
   $S^3 = \{p_2, p_3, p_6, p_7, p_{12}, p_{15} - p_{17}, p_{21} - p_{24}\}$;
   $S_3 = \{p_2, p_6, p_{15}, p_{23}, p_{24}\}$;
   $Th_{S_3} = p_1 + p_{14}$;
   $M_0(V_{S3}) = M_0(p_{23}) + M_0(p_{24}) - 1 = 3$;
   s.t. 3: $M(p_1) + M(p_{14}) \leq 3$

After three iterations in the *siphons control* stage, the original linear programming problem is:

$$G^{\mathrm{MIP}} = \min \sum_{i=1}^{27} v_p$$

s.t.

$$\begin{cases} z_t \geq \sum_{p \in {}^\bullet t} v_p - |{}^\bullet t| + 1, \forall t \in T. \\ v_p \geq z_t, \forall (t, p) \in F. \\ v_p, z_t \in \{0, 1\}. \\ v_p \geq (M(p) - W(p, t) + 1)/B(p), \forall p \in P, \forall W(p, t) > 0. \\ M = M_0 + [N]Y, M \geq 0, Y \geq 0. \\ M(p_3) + M(p_7) + M(p_{12}) \leq 5 \\ M(p_2) + M(p_3) + M(p_6) + M(p_7) + M(p_{12}) + 2M(p_{13}) \leq 7 \\ M(p_1) + M(p_{14}) \leq 3 \end{cases}$$

Lindo (2011) gives $G^{\mathrm{MIP}} = 27$, three control places V_{S1}, V_{S2} and V_{S3} with the associated arcs added to the net such that no deadly siphon in $(N_{\mu 0}, M_{\mu 0})$ can be obtained, as shown in Fig. 4, the augmented net is denoted as $(N_{\mu 1}, M_{\mu 1})$.

After the *siphons control* stage, Apply the MIP to the augmented net $(N_{\mu 1}, M_{\mu 1})$, the new linear programming problem is:

$$G^{\mathrm{MIP1}} = \min \sum_{i=1}^{30} v_p$$

s.t.

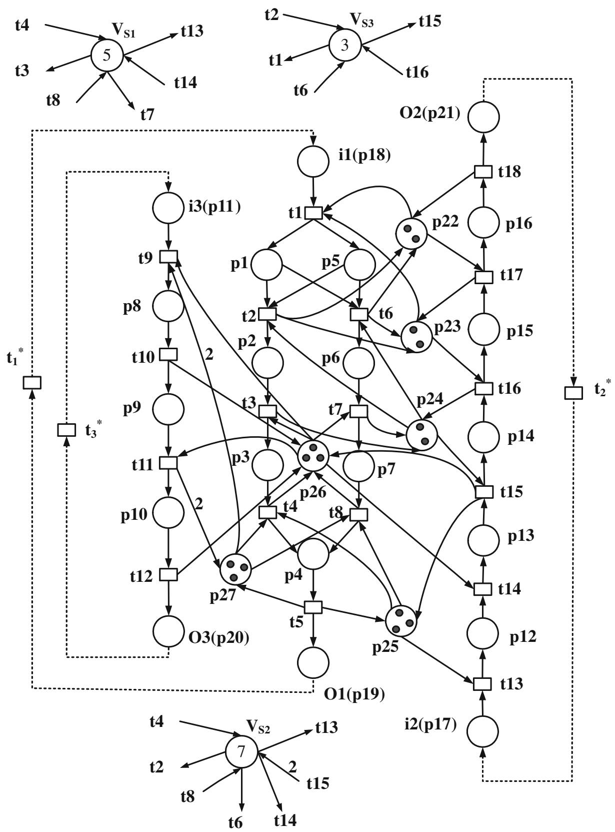
$$\begin{cases} z_t \geq \sum_{p \in {}^\bullet t} v_p - |{}^\bullet t| + 1, \forall t \in T. \\ v_p \geq z_t, \forall (t, p) \in F. \\ v_p, z_t \in \{0, 1\}. \\ v_p \geq (M(p) - W(p, t) + 1)/B(p), \forall p \in P, \forall W(p, t) > 0. \\ M = M_0 + [N]Y, M \geq 0, Y \geq 0. \end{cases}$$

We have $G^{\mathrm{MIP1}} = 30$, which implies that there is no deadly marked siphon in the resultant net $(N_{\mu 1}, M_{\mu 1})$. After the *Control-induced siphons control* stage, the net is live, denoted as $(N_{\mu 1}, M_{\mu 1})$ shown in Fig. 4.

4.2 Comparison of deadlock control methods

As is well known, computational complexity, behavior permissiveness and structural complexity are major criteria when designing a liveness-enforcing Petri net supervisor for a plant model. Most existing deadlock prevention polices in the literature for *G*-systems are not maximally permissive (Barkaoui et al. 1997; Li and Zhao 2008).

Fig. 4 A G-system supervisor $(N^*_{\mu 1}, M^*_{\mu 1})$



In this subsection, a comparison is made, via the example in Fig. 3, between the proposed method and that in Li and Zhao (2008).

For the plant net model shown in Fig. 3, the plant model has 123,595 reachable states, 123,060 of which are legal, i.e., either good or dangerous states. Based on the method in Li and Zhao (2008), the net have 13 minimal siphons, five of those are elementary siphons, others are dependent ones. As a result, the elementary siphon-based supervisor of the example has five monitors only for elementary siphons, namely $V_{S_1}, V_{S_2}, \cdots, V_{S_5}$, as shown in Table 1. The corresponding controlled system has 53788 reachable states. It shows that the supervisor obtained by the policy in Li and Zhao (2008) can provide 44% (53,788/123,060) of the optimal behavior. It is easy to see that the supervisor permissive is restrictive.

Table 1 Monitors for the net system in Fig. 3 due to Li and Zhao (2008)

V_S	*Pre*	*Post*	M_0
V_{S_1}	t_4, t_8, t_{14}	t_1, t_{13}	5
V_{S_2}	$t_4, t_8, 2t_{11}$	$2t_9, t_1$	4
V_{S_3}	$2t_2, 2t_6, t_{17}$	$2t_1, t_{13}$	5
V_{S_4}	t_2, t_6, t_{16}	t_1, t_{13}	3
V_{S_5}	t_3, t_7, t_{15}	t_1, t_{13}	1

Table 2 Monitors for the net system in Fig. 3 by the proposed method

V_S	*Pre*	*Post*	M_0
V_{S_1}	t_4, t_8, t_{14}	t_3, t_7, t_{13}	5
V_{S_2}	$t_4, t_8, 2t_{15}$	t_2, t_6, t_{13}, t_{14}	7
V_{S_3}	t_2, t_6, t_{16}	t_1, t_{15}	3

For the model shown in Fig. 3, the proposed iteration method computes three monitors, namely V_{S_1}, V_{S_2} and V_{S3} as shown in Table 2. It can be verified that the controlled system has 123060 reachable states. It means that the corresponding supervisor obtained by the proposed method can provide 100% (123,060/123,060) of the optimal permissive behavior. Table 3 shows the performance comparison of the proposed method and the one in Li and Zhao (2008). Consequently, we notice that some monitors computed by the elementary siphon-based method are still redundant. This means that the number of additional monitors can be further reduced. Though the approach in Li and Zhao (2008) lead to structurally simple liveness-enforcing monitor-based supervisors using elementary siphons, complete siphon enumeration-based methods are infeasible when the size of a plant net model is large, since it suffers from the computational complexity and behavior permissiveness issues.

4.3 Discussion

The major disadvantage of the existing deadlock control policies is that too many monitors are added, which leads to a much more structurally complex liveness-enforcing net supervisor than the plant net model. Although elementary siphon-based method can lead to a structurally simple liveness-enforcing supervisor. However, it suffers from the computational complexity and behavior permissiveness. Complete siphon enumeration-based methods are infeasible when the size of a plant net model is large. The significance of this paper can be summarized as follows. First, the MIP-based deadlock detection method is used to derive siphons that need to be controlled. That is to say, we do not need to know all siphons beforehand. The complete siphon enumeration is avoided, leading to better computational efficiency. Second, the two-stage iteration method can ensure the number of monitors is, to some extent, minimized. Finally, a case study shows that our method is more permissive with fewer additional monitors and arcs.

However, we notice that, for the FMS example in Fig. 3, the minimal siphon derived by MIP-detection method at each iteration is not unique. For instance, at second iteration, we obtain a maximal deadly siphon $S^2 = \{p_2, p_3, p_4, p_7 - p_{10}, p_{12}, p_{14} - p_{21}, p_{24} - p_{26}\}$, and two minimal siphons can be derived from S^2, that is, $S_2 = \{p_4, p_8, p_{10}, p_{14}, p_{24}, p_{25}, p_{26}\}$ and $S'_2 = \{p_3, p_7, p_8, p_{10}, p_{14}, p_{24}, p_{26}\}$. If S_2 is chosen to be controlled, three iterations are needed in the *siphons control* stage

Table 3 The performance of supervisors due to different deadlock prevention policies

Parameter	The method in Li and Zhao (2008)	The proposed method
Permissive behavior	53,778	123,060
Monitors/arcs	5/25	3/18

totally. On the contrary, if S_2' is chosen, the different results can be obtained. The detail iterations are shown as follows:

1. $S^1 = \{p_1, p_2, p_4 - p_{10}, p_{13} - p_{21}, p_{25}, p_{26}\}$,
   $S_1 = \{p_4, p_8, p_{10}, p_{13}, p_{25}, p_{26}\}$;
   $Th_{S_1} = p_3 + p_7 + p_{12}$;
   $M_0(V_{S1}) = M_0(p_{25}) + M_0(p_{26}) - 1 = 5$;
   s.t. 1: $M(p_3) + M(p_7) + M(p_{12}) \leq 5$,
2. $G^{\mathrm{MIP}} = 8$;
   $S^2 = \{p_2, p_3, p_4, p_7 - p_{10}, p_{12}, p_{14} - p_{21}, p_{24} - p_{26}\}$;
   $S_2' = \{p_3, p_7, p_8, p_{10}, p_{14}, p_{24}, p_{26}\}$;
   $Th_{S_2'} = p_2 + p_6 + p_{13}$;
   $M_0(V_{S2}') = M_0(p_{24}) + M_0(p_{26}) - 1 = 4$;
   s.t. 2: $M(p_2) + M(p_6) + M(p_{13}) \leq 4$
3. $G^{\mathrm{MIP}} = 8$;
   $S^3 = \{p_1 - p_5, p_8, p_9, p_{10}, p_{14} - p_{21}, p_{24} - p_{26}\}$;
   $S_3 = \{p_4, p_8, p_{10}, p_{14}, p_{24}, p_{25}, p_{26}\}$;
   $Th_{S_3} = p_2 + p_3 + p_6 + p_7 + p_{12} + 2p_{13}$;
   $M_0(V_{S3}) = M_0(p_{24}) + M_0(p_{25}) + M_0(p_{26}) - 1 = 7$;
   s.t. 3: $M(p_2) + M(p_3) + M(p_6) + M(p_7) + M(p_{12}) + 2M(p_{13}) \leq 7$
4. $G^{\mathrm{MIP}} = 13$
   $S^4 = \{p_2 - p_4, p_6, p_7, p_{12}, p_{15} - p_{17}, p_{19}, p_{21} - p_{24}\}$;
   $S_4 = \{p_2, p_6, p_{15}, p_{23}, p_{24}\}$;
   $Th_{S_4} = p_1 + p_{14}$;
   $M_0(V_{S4}) = M_0(p_{23}) + M_0(p_{24}) - 1 = 3$;
   s.t. 4: $M(p_1) + M(p_{14}) \leq 3$

After four iterations in the *siphons control* stage, four constraints s.t.1–s.t.4 are added to the original linear programming problem $G^{\mathrm{MIP}} = \min \sum_{i=1}^{27} v_p$, which gives $G^{\mathrm{MIP}} = 27$. That is to say, four control places V_{S1}, V_{S2}, V_{S3} and V_{S4} with the associated arcs are added to the net such that no deadly siphon in $(N_{\mu 0}, M_{\mu 0})$. After the *siphons control* stage, we apply the MIP to the resultant net $(N_{\mu 1}, M_{\mu 1})$, the new linear programming problem is: $G^{\mathrm{MIP1}} = \min \sum_{i=1}^{31} v_p$=31, which implies that there is no deadly marked siphon in the augmented net $(N_{\mu 1}, M_{\mu 1})$. After the *Control-induced siphons control* stage, the net is live, the corresponding controlled system has 123,060 reachable states. It is worthy to note that the first stage adds monitors to the plant model such that all siphons in the plant are controlled. In fact, this stage is optimal or maximal permissive in the sense that no good states are removed due to the addition of monitors. The second stage aims at making the newly generated siphons controlled. To accelerate the convergence rate, the output arcs of the monitors added in this stage point to only the source transitions of the plant model. Consequently, permissive behavior of the resultant supervisor depends on the siphons controlled in the second stage. That is to say, selecting different siphons in the second stage to control may lead to the supervisors with different permissive behavior.

5 Concluding remarks

Existing monitor-based deadlock prevention policies for a generalized Petri net model does not provide a controlled system as permissive as the case in an ordinary net. Moreover, it usually leads to a more computational complex liveness-enforcing Petri net supervisor due to the complete siphon enumeration is necessary and the number of siphons grows fast with respect to the structural size of a plant model.

The proposed deadlock control policy focuses on improving the computational efficiency and behavior permissive for a class of generalized Petri nets, namely G-systems. To achieve this, an MIP-based detection method for generalized Petri nets without complete siphon enumeration is firstly utilized. Our deadlock prevention policy based on an iterative approach for G-systems consists of two main stages: *siphons control* and *control-induced siphons control*. The *siphons control* stage aims to add control places to for each insufficiently marked siphon in the original net. The *control-induced siphons control* stage adds control places to the augmented net with its output arcs connecting to the source transitions, which assures that there are no new insufficient siphons generated in the result net system.

One of our future work is utilizing the concept of max′-controlled or max″-controlled siphons (Chao 2007; Liu and Li 2010; Liu et al. 2010b) to derive siphon controllability condition such that a more permissive behavior supervisor can be obtained. Moreover, motivated by the discussion at the end of Section 4, a natural research direction is to improve the proposed deadlock prevention policy in this work such that the liveness-enforcing Petri net supervisors with a smaller number of monitors and optimal permissive behavior after *siphons control* stage.

Acknowledgements The authors would like to thank the National Nature Science Foundation of China under Grant No. 61104110, and the High-level Talents Research Foundation of Shihezi University under Grant No. RCZX200942.

Appendix: Basics of 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 the set of places and T is 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 flow relation of the net, represented by arcs with arrows from places to transitions or from transitions to places. $W : F \to \mathbf{N}^+$ is a mapping that assigns a weight to any arc, where $\mathbf{N}^+=\{1, 2, \cdots\}$. A net is self-loop free if $\nexists x, y \in P \cup T$, $f(x, y) \in F \wedge f(y, x) \in F$. A self-loop free net $N = (P, T, F, W)$ can be alternatively represented by its incidence matrix $[N]$, where $[N]$ is a $|P| \times |T|$ integer matrix and $[N](p, t)=W(t, p) - W(p, t)$.

A marking M of N is a mapping from P to $\mathbf{IN}$, where $\mathbf{IN} = \{0, 1, 2, \cdots\}$. $M(p)$ denotes the number of tokens contained in place p, which is marked by M iff $M(p) > 0$. Let $S \subseteq P$ be a set of places. $M(S)$ denotes the sum of tokens contained in S at marking M, where $M(S) = \sum_{p \in S} M(p)$. (N, M_0) is called a net system or marked net. For economy of space, we use $\sum_{p \in P} M(p)p$ to denote vector M.

Let $x \in P \cup T$ be a node of net $N = (P, T, F, W)$. The preset of x is defined as ${}^\bullet x = \{y \in P \cup T | (y, x) \in F\}$. While the postset of x is defined as $x^\bullet = \{y \in P \cup T | (x, y) \in F\}$. This notation can be extended to a set of nodes as follows: given $X \subseteq P \cup T$, ${}^\bullet X = \cup_{x \in X} {}^\bullet x$, and $X^\bullet = \cup_{x \in X} x^\bullet$.

A transition $t \in T$ is enabled at a marking M if $\forall p \in {}^\bullet t$, $M(p) \geq W(f(p,t))$, which will be denoted as $M[t\rangle$; when fired in a usual way, it gives a new marking M' such that $\forall p \in P$, $M'(p) = M(p) - W(p,t) + W(t,p)$, which will be denoted as $M[t\rangle M'$. Marking M' is said to be reachable from M if there exists a sequence of transitions $\sigma = t_0 t_1 \cdots t_n$ and markings $M_1, M_2, \cdots$, and M_n such that $M[t_0\rangle M_1[t_1\rangle$ $M_2 \cdots M_n[t_n\rangle M'$ holds. The set of markings reachable from M in N is denoted as $R(N, M)$.

A transition $t \in T$ is live under M_0 if $\forall M \in R(N, M_0)$, $\exists M' \in R(N, M)$, $M'[t\rangle$. N is dead under M_0 if $\nexists t \in T$, $M_0[t\rangle$ holds. (N, M_0) is deadlock-free if $\forall M \in R(N, M_0)$, $\exists t \in T$, $M[t\rangle$ holds. (N, M_0) is quasi-live if $\forall t \in T$, $\exists M \in R(N, M_0)$, $M[t\rangle$ holds. (N, M_0) is live if $\forall t \in T$, t is live under M_0. (N, M_0) is bounded if $\exists k \in \mathbf{IN}$, $\forall M \in R(N, M_0)$, $\forall p \in P$, $M(p) \leq k$ holds. (N, M_0) is said to be reversible, if for each marking $M \in R(G, M_0)$, M_0 is reachable from M. A marking M' is said to be a home state, if for each marking $M \in R(G, M_0)$, M' is reachable from M. Reversibility is a special case of the home state property, i.e. if the home state $M' = M_0$, then the net is reversible.

A P-vector is a column vector $I: P \rightarrow Z$ indexed by P and a T-vector is a column vector $J: T \rightarrow Z$ indexed by T, where Z is the set of integers. A $P(T)$-vector $I(J)$ is denoted by $\sum_{p \in P} I(p)p$ $(\sum_{t \in T} J(t)t)$ for economy of space. We denote column vectors where every entry equals 0(1) by $\mathbf{0}(\mathbf{1})$. I^T and $[N]^T$ are the transposed versions of a vector I and a matrix $[N]$, respectively. P-vector I is a P-invariant (place invariant) iff $I \neq \mathbf{0}$ and $I^T[N] = \mathbf{0}^T$. P-invariant I is said to be a P-semiflow if no element of I is negative. $||I|| = \{p \in P \mid I(p) \neq 0\}$ is called the support of I. $||I||^+ = \{p \mid I(p) > 0\}$ denotes the positive support of P-invariant I, while $||I||^- = \{p \mid I(p) < 0\}$ denotes the negative support of I. An invariant is called minimal when its support is not a strict superset of the support of any other, and the greatest common divisor of its elements is one. If I is a P-invariant of (N, M_0) then $\forall M \in R(N, M_0)$, $I^T M = I^T M_0$.

Let P be set of places of net N, I be a P-vector, and $S \subseteq P$ be a subset of places of N. $I \backslash S$ is defined to be $\sum_{p \in P \backslash S} I(p)p$. For example, $I = 2p_1 + 3p_2 + p_3 + 4p_4$ is a P-vector and $S = \{p_1, p_4\}$ in some net. Then we have $I \backslash S = 3p_2 + p_3$. $I \cap S \neq \emptyset$ means that $\exists p \in S$, $I(p) \neq 0$.

Let S be a non-empty subset of places. $S \subseteq P$ is a siphon (trap) if ${}^\bullet S \subseteq S^\bullet$ ($S^\bullet \subseteq {}^\bullet S$). Siphon S is said to be minimal if it contains no other siphons as its proper subset. A minimal siphon S is strict if it does not contain a trap.

The following notations of generalized Petri nets are from Barkaoui and Pradat-Peyre (1996).

Definition 13 Let (N, M_0) be a marked net and S be a siphon of N. S is said to be max-marked at a marking M if $\exists p \in S$ such that $M(p) \geq \max_{p^\bullet}$, where $\max_{p^\bullet} = \max\{W(p,t) \mid t \in p^\bullet\}$. S is said to be max-controlled if S is max-marked at any reachable marking.

Definition 14 A net (N, M_0) is said to satisfy the max cs-property (controlled-siphon property) if each minimal siphon of N is max-controlled.

The cs-property is an important concept in net theory on which our approaches rely. A siphon satisfying the cs-property can be always marked sufficiently to allow firing a transition once at least. In order to check and use the cs-property, Barkaoui and Pradat-Peyre (1996) proposed the conditions to determine whether a given siphon is max-controlled and establish the relationship of the cs-property and liveness property.

References

Barkaoui K, Abdallah IB (1994) An efficient deadlock avoidance control policy in FMS using structural analysis of Petri nets. In: Proc. int. conf. syst., man, cybern., vol 1, pp 525–530. San Antonio, TX

Barkaoui K, Pradat-Peyre JF (1996) On liveness and controlled siphons in Petri nets. In: Proc. 17th int. conf. application and theory of Petri nets, Osaka, Japan. LNCS, vol 1091. Springer, New York, pp 57–72

Barkaoui K, Chaoui A, Zouari B (1997) Supervisory control of discrete event systems based on structure theory of Petri nets. In: Proc. IEEE int. conf. syst., man, cybern., Orlando, Florida, USA, pp 3750–3755

Chao DY (2007) Max'-controlled siphons for liveness of S3PGR2. IET Control Theory Appl 1(4):933–936

Chao DY (2008) Incremental approach to computation of elementary siphons for arbitrary S^3PR. IET Control Theory Appl 2(2):168–179

Chu F, Xie XL (1997) Deadlock analysis of Petri nets using siphons and mathematical programming. IEEE Trans Robot Autom 13:793–804

Ezpeleta J, Colom JM, Martinez J (1995) A Petri net based deadlock prevention policy for flexible manufacturing systems. IEEE Trans Robot Autom 11:173–184

Ezpeleta J, Recalde L (2004) A deadlock avoidance approach for non-sequential resource allocation systems. IEEE Trans Syst Man Cybern A 34:93–101

Fanti MP, Zhou MC (2004) Deadlock control methods in automated manufacturing systems. IEEE Trans Syst Man Cybern A 34:5–22

Hsien FS, Chang SC (1994) Dispatching-driven deadlock avoidance controller synthesis for flexible manufacturing systems. IEEE Trans Robot Autom 10:196–209

Huang YS, Jeng MD, Xie XL, Chung SL (2001) Deadlock prevention policy based on Petri nets and siphons. Int J Prod Res 39(2):283–305

Iordache MV, Antsaklis PJ (2006) Decentralized supervision of Petri nets. IEEE Trans Automat Contr 51(2):376–381

Jeng MD, Xie XL, Chung SL (2004) ERCN* merged nets for modeling degraded behavior and parallel processes in semiconductor manufacturing systems. IEEE Trans Syst Man Cybern A 34:102–112

Li ZW, Hu HS (2007) On systematic methods to remove redundant monitors from liveness-enforcing net supervisors. Comput Ind Eng 56(1):53–62

Li ZW, Zhou MC (2004a) Elementary siphons of Petri nets and their application to deadlock prevention in flexible manufacturing systems. IEEE Trans Syst Man Cybern A 34:38–51

Li ZW, Zhou MC (2006a) Clarifications on the definitions of elementary siphons of Petri nets. IEEE Trans Syst Man Cybern A 36(6):1227–1229

Li ZW, Zhou MC (2006b) Two-stage method for synthesizing liveness-enforcing supervisors for flexible manufacturing systems using Petri nets. IEEE Trans Ind Inform 2(4):313–325

Li ZW, Zhao M (2008) On controllability of dependent siphons for deadlock prevention in generalized Petri nets. IEEE Trans Syst Man Cybern A Syst Humans 38(2):369–384

Li ZW, Hu HS, Wang AR (2007a) Design of liveness-enforcing supervisors for flexible manufacturing systems using Petri nets. IEEE Trans Syst Man Cybern C 37(4):517–526

Li ZW, Zhang J, Zhao M (2007b) Liveness-enforcing supervisor design for a class of generalized Petri net models of flexible manufacturing systems. IET Control Theory Appl 1(4):955–967

Lindo (2011) Premier optimization modeling tools. http://www.lindo.com/

Liu GY, Li ZW (2010) General mixed integer programming-based liveness test for system of sequential systems with shared resources nets. IET Control Theory Appl 4(12):2867–2878

Liu GJ, Jiang CJ, Chen LJ, Wu ZH (2009a) Two types of extended RSNBs and their application in modeling flexible manufacturing systems. Int J Adv Manuf Technol 45(5–6):573–582

Liu GJ, Jiang CJ, Wu ZH, Chen LJ (2009b) A live subclass of Petri nets and their application in modeling flexible manufacturing systems. Int J Adv Manuf Technol 41(1–2):66–74

Liu GJ, Jiang CJ, Zhou MC (2010a) Two simple deadlock prevention policies for S3PR based on key-resource/operation- place pairs. IEEE Trans Autom Sci Eng 7(4):945–957

Liu GY, Li ZW, Zhong CF (2010b) New controllability condition for siphons in a class of generalised Petri nets. IET Control Theory Appl 4(5):854–864

Moody JO, Antsaklis PJ (2001) Petri net supervisors for DES with uncontrollable and unobservable transitions. IEEE Trans Automat Contr 45(3):462–476

Ohta A, Tsuji K (2003) Insufficiently marked siphon of Petri nets-extension of token-free siphon. In: Proceedings of the 2003 international symposium on circuits and systems, vol 3, pp 244–247

Park J, Reveliotis SA (2001) Deadlock avoidance in sequential resource allocation systems with multiple resource acquisitions and flexible routings. IEEE Trans Automat Contr 46:1572–1583

Reveliotis SA (2003) On the siphon-based characterization of liveness in sequential resource allocation systems. In: van der Aalst WMP, Best E (eds) Proc. int. conf. application and theory of Petri nets. LNCS, vol 2679, pp 241–255

Tricas F (2003) Deadlock analysis, prevention and avoidance in sequential resource allocation systems. Ph.D. thesis, Departamento de Informatica e Ingenieria de Sistemas. Universidad de Zaragoza

Uzam M (2002) An optimal deadlock prevention policy for flexible manufacturing systems using Petri net models with resources and the theory of regions. Int J Adv Manuf Technol 19:192–208

Wei N, Li ZW (2008) On the suboptimal liveness-enforcing supervisors based on Petri net structural analysis and the theory of regions. Int J Adv Manuf Technol 38:195–204

Xing KY, Zhou MC, Liu HX, Tian F (2009) Optimal Petri-net-based polynoical-complexity deadlock-avoidance polices for automated manufacturing systems. IEEE Trans Syst Man Cybern A 39(1):188–199

Zouari B, Barkaoui K (2003) Parameterized supervisor synthesis for a modular class of discrete event systems. In: Proc. IEEE int. conf. syst. man. cybern., Washington, DC, USA., pp 1874–1879

Mi Zhao received her B.S., M.S., and Ph.D. degrees all from Xidian University, Xi'an, China, in 2002, 2006, and 2009, respectively.

She joined Shihezi University, Xinjiang, in 2009, where she is currently associate professor of Machinery and Electricity College. She served as a member of the Program Committee of the 2010 IEEE International Conference on Mechatronics and Automation, Xi'an, China, August 5–7, 2010. Her main research interests include Petri net theory and application, supervisory control of discrete event systems.

Yifan Hou received her B.S. from Daqing Petroleum Institute, HeiLongJiang, China, in 2005. Currently, she is a Ph.D. candidate of mechanical engineering in School of Electro-Mechanical Engineering, Xidian University. Her main research interests include Petri net theory and application, supervisory control of Flexible Manufacturing Systems.