第 28 卷第 6 期
2011 年 6 月

控 制 理 论 与 应 用
Control Theory & Applications

Vol. 28 No. 6
Jun. 2011

文章编号: 1000−8152(2011)06−0771−10

应用必需信标的Petri网死锁预防策略

李绍勇[1,2], 王安荣[1]

(1. 西安电子科技大学 机电工程学院, 陕西 西安 710071; 2. 兰州理工大学 土木工程学院, 甘肃 兰州 730050)

摘要: 本文提出了表征一个Petri网子类, 即S^4R网(system of sequential systems with shared resources)中死锁问题的必需信标的概念和一种将混合整数规划算法与必需信标控制相结合的死锁预防策略. 在该策略的迭代过程中, 混合整数规划算法发现被控的Petri网中是否存在最大的死标识信标, 若存在, 则通过库所分类和迭代式的信标提取,得到必需信标, 添加相应的控制库所, 满足必需信标的最大可控性, 从而实现被控的Petri网活性的目的. 理论分析和算例验证表明了该策略的正确性和有效性.

关键词: Petri网; 死锁预防; 混合整数规划; 必需信标

中图分类号: TP273　　　**文献标识码**: A

A deadlock prevention policy in Petri nets using necessary siphons

LI Shao-yong[1,2], WANG An-rong[1]

(1. School of Electro-Mechanical Engineering, Xidian University, Xi'an Shaanxi 710071, China;
2. School of Civil Engineering, Lanzhou University of Technology, Lanzhou Gansu 730050, China)

Abstract: This paper puts forward the concept of necessary siphons that characterizes the deadlock problem in a subclass of Petri nets S^4R(system of sequential systems with shared resources) and a deadlock prevention policy combining the mixed integer programming(MIP) and the control of necessary siphons. In the iteration of the proposed policy, the MIP-based deadlock-detection method explores whether a maximal deadly marked siphon exists in the controlled Petri net; if it is, this siphon is identified by the method of classification of places, and extracted as a necessary siphon to be controlled. A proper control place(CP) is applied to the necessary siphon to make it max-controlled, thus, ensuring the liveness of the controlled Petri nets. Theoretical analysis and an experimental example show the efficacy of the proposed policy.

Key words: Petri nets; deadlock prevention; mixed integer programming(MIP); necessary siphons

1 引言(Introduction)

在对离散事件动态系统(discrete event dynamic system, DEDS)的建模、分析、性能评价及其控制设计中, Petri网[1]作为一种数学的方法, 得到了广泛的应用. 柔性制造系统(flexible manufacturing system, FMS)作为一类典型的离散事件动态系统一直是Petri网研究的重要对象和应用领域. 由于竞争有限的共享资源(如机器人、机床、夹具和传送带等), 死锁(deadlock)经常发生[2~6], 导致部分或整个系统的运行停顿, 甚至可能造成灾难性的后果和重大的经济损失[4,6]. 人们基于Petri网已经研究了许多方法来解决死锁问题, 一般有死锁的检测与恢复方法(deadlock detection and recovery)、死锁避免方法[2,6,7](deadlock avoidance)、死锁预防方法[2,3,5,6,8](deadlock prevention).

Petri网在死锁预防方面主要采用网结构分析和可达图分析[9,10]. 网结构的分析方法是基于FMS中的死锁问题与对应建模Petri网中信标的强相关性[3,5,10], 信标是Petri网中的一种特殊结构, 一旦在某个标识下被清空, 则永久地在这个标识的所有后继标识下保持清空状态, 从而导致死锁的发生. Ezpeleta等[3]采用完全信标枚举法, 提出的死锁预防策略首先求取所有未被标识的严格极小信标(strict minimal siphon, SMS), 并且添加相应的控制库所防止其被清空. 由于网模型中的SMS个数与网的规模在理论上是指数递增关系, 添加了非常多的控制库所和连接弧, 得到一个结构复杂的活的Petri网受控系统. Li和Zhou[5,10]提出了基本和从属信标的概念, 通过显式控制基本信标, 使得从属信标隐式可控. 因为网模型中的基本信标个数远远小于SMS个数, 获

收稿日期: 2010−03−18; 收修改稿日期: 2010−06−25.
基金项目: 国家自然科学基金资助项目(60773001, 61074035, 61064003); 教育部高等学校博士点基金资助项目(20090203110009); 国家高科技发展规划“863”计划资助项目(2008AA04Z109); 西安电子科技大学基本科研业务费资助项目; Alexander von Humboldt Foundation研究基金资助项目.

取了结构较为简单的活性Petri网系统. 但是, 该策略需要完全的信标枚举和适当选择基本信标. 可达图的分析方法是通过生成Petri网模型的全部可达状态, 从中诊断出死锁标识, 肯定导致死锁的坏标识, 危险标识和好标识, 从而确定出保证网系统最大许可行为的预防策略. 由于Petri网固有的状态爆炸特性, 使得其应用很难具有普遍性.

本文针对S^{4}R网中的死锁问题, 基于不完全信标枚举法, 提出了混合整数规划算法与必需信标控制相结合的迭代式死锁预防策略. 在迭代过程中的每一步, 混合整数规划(MIP)方法能够便捷地确定一个给定标识的S^{4}R网中是否存在最大的死标识信标, 表示为$S_{\max}$. 若存在, MIP的可行解则对应于一个$S_{\max}$, 通过库所的分类和迭代式的信标提取, 得到对应的必需信标, 添加适当的控制库所, 使得必需信标满足最大可控性[2]. 检验$G^{\mathrm{MIP}}(M_0)=|P|+|V|$(其中$G^{\mathrm{MIP}}(M_0), |P|, |V|$分别表示MIP的可行解、原网中库所数目的总和以及添加控制库所的数目)是否成立. 如果成立, 说明添加控制库所后, 被控Petri网系统是活的, 迭代过程结束; 否则, 表明被控Petri网系统还存在$S_{\max}$, 是不活的. 继续迭代提取必需信标, 添加控制库所, 使其满足最大可控性, 直至$G^{\mathrm{MIP}}(M_0)=|P|+|V|$成立, 表明无$S_{\max}$的存在, 实现被控Petri网系统活性的目的.

2 基本定义和定理(Basic definitions and theorems)

定义 1[1,9,10] 一个Petri网N是一个四元组(P, T, F, W), P和T分别称为库所和变迁的集合, 满足$P\neq\varnothing, T\neq\varnothing, P\cap T=\varnothing$. $F\subseteq(P\times T)\cup(T\times P)$称为流关系. $W:(P\times T)\cup(T\times P)\rightarrow\mathbb{N}$是一个映射, $\mathbb{N}\in\{0,1,2,\cdots\}$, 该映射为每一条弧分配一个权值, 即, 若$f\in F$, 则$W(f)>0$; 若$f\notin F$, 则$W(f)=0$. W称为Petri网N的权函数.

定义 2[1,9,10] 若$\forall f\in F, W(f)=1$, 则Petri网$N=(P,T,F,W)$称为普通网, 可记做$N=(P,T,F)$; 若$\forall f\in F, W(f)\geqslant 1$, 则$N$称为一般网.

定义 3[1,9,10] Petri网$N=(P,T,F,W)$的标识M是一个从P到$\mathbb{N}$的映射. (N,M_0)称为标识网, M_0称为N的初始标识. 从M_0可达的所有标识的集合称为(N,M_0)的可达集, 记为$R(N,M_0)$.

定义 4[1,9,10] 令$N=(P,T,F,W)$是一个Petri网, N的P–向量I是映射I: $P\rightarrow\mathbb{Z}$, P–向量是以P为序标的列向量, $\mathbb{Z}$是整数的集合. 若P–向量I满足$I\neq\mathbf{0}$, 且$I^{\mathrm{T}}[N]=\mathbf{0}^{\mathrm{T}}$, 则称$I$是Petri网$N$的P–不变式, 其中$\mathbf{0}$是一个所有元素都等于0的列向量. $\|I\|=\{p\in P\mid I(p)\neq 0\}$称为$I$的支撑. $\|I\|^{+}=\{p\mid I(p)>0\}$称为$I$的正支撑, $\|I\|^{-}=\{p\mid I(p)<0\}$称为$I$的负支撑.

由结构导出的P–不变式I满足托肯守恒律. 也就是说, 在每一个可达标识下, $\|I\|$中托肯的加权和保持为常数, 即$\forall M\in R(N,M_0), I^{\mathrm{T}}M=I^{\mathrm{T}}M_0$.

定义 5[1,9,10] 令$N=(P,T,F,W)$是一个Petri网. 若非空集合$S\subseteq P$满足${}^{\bullet}S\subseteq S^{\bullet}$, 则称$S$为信标. 若信标$S$不包含其他任何信标作为它的真子集, 则称其为极小信标. 若信标S不包含任何P–不变式的支撑, 则称其为严格信标. 称一个既是极小的又是严格的信标为严格极小信标.

定义 6[1,9,10] 令$N=(P,T,F,W)$是一个Petri网. 称$N_{\mathrm{X}}=(P_{\mathrm{X}},T_{\mathrm{X}},F_{\mathrm{X}},W_{\mathrm{X}})$为$P_{\mathrm{X}}\cup T_{\mathrm{X}}$导出的子网, 其中:

$$P_{\mathrm{X}}\subseteq P,\ T_{\mathrm{X}}\subseteq T,$$
$$F_{\mathrm{X}}=F\cap[(P_{\mathrm{X}}\times T_{\mathrm{X}})\cup(T_{\mathrm{X}}\times P_{\mathrm{X}})],$$
$$\forall f\in F_{\mathrm{X}},\ W_{\mathrm{X}}(f)=W(f).$$

定义 7[2] 令S是网系统(N,M_0)的信标. 若$\exists p\in S$使得$M(p)\geqslant\max_{p^{\bullet}}$, 其中

$$\max_{p^{\bullet}}=\max\{W(p,t)|t\in p^{\bullet}\},$$

则称S在标识M下是最大标记的(max-marked). 若信标S在每个可达标识下均为最大标记的, 则称该信标是最大可控的(max-controlled). 若N的每一个严格极小信标都是最大可控的, 则称Petri网(N,M_0)满足最大可控信标性质(max-cs-property).

定义 8[10] 初始标识为M_0的S^{4}R(system of sequential systems with shared resources)网$N=(P,T,F,W)$满足

1) $P=P_{\mathrm{A}}\cup P^0\cup P_{\mathrm{R}}$, 其中$P_{\mathrm{A}}=\bigcup\limits_{j=1}^{n}P_{A_j}$称为操作库所集合, 且$\forall i,j\in\mathbb{N}_n, i\neq j, P_{A_i}\cap P_{A_j}=\varnothing$; $P^0=\bigcup\limits_{i=1}^{n}\{p_i^0\}$称为闲置库所集合, 且$P^0\cap P_{\mathrm{A}}=\varnothing$; $P_{\mathrm{R}}=\{r_1,r_2,\cdots,r_m\}$称为资源库所集合, 且$(P_{\mathrm{A}}\cup P^0)\cap P_{\mathrm{R}}=\varnothing$.

2) $T=\bigcup\limits_{j=1}^{n}T_j$, 且$\forall i,j\in\mathbb{N}_n, i\neq j, T_i\cap T_j=\varnothing$.

3) $W=W_{\mathrm{A}}\cup W_{\mathrm{R}}$, 其中:

$$W_{\mathrm{R}}:(P_{\mathrm{R}}\times T)\cup(T\times P_{\mathrm{R}})\rightarrow\mathbb{N},$$
$$W_{\mathrm{A}}:((P_{\mathrm{A}}\cup P^0)\times T)\cup(T\times(P_{\mathrm{A}}\cup P^0))\rightarrow\{0,1\},$$

且$\forall i,j\in\mathbb{N}_n, i\neq j$, $((P_{A_j}\cup\{p_j^0\})\times T_i)\cup(T_i\times(P_{A_j}\cup\{p_j^0\}))\rightarrow\{0\}$.

4) $\forall j\in\mathbb{N}_n$, 由$P_{A_j}\cup\{p_j^0\}\cup T_j$导出的子网$N_j$是强连通的状态机且每个回路包含$p_j^0$.

5) $\forall r\in P_{\mathrm{R}}$, 存在唯一的P–不变式$I_{\mathrm{r}}$使得

$$\|I_{\rm r}\| \cap P_{\rm R} = \{r\},\ \|I_{\rm r}\| \cap P^0 = \varnothing,$$

$$\|I_{\rm r}\| \cap P_{\rm A} \neq \varnothing,\ I_{\rm r}(r) = 1.$$

进一步, $P_{\rm A} = (\bigcup\limits_{r \in P_{\rm R}} \|I_{\rm r}\|) \setminus P_{\rm R}$.

6) N是强连通的纯网.

7) $\forall p \in P_{\rm A}, M_0(p) = 0$; $\forall r \in P_{\rm R}$; $M_0(r) \geqslant \max\limits_{p \in \|I_{\rm r}\|} I_{\rm r}(p)$; $\forall p^0 \in P^0, M_0(p^0) \geqslant 1$.

定义 9[10]　令r是$\rm S^4R$中的一个资源库所, S是一个SMS, 其中: $S = S^{\rm R} \cup S^{\rm A}, S^{\rm R} = S \cap P_{\rm R}, S^{\rm A} = S \cap P_{\rm A}$. 使用$r$的操作库所集合$H(r)$称为$r$的持有者, 定义为$I_{\rm r}$和$r$之差, 即$H(r) = I_{\rm r} - r$, 这里$H(r)$, $I_{\rm r}$, r均为多集形式. 多集

$$\mathrm{Th}(S) = \sum_{r \in S^{\rm R}} H(r) - \sum_{r \in S^{\rm R}, p \in S^{\rm A}} I_{\rm r}(p) \cdot p$$

称为S的补集. 同样也可以用$\sum\limits_{p \in \|\mathrm{Th}(S)\|} h_{\rm s}(p)p$来表示$\mathrm{Th}(S)$. $h_{\rm s}(p)$表示$\mathrm{Th}(S)$中的库所p的托肯数增加1时, 信标S失去$h_{\rm s}(p)$个托肯.

由定义9可知, 信标的补集$\mathrm{Th}(S)$也是由操作库所组成的, 它具有明确的物理含义. $\mathrm{Th}(S)$中的操作库所与信标S中的操作库所竞争信标中的资源, 当资源库所中的托肯全部流入信标补集的库所中时, 该信标即被清空, 从而导致产生死变迁.

定理 1[2]　令(N, M_0)是一个$\rm S^4R$网, N在M_0是活的当且仅当它满足最大可控信标性质(max-controlled-siphon-property).

3　MIP方法(MIP method)

文献[7]中的MIP方法可以便捷地确定一个一般Petri网中的最大死标识信标. 令S是一个可达标识M下的最大死标识信标, 即$\forall p \notin S, M(p) > 0$. 最大死标识信标$S$记作$S_{\max}$. 文献[7]确定$S$的算法如下: 首先移去所有含有托肯的库所, 然后移去没有输入库所的变迁, 以及这些变迁的输出库所. 重复以上步骤, 直到没有节点可以被移去, 则留下的库所构成M下最大死标识信标. Park和Reveliotis[7]指出该算法对应着一个MIP问题, 所以他们引入3个指示符, 定义为

$$\nu_{\rm p} = 1\{p \notin S\},\ z_{\rm t} = 1\{t \notin S^{\bullet}\},$$

$$f_{\rm pt} = 1\{M_{\rm p} \geqslant W(p,t) \vee \nu_{\rm p} = 1,\ \forall W(p,t) > 0\}.$$

显然, 在文献[7]确定S的算法中, 任何$\nu_{\rm p} = 1$的库所p和$z_{\rm t} = 1$的变迁t将被移去. 因为S是信标,

$$\forall t \in p^{\bullet},\ \nu_{\rm p} = 0 (p \in S) \Longrightarrow z_{\rm t} = 0 (t \in S^{\bullet});$$

$$\forall p \in t^{\bullet},\ z_{\rm t} = 1 (t \notin S^{\bullet}) \Longrightarrow \nu_{\rm p} = 1 (p \notin S);$$

$$f_{\rm pt} = 1 \Longleftrightarrow M_{\rm p} \geqslant W(p,t) \vee \nu_{\rm p} = 1,\ \forall W(p,t) > 0.$$

所以

$$S = \{p \in P | \nu_{\rm p} = 0\}. \tag{1}$$

这里$\nu_{\rm p}$, $p \in P$可以通过求解下列问题获得:

$$G^{\rm MIP}(M_0) = \min \sum_{p \in P} \nu_{\rm p}, \tag{2}$$

s.t.

$$f_{\rm pt} \geqslant \frac{M(p) - W(p,t) + 1}{SB(p)},\ \forall W(p,t) > 0, \tag{3}$$

$$f_{\rm pt} \geqslant \nu_{\rm p},\ \forall W(p,t) > 0, \tag{4}$$

$$z_{\rm t} \geqslant \sum_{p \in {}^{\bullet}t} f_{\rm pt} - |{}^{\bullet}t| + 1,\ \forall t \in T, \tag{5}$$

$$\nu_{\rm p} \geqslant z_{\rm t},\ \forall W(t,p) > 0, \tag{6}$$

$$\nu_{\rm p},\ z_{\rm t},\ f_{\rm pt} \in \{0,1\},\ \forall p \in P,\ \forall t \in T, \tag{7}$$

$$M = M_0 + [N]Y,\ M \geqslant 0, Y \geqslant 0. \tag{8}$$

简要地说, 一般Petri网存在最大的死标识信标$S_{\max}$的充要条件是$G^{\rm MIP}(M_0) < |P|$.

4　库所分类和必需信标(Classification of places and necessary siphon)

4.1　库所分类(Classification of places)

为了表述库所分类的理念, 考虑图1所示的$\rm S^4R$网[11], 其中:

$$P^0 = \{p_{15}, p_{16}\},\ P_{\rm R} = \{p_{12}, p_{13}, p_{14}\}$$

$$P_{\rm A} = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}, p_{11}\}.$$

显然, $|P| = |P^0| + |P_{\rm A}| + |P_{\rm R}| = 16$.

$$S_{\max} = \{p_1, p_2, p_4, p_5, p_8, p_9, p_{10}, p_{11}, p_{12}, p_{13}, p_{14}\}$$

能够便捷地由文献[7]中的MIP方法得到. 由$(S_{\max} \cup {}^{\bullet}S_{\max} \cup S_{\max}{}^{\bullet})$导出的子网$N_{S_{\max}}$如图2所示.

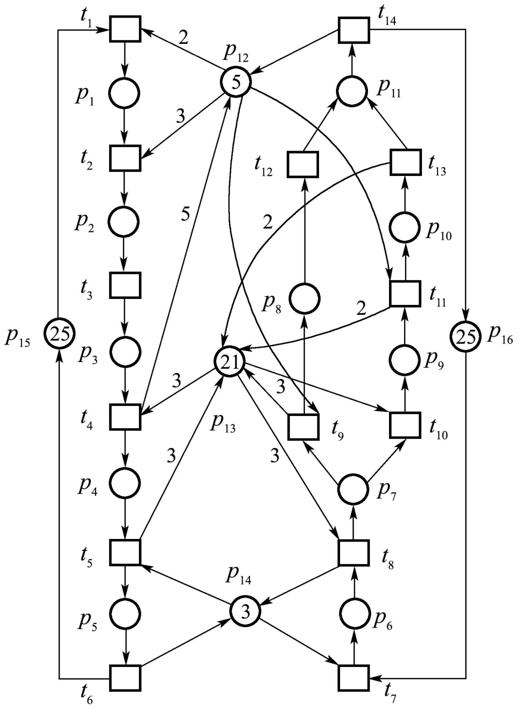

图 1　一个标识的$\rm S^4R$网

Fig. 1　A marked $\rm S^4R$ with deadlocks

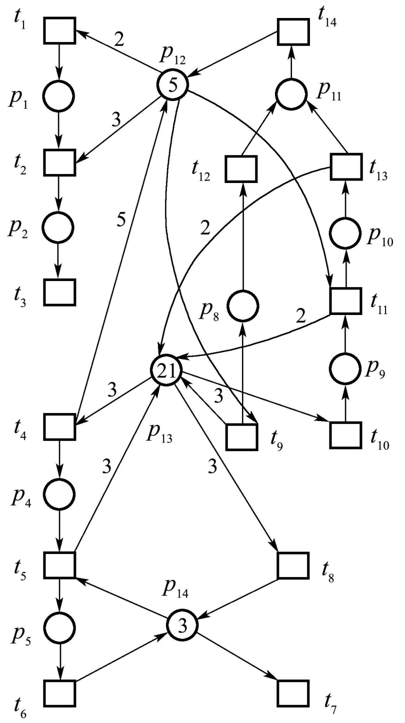

图 2 $(S_{\max}\cup {}^\bullet S_{\max}\cup S_{\max}{}^\bullet)$导出的子网

Fig. 2 The subnet generated by $(S_{\max}\cup {}^\bullet S_{\max}\cup S_{\max}{}^\bullet)$

图3所示的是能够实现$S_{\max}$中库所分类的算法1的流程图, 其中m, p_i, i和$t_{\rm s}$分别表示$N_{S_{\max}}$中库所的总数、$S_{\max}$中的测试库所、测试次数和从$N_{S_{\max}}$中移除p_i及其连接弧后所生成的$N'_{S_{\max}}$中的源变迁.

首先, 选取库所$p_i(i=1,2,\cdots,m)$并且测试其是否属于P'. 该过程迭代进行, 直至$i>m$成立, 得到P'.

然后, 选取库所$p_i(i=m,m-1,\cdots,1)$并且测试其是否属于P''. 迭代进行该过程, 直至$i=0$成立, 得到P''. 最后通过P'和P''的集合运算, 得到3类库所的集合$P'_{\rm d}$, $P'_{\rm s}$和$P'_{\rm u}$, 完成了对$S_{\max}$中库所的分类, 为下一步获取必需信标做好准备. 例如, 算法1应用于图2所示的$N_{S_{\max}}$, 得到:

$$P'=\{p_4,p_8,p_{10},p_{11},p_{12},p_{13}\},$$
$$P''=\{p_5,p_8,p_{10},p_{11},p_{12},p_{13},p_{14}\},$$
$$P'_{\rm d}=\{p_8,p_{10},p_{11},p_{12},p_{13}\},$$
$$P'_{\rm s}=\{p_4,p_5,p_{14}\},\ P'_{\rm u}=\{p_1,p_2,p_9\}.$$

定理 2 算法1得到的P'和P''都是极小信标.

证 对于P', 令U是以$p_1\sim p_m$方式从其中依次移除库所, 都不会在$N'_{S_{\max}}$中产生源变迁$t_{\rm s}$的库所集合. 因此, 得到$P'=S_{\max}\backslash U$. 由于$S_{\max}$是信标, ${}^\bullet S_{\max}\subseteq S_{\max}{}^\bullet$是成立的. 由算法1, 可得:

$${}^\bullet(S_{\max}\backslash U)\subseteq {}^\bullet S_{\max},\ S_{\max}{}^\bullet=(S_{\max}\backslash U)^\bullet,$$

即${}^\bullet(S_{\max}\backslash U)\subseteq(S_{\max}\backslash U)^\bullet$. 所以, ${}^\bullet P'\subseteq P'^\bullet$是成立的, P'是信标.

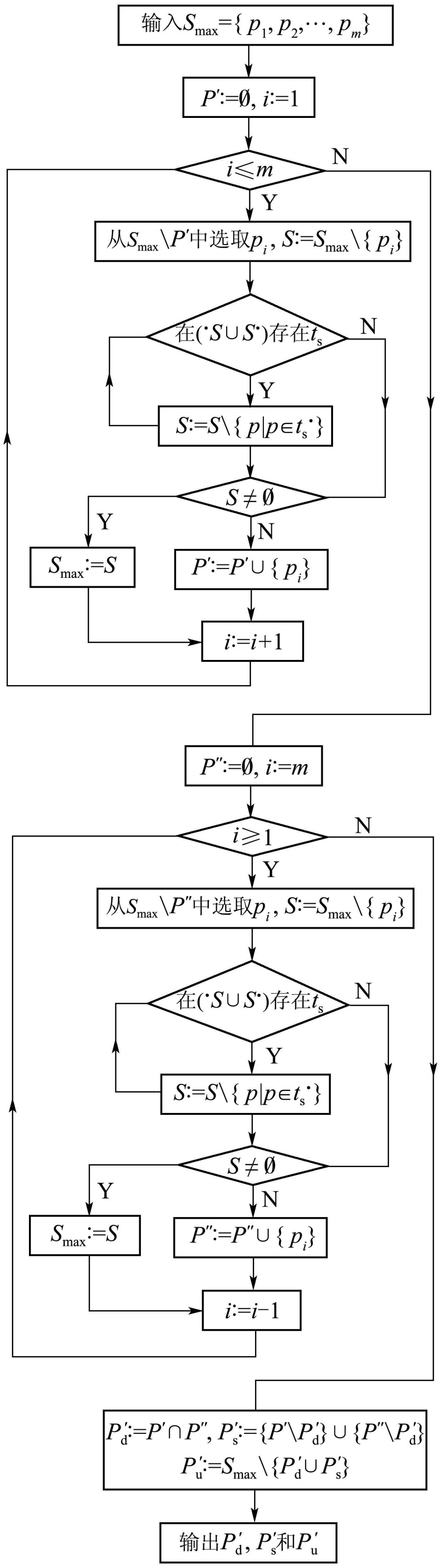

图 3 库所分类算法1的流程图

Fig. 3 Flowchart of algorithm 1

下一步证明P'是极小的. 由定义5可知, 若一个信标不包含其他的信标作为其子集, 它就是极小的. 也就是说, 若从P'中移除任何一个库所p, 使得$P'\backslash\{p\}$不再是一个信标, P'就是极小的.

由算法1, 可得$p\in P'$,$\exists t\in p^\bullet$使得$|{}^\bullet t\cap P'|=$

$|\{p\}| = 1$成立, 意味着从P'移除任何库所都会产生$N'_{\mathrm{S_{max}}}$中的源变迁t_{s}. 由反证法, 假设从P'移除库所p会产生源变迁$t_{\mathrm{s}}(t_{\mathrm{s}} \in p^{\bullet})$. 显然, $t_{\mathrm{s}} \in {}^{\bullet}(P'\backslash\{p\})$是成立的. 但是, 由算法1得到$t_{\mathrm{s}} \in (P'\backslash\{p\})^{\bullet}$. 所以, $P'\backslash\{p\}$不再是一个信标, P'是极小的. 对于P''是极小信标的证明, 类似地, 可参照P'进行. 证毕.

基于上述的库所分类算法1, 可以将$S_{\max}$中的库所分为肯定、选择和无关库所.

定义 10 令$|T'|$是$N_{\mathrm{S_{max}}}$中总的变迁数目, $\alpha(t_{\mathrm{s}})$是执行库所分类算法过程中t_{s}激发的总数目. 如果一个库所$p_i \in S_{\max}$同时满足条件: 1) $\exists t_{\mathrm{s}} \in N'_{\mathrm{S_{max}}}$使得$\alpha(t_{\mathrm{s}}) = |T'|$; 2) $p_i \in P'_{\mathrm{d}}$, 则称之为肯定库所, 记作p_{d}.

肯定库所的集合表示为P_{d}, 可以得到:

$$P_{\mathrm{d}} = P_{\mathrm{d}}^{\mathrm{R}} \cup P_{\mathrm{d}}^{\mathrm{A}},\ P_{\mathrm{d}}^{\mathrm{R}} \subset P_{\mathrm{R}},\ P_{\mathrm{d}}^{\mathrm{A}} \subset P_{\mathrm{A}}.$$

定义 11 如果库所$p_i \in S_{\max}$同时满足条件: 1) $\exists t_{\mathrm{s}} \in (S_{\max}\backslash\{p_i\})^{\bullet}$使得$\alpha(t_{\mathrm{s}}) = |T'|$; 2) $p_i \in P'_{\mathrm{s}}$, 则称之为选择库所, 记作p_{s}.

肯定库所的集合表示为P_{s}, 可以得到:

$$P_{\mathrm{s}} = P_{\mathrm{s}}^{\mathrm{R}} \cup P_{\mathrm{s}}^{\mathrm{A}},\ P_{\mathrm{s}}^{\mathrm{R}} \subset P_{\mathrm{R}},\ P_{\mathrm{s}}^{\mathrm{A}} \subset P_{\mathrm{A}}.$$

定义 12 如果库所$p_i \in S_{\max}$同时满足条件: 1) $\nexists t_{\mathrm{s}} \in N'_{\mathrm{S_{max}}}$使得$\alpha(t_{\mathrm{s}}) = 0$; 2) $p_i \in P'_{\mathrm{u}}$, 则称之为无关库所, 记作p_{u}.

无关库所的集合表示为P_{u}, 可以得到$P_{\mathrm{u}} \not\subset P'(P'')$.

基于算法1以及定义10∼12, 对图2所示的$S_{\max}$中的库所进行分析, 得到:

$$P_{\mathrm{d}} = \{p_8, p_{10}, p_{11}, p_{12}, p_{13}\},$$
$$P_{\mathrm{d}}^{\mathrm{A}} = \{p_8, p_{10}, p_{11}\},\ P_{\mathrm{d}}^{\mathrm{R}} = \{p_{12}, p_{13}\},$$
$$P_{\mathrm{s}} = \{p_4, p_5, p_{14}\},\ P_{\mathrm{s}}^{\mathrm{A}} = \{p_4, p_5\},$$
$$P_{\mathrm{s}}^{\mathrm{R}} = \{p_{14}\},\ P_{\mathrm{u}} = \{p_1, p_2, p_9\}.$$

4.2 必需信标(Necessary siphon)

如前所述, Petri网中未被标识的SMS会直接导致死锁. 如何从上述的库所分类算法得到的P'或者P''中, 确定出一个必需信标并加以控制是本文提出的死锁预防策略的关键. 4.1节中的结果表明必需信标的确定主要取决于P_{s}中的库所选择. 因此, 需要进一步分析P_{s}中的库所元素, 从而确定出必需信标.

定义 13 令SP是$N_{\mathrm{S_{max}}}$中的简单路径, x和y是SP中的两个节点. 若SP中存在一条从x到y的路径, 则称节点x先于y, 记为$x <_{\mathrm{SP}} y$. 对于$N_{\mathrm{S_{max}}}$中的节点x和y, 若$N_{\mathrm{S_{max}}}$中存在一条SP, 使得$x <_{\mathrm{SP}} y$, 则称节点x在$N_{\mathrm{S_{max}}}$中先于y, 记为$x <_{N_{\mathrm{S_{max}}}} y$.

定义 14 分别称P_{x}和P_{y}为P_{s}中的上游和下游库所集合, 若其同时满足以下3个条件: 1) $P_{\mathrm{x}} \cap P_{\mathrm{y}} = \varnothing$和$P_{\mathrm{x}} \cup P_{\mathrm{y}} = P_{\mathrm{s}}$; 2) $P_{\mathrm{x}}{}^{\bullet} \cap {}^{\bullet}P_{\mathrm{y}} \neq \varnothing$和${}^{\bullet}P_{\mathrm{x}} \cap P_{\mathrm{y}}{}^{\bullet} = \varnothing$; 3) $\forall p_{\mathrm{x}} \in P_{\mathrm{x}}$和$\forall p_{\mathrm{y}} \in P_{\mathrm{y}}, p_{\mathrm{x}} <_{N_{\mathrm{S_{max}}}} p_{\mathrm{y}}$.

由定义14, 可以得到:

$$P_{\mathrm{s}} = P_{\mathrm{x}} \cup P_{\mathrm{y}},\ P_{\mathrm{x}} = P_{\mathrm{x}}^{\mathrm{A}} \cup P_{\mathrm{x}}^{\mathrm{R}},\ P_{\mathrm{y}} = P_{\mathrm{y}}^{\mathrm{A}} \cup P_{\mathrm{y}}^{\mathrm{R}},$$
$$P_{\mathrm{x}}^{\mathrm{A}}, P_{\mathrm{y}}^{\mathrm{A}} \subset P_{\mathrm{A}},\ P_{\mathrm{x}}^{\mathrm{R}}, P_{\mathrm{y}}^{\mathrm{R}} \subset P_{\mathrm{R}}.$$

例如图2所示的$N_{\mathrm{S_{max}}}$, 可以得到:

$$P_{\mathrm{s}} = \{p_4, p_5, p_{14}\},\ P_{\mathrm{x}}^{\mathrm{R}} = \varnothing,\ P_{\mathrm{x}}^{\mathrm{A}} = \{p_4\},$$
$$P_{\mathrm{x}} = \{p_4\},\ P_{\mathrm{y}}^{\mathrm{R}} = \{p_{14}\},\ P_{\mathrm{y}}^{\mathrm{A}} = \{p_5\},$$
$$P_{\mathrm{y}} = \{p_5, p_{14}\},\ p_4 <_{N_{\mathrm{S_{max}}}} p_5,\ p_4 <_{N_{\mathrm{S_{max}}}} p_{14}.$$

因此可得$\{p_4\} <_{N_{\mathrm{S_{max}}}} \{p_5, p_{14}\}$.

从文献[12, 13]可知, 包含较多资源库所的信标是能够由那些包含较少资源库所的信标来组成的. 在本文中, 如何使得必需信标包含的资源库所数量最少是确定必需信标的关键, 这类似于文献[14]中的最少资源需求的概念. 所以, 本文提出的必需信标可以定义如下.

定义 15 $S_{\min}^{*}$被称作一个必需信标, 若

1) $S_{\min}^{*} = P_{\mathrm{d}} \cup P_{\min}$;

2) $P_{\min} = \begin{cases} P_{\mathrm{x}},\ |P_{\mathrm{x}}^{\mathrm{R}}| \leqslant |P_{\mathrm{y}}^{\mathrm{R}}|, \\ P_{\mathrm{y}},\ |P_{\mathrm{x}}^{\mathrm{R}}| > |P_{\mathrm{y}}^{\mathrm{R}}|, \end{cases}$

其中$|P_{\mathrm{x}}^{\mathrm{R}}|$和$|P_{\mathrm{y}}^{\mathrm{R}}|$分别表示$P_{\mathrm{x}}$和$P_{\mathrm{y}}$中的资源库所的数量.

例如, 对于图2所示的$N_{\mathrm{S_{max}}}$, 有:

$$P_{\mathrm{x}} = \{p_4\},\ |P_{\mathrm{x}}^{\mathrm{R}}| = |\varnothing| = 0,$$
$$P_{\mathrm{y}} = \{p_5, p_{14}\},\ |P_{\mathrm{y}}^{\mathrm{R}}| = |\{p_{14}\}| = 1,$$
$$S_{\min}^{*} = P_{\mathrm{d}} \cup P_{\min} = P_{\mathrm{d}} \cup P_{\mathrm{x}} = \{p_8, p_{10}, p_{11}, p_{12}, p_{13}\} \cup \{p_4\} = \{p_4, p_8, p_{10}, p_{11}, p_{12}, p_{13}\}.$$

5 死锁预防策略(Deadlock prevention policy)

5.1 控制库所(Control place)

确定$S_{\min}^{*}$后, 其补集$\mathrm{Th}(S_{\min}^{*})$可由定义9得到. 例如，如图2所示的$N_{\mathrm{S_{max}}}$, 在确定$S_{\min}^{*}$后, 可得:

$$\mathrm{Th}(S_{\min}^{*}) = \{2p_1, 5p_2, 5p_3, 3p_7, 4p_9\},$$
$$h_{\mathrm{S_{min}^{*}}}(p_1) = 2,\ h_{\mathrm{S_{min}^{*}}}(p_2) = h_{\mathrm{S_{min}^{*}}}(p_3) = 5,$$
$$h_{\mathrm{S_{min}^{*}}}(p_7) = 3,\ h_{\mathrm{S_{min}^{*}}}(p_9) = 4.$$

为了使得$S_{\min}^{*}$最大可控, 本文引入控制库所$V_{\mathrm{S_{min}^{*}}}$来实现需求的控制. 根据$\mathrm{Th}(S_{\min}^{*})$, 控制库所可分为两种类型: 1) 若$\forall p \in \mathrm{Th}(S_{\min}^{*})$, $h_{\mathrm{S_{min}^{*}}}(p) = 1$, 则对原网$(N_0, M_0)$添加普通的控制库所, 其中$W(p,t) = W(t,p) = 1$; 2) 若$\exists p \in \mathrm{Th}(S_{\min}^{*})$, $h_{\mathrm{S_{min}^{*}}}(p) > 1$, 则对原网$(N_0, M_0)$添加一般的控制库

所, 其中$W(p,t) \geqslant 1$和$W(t,p) \geqslant 1$. 添加控制库所还包括确定初始标识以及相应的输入、输出弧.

引理 1[15] 令(N, M_0)是1个标识网, S是N的1个信标. 如果存在P–不变式I, 使得$\forall p \in (\|I\|^- \cap S)$,

$$\max_{p^\bullet} = 1,\ \|I\|^+ \subseteq S,$$
$$\sum_{p\in P} I(p)M_0(p) > \sum_{p\in S} I(p)(\max_{p^\bullet} - 1)$$

成立, 则S是最大可控的.

假定$S^*_{\min} = \{p_1, p_2, \cdots, p_k\}$是原网$(N_0, M_0)$的一个必需信标. 添加$V_{S^*_{\min}}$, 使得$I$成为扩展网$(N_j, M_j)$的P–不变式, 这里$\forall p \in S^*_{\min}$, $I(p) = 1$, $\forall p \in P$, $M_j(p) = M_0(p)$, $I(V_{S^*_{\min}}) = -1$. 由引理1, $S^*_{\min}$是最大可控的,

$$I^{\mathrm T}M_0 = I^{\mathrm T}M_j = M_0(S^*_{\min}) - M_0(V_{S^*_{\min}}) > \sum_{p\in S^*_{\min}} I(p)(\max_{p^\bullet} - 1).$$

引入$S^*_{\min}$的控制深度变量$\xi_{S^*_{\min}}$, 来确定$V_{S^*_{\min}}$的初始标识, 表示为$M_0(V_{S^*_{\min}})$:

$$M_0(V_{S^*_{\min}}) = M_0(S^*_{\min}) - \xi_{S^*_{\min}},$$

其中:

$$\sum_{p\in S} I(p)(\max_{p^\bullet} - 1) < \xi_{S^*_{\min}},$$
$$\xi_{S^*_{\min}} \in \mathbb{N}_n = \{1, 2, \cdots, n\}.$$

在满足$S^*_{\min}$最大可控的条件下, 期望$\xi_{S^*_{\min}}$尽可能的小, 这样就可以减小对被控的Petri网许可行为的约束.

控制库所输入、输出弧的连接问题. 如果对$S^*_{\min}$添加的是一个普通控制库所, 其输出弧(输入弧)连接到对应的$\mathrm{Th}(S^*_{\min})$的输入集(输出集). 尽管这种方式会产生新的控制诱导信标[10](即, 包含控制库所的信标), 但是可以尽可能地减小了对被控的Petri网许可行为的约束[10]. 若对$S^*_{\min}$添加的是一个一般控制库所, 其输入弧连接到对应的$\mathrm{Th}(S^*_{\min})$的输出集, 其输出弧则连接到原网系统的源变迁. 因为源变迁表示进入系统的输入点, 该方式会限制系统的许可行为, 但是能够防止控制诱导信标的产生.

基于以上的表述, 下面形式化地给出两种添加控制库所的方法:

1) 对于一个$S^*_{\min}$, 若$\forall p \in \mathrm{Th}(S^*_{\min})$, $h_{S^*_{\min}}(p) = 1$, 则添加一个普通控制库所($W(p,t) = W(t,p) = 1$)使得: a) $M_0(V_{S^*_{\min}}) = M_0(S^*_{\min}) - \xi_{S^*_{\min}}$. 因为$\sum_{p\in S} I(p)(\max_{p^\bullet} - 1) = 0$, 所以$\xi_{S^*_{\min}} = 1$. b) 对于该普通控制库所的任何输出变迁$t$, 存在一条从CP指向$t$的弧. c) 对于该普通控制库所的任何输入变迁$t$, 存在一条从$t$指向CP的弧.

2) 对于一个$S^*_{\min}$, 若$\exists p \in \mathrm{Th}(S^*_{\min})$, $h_{S^*_{\min}}(p) > 1$, 则添加一个一般控制库所($W(p,t) \geqslant 1, W(t,p) \geqslant 1$)使得: a)

$$M_0(V_{S^*_{\min}}) = M_0(S^*_{\min}) - \xi_{S^*_{\min}},$$
$$\sum_{p\in S} I(p)(\max_{p^\bullet} - 1) < \xi_{S^*_{\min}},\ \xi_{S^*_{\min}} \in \mathbb{N}_n.$$

b) 该一般控制库所的输出弧连接到原网系统的源变迁. c) 该一般控制库所的输入弧连接到对应的$\mathrm{Th}(S^*_{\min})$的输出集.

5.2 死锁预防策略(Deadlock prevention policy)

用上述的方法可以确定出一个$S^*_{\min}$, 并且添加适当的控制库所使其最大可控. 然而, 一个死锁预防策略是要确定出所有的$S^*_{\min}$, 并对它们添加适当的控制库所, 使得被控网系统实现活性的目的. 所以, 本节提出能够实现这样目的的死锁预防策略, 即算法2, 其中: $|P| = |P_{\mathrm A}| + |P^0| + |P_{\mathrm R}|$, $j \in \mathbb{N}^+$, $\mathbb{N}^+ = \{1, 2, \cdots\}$, $\varPi_{S^*_{\min}}$和$|V|$分别表示原网系统(N, M_0)中的库所总数量、迭代步骤, 扩展网(N_j, M_j)中的必需信标的集合和添加的控制库所数量. 其算法流程图如图4所示.

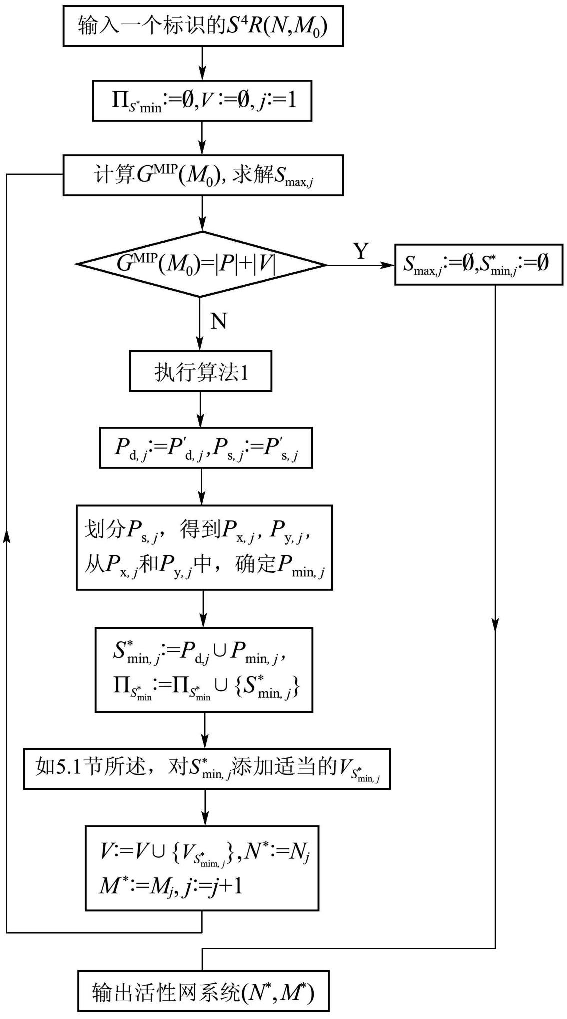

图 4 死锁预防算法2的流程图

Fig. 4 Flowchart of algorithm 2

在第一次迭代, 算法2求取出$S_{\max,1}$并且计算$G^{\mathrm{MIP}}(M_0)$. 验证$G^{\mathrm{MIP}}(M_0)=|P|+|V|(|V|=0)$是否成立, 若成立, 得到$S_{\max,1}=S^*_{\min,1}=\varnothing$, 表明原网系统$(N,M_0)$是活的. 否则, 算法2确定出$S^*_{\min,1}$并且添加适当的控制库所$V_{S^*_{\min,1}}$. 在第$j$次迭代, 算法2求取出$S_{\max,j}$并且计算出相应的$G^{\mathrm{MIP}}(M_0)$. 类似地, 验证$G^{\mathrm{MIP}}(M_0)=|P|+|V|\,(|V|\neq 0)$是否成立, 若成立, 得到$S_{\max,j}=\varnothing$, $S_{\min,j}=\varnothing$. 表明受控网系统是活的, 迭代控制进程结束. 否则, 算法2确定出$S_{\min,j}$并且添加适当的控制库所$V_{S^*_{\min,j}}$. 迭代控制持续进行, 直至无最大的死标识信标存在, 即$G^{\mathrm{MIP}}(M_0)=|P|+|V|(|V|\neq 0)$, 受控网系统是活的.

定理 3　令(N,M_0)是一个标识的S^{4}R. 对其施加算法2, 最后得到的扩展网(N^*,M^*)是活的.

证　假定$S^*_{\min,j}$和(N_j,M_j)是通过第j次迭代, 得到的必需信标和扩展网, 其中: $\forall p\in P_{\mathrm{A}}\cup P^0\cup P_{\mathrm{R}}$,

$$M_j(p)=M_0(p),$$
$$M_j(V_{S^*_{\min,j}})=M_j(S^*_{\min,j})-\xi_{S^*_{\min,j}},$$
$$\sum_{p\in S^*_{\min,j}} I(p)(\max_{p^\bullet}-1)<\xi_{S^*_{\min,j}}.$$

由定义9, 可得$\mathrm{Th}(S^*_{\min,j})$和$h_{S^*_{\min,j}}(p)$, 据此对$S^*_{\min,j}$添加适当的$V_{S^*_{\min,j}}$, 使得$I$成为扩展网$(N_j,M_j)$的P不变式, 这里$\forall p\in S^*_{\min,j}, I(p)=1, \forall p\in P, M_j(p)=M_0(p), I(V_{S^*_{\min,j}})=-1$. 通过调整$S^*_{\min,j}$的控制深度变量$\xi_{S^*_{\min,j}}(\sum\limits_{p\in S} I(p)(\max_{p^\bullet}-1)<\xi_{S^*_{\min,j}}$, $\xi_{S^*_{\min,j}}\in\mathbb{N}_n)$的大小, 来确定普通(一般)$V_{S^*_{\min}}$的初始标识$M_0(V_{S^*_{\min}})$, 使得

$$I^{\mathrm{T}}M_0=I^{\mathrm{T}}M_j=M_0(S^*_{\min})-M_0(V_{S^*_{\min}})>\sum_{p\in S^*_{\min}} I(p)(\max_{p^\bullet}-1)$$

成立, 满足引理1, 保证了$S^*_{\min,j}$满足最大可控性.

如5.1节所述, 普通和一般控制库所的添加都能够实现对相应必需信标的控制. 前者可以尽可能地减小对被控的Petri网许可行为的约束[10]. 但是, 会产生新的必需信标(即, 原网(N,M_0)没有的). 后者可以消除新的必需信标的产生, 然而, 会限制被控Petri网的许可行为. 所以, 随着迭代控制必需信标进程的演化, 不断的添加普通和一般控制库所, 不但使得相应的必需信标是最大可控的, 而且能够消除原网(N,M_0)已有的和扩展网(N_j,M_j)新产生的必需信标. 也就是说, 所需要控制的必需信标的数目随着该进程的演化是不断减少的. 该迭代控制必需信标的进程在$G^{\mathrm{MIP}}(M_0)=|P|+|V|$时结束, 表明无最大的死标识信标和必需信标的存在, 即, $S_{\max,j}=S^*_{\min,j}=\varnothing$. 算法2同样在$G^{\mathrm{MIP}}(M_0)=|P|+|V|$时终止, 由定理1和引理1可知, 得到的最后扩展网(N^*,M^*)是活的.　　证毕.

该策略采用MIP方法判定S^{4}R网中最大死标识信标的存在和被控的Petri网活性的问题. 尽管MIP问题的求解在理论上是NP-hard[10], 但是, 在计算上相比完全信标枚举方法更为有效. 此外, 在迭代过程的每一步, 求解出一个最大的死标识信标, 通过迭代式的信标提取和库所的分类, 得到必需信标. 通过显式控制必需信标, 从而达到隐式控制那些包含较多资源库所的信标的目的. 比对其他文献中的死锁预防策略, 获得了结构简单的活性被控网系统, 能够一定程度地改善控制器设计的计算复杂性和结构复杂性.

6　算例(Example)

对图1所示的S^{4}R网应用本死锁预防策略, 其迭代控制过程表述如下:

在第1次迭代, 应用MIP方法可得

$$S_{\max,1}=\{p_1,p_2,p_4,p_5,p_8,p_9,p_{10},p_{11},p_{12},p_{13},p_{14}\}.$$

由$(S_{\max,1}\cup {}^\bullet S_{\max,1}\cup S_{\max,1}{}^\bullet)$导出的子网$N_{S_{\max,1}}$如图2所示. 由于$G^{\mathrm{MIP}}(M_0)=5\neq|P|+|V|=16$, 则算法1输出:

$$P_1'=\{p_4,p_8,p_{10},p_{11},p_{12},p_{13}\},$$
$$P_1''=\{p_5,p_8,p_{10},p_{11},p_{12},p_{13},p_{14}\},$$
$$P^{\mathrm{R}}_{\mathrm{d},1}=\{p_{12},p_{13}\},\ P^{\mathrm{A}}_{\mathrm{d},1}=\{p_8,p_{10},p_{11}\},$$
$$P_{\mathrm{u},1}=\{p_1,p_2,p_9\},\ P_{\mathrm{s},1}=\{p_4,p_5,p_{14}\}.$$

进一步可得:

$$P^{\mathrm{R}}_{\mathrm{x},1}=\varnothing,\ P^{\mathrm{A}}_{\mathrm{x},1}=\{p_4\},\ P_{\mathrm{x},1}=\{p_4\},$$
$$P^{\mathrm{R}}_{\mathrm{y},1}=\{p_{14}\},\ P^{\mathrm{A}}_{\mathrm{y},1}=\{p_5\},\ P_{\mathrm{y},1}=\{p_5,p_{14}\}.$$

由$|P^{\mathrm{R}}_{\mathrm{x},1}|=|\varnothing|=0<|P^{\mathrm{R}}_{\mathrm{y},1}|=|\{p_{14}\}|=1$, 则

$$P_{\min,1}=P_{\mathrm{x},1},$$
$$S^*_{\min,1}=P_{\mathrm{d},1}\cup P_{\mathrm{x},1}=\{p_4,p_8,p_{10},p_{11},p_{12},p_{13}\}.$$

由定义9, 得到

$$\mathrm{Th}(S^*_{\min,1})=\{2p_1,5p_2,5p_3,3p_7,4p_9\}.$$

由$\sum\limits_{p\in S^*_{\min,1}} I(p)(\max_{p^\bullet}-1)=4$, 可得$\xi_{S^*_{\min,1}}=5$, $M_0(V_{S^*_{\min,1}})=21+5-5=21$. 由于$h_{S^*_{\min}}(p_i)>1(i=1,2,3,7,9)$, 则

$${}^\bullet CP_1=\{5t_4,3t_9,4t_{11}\},$$
$$CP_1{}^\bullet=\{2t_1,3t_2,t_7,2t_8,t_{10}\}.$$

所以, 对$S^*_{\min,1}$添加一个一般控制库所$V_{S^*_{\min,1}}(p_{17})$.

在第2次迭代, 应用MIP方法同样得到

$$S_{\max,2}=\{p_1,p_2,p_5,p_7,p_8,p_9,p_{10},p_{11},p_{12},p_{13},p_{14}\}.$$

由$(S_{\max,2}\cup {}^\bullet S_{\max,2}\cup S_{\max,2}{}^\bullet)$导出的子网$N_{S_{\max,2}}$如图5所示.

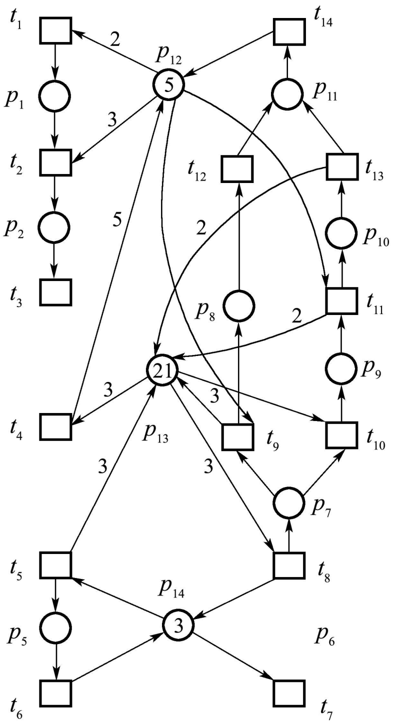

图 5 $(S_{\max,2} \cup {}^{\bullet}S_{\max,2} \cup S_{\max,2}{}^{\bullet})$导出的子网

Fig. 5 The subnet generated by $(S_{\max,2}\cup{}^{\bullet}S_{\max,2}\cup S_{\max,2}{}^{\bullet})$

由于$G^{\mathrm{MIP}}(M_0) = 6 \neq |P| + |V| = 17$, 则算法1输出:

$$P_2' = \{p_5, p_7, p_9, p_{10}, p_{13}, p_{14}\},$$
$$P_2'' = \{p_5, p_8, p_{10}, p_{11}, p_{12}, p_{13}, p_{14}\},$$
$$P_{\mathrm{d},2}^{\mathrm{R}} = \{p_{13}, p_{14}\},\ P_{\mathrm{d},2}^{\mathrm{A}} = \{p_5, p_{10}\},$$
$$P_{\mathrm{u},2}=\{p_1, p_2\},\ P_{\mathrm{s},2}=\{p_7, p_8, p_9, p_{11}, p_{12}\}.$$

进一步得到:

$$P_{\mathrm{x},2}^{\mathrm{R}} = \varnothing,\ P_{\mathrm{x},2}^{\mathrm{A}} = \{p_7, p_9\},\ P_{\mathrm{x},2} = \{p_7, p_9\},$$
$$P_{\mathrm{y},2}^{\mathrm{R}} = \{p_{12}\},\ P_{\mathrm{y},2}^{\mathrm{A}} = \{p_8, p_{11}\},$$
$$P_{\mathrm{y},2} = \{p_8, p_{11}, p_{12}\}.$$

由$|P_{\mathrm{x},2}^{\mathrm{R}}| = |\varnothing| = 0 < |P_{\mathrm{y},2}^{\mathrm{R}}| = |\{p_{12}\}| = 1$, 则

$$P_{\min,2} = P_{\mathrm{x},2},$$
$$S_{\min,2}^{*} = P_{\mathrm{d},2} \cup P_{\mathrm{x},2} = \{p_5, p_7, p_9, p_{10}, p_{13}, p_{14}\}.$$

由定义9, 可得$\mathrm{Th}(S_{\min,2}^{*}) = \{3p_4, p_6\}$.

由 $\sum\limits_{p\in S_{\min,2}^{*}} I(p)(\max_{p^{\bullet}} - 1) = 2$, 可得:

$$\xi_{\mathrm{S}_{\min,2}^{*}} = 3,\ M_0(V_{\mathrm{S}_{\min,2}^{*}}) = 21 + 3 - 3 = 21.$$

由于$h_{\mathrm{S}_{\min,2}^{*}}(p_i) \geqslant 1 (i = 4, 6)$, 则

$${}^{\bullet}CP_2 = \{3t_5, t_8\},\ CP_2{}^{\bullet} = \{t_1, 2t_4, t_7\}.$$

因此, 对$S_{\min,2}^{*}$添加一个一般控制库所$V_{\mathrm{S}_{\min,2}^{*}}(p_{18})$.

在下一次迭代, 应用MIP方法可得

$$S_{\max,3} = \{p_1, p_2, p_5, p_8, p_9, p_{10}, p_{11}, p_{12}, p_{13}, p_{14}\}.$$

由$(S_{\max,3} \cup {}^{\bullet}S_{\max,3} \cup S_{\max,3}{}^{\bullet})$导出的子网$N_{\mathrm{S}_{\max,3}}$如图6所示.

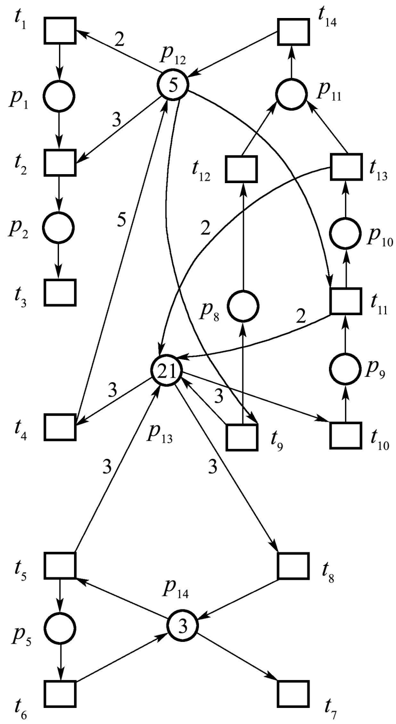

图 6 $(S_{\max,3} \cup {}^{\bullet}S_{\max,3} \cup S_{\max,3}{}^{\bullet})$导出的子网

Fig. 6 The subnet generated by $(S_{\max,3}\cup{}^{\bullet}S_{\max,3}\cup S_{\max,3}{}^{\bullet})$

由于$G^{\mathrm{MIP}}(M_0) = 8 \neq |P| + |V| = 18$, 则算法1输出

$$P_3' = P_3'' = \{p_5, p_8, p_{10}, p_{11}, p_{12}, p_{13}, p_{14}\},$$
$$P_{\mathrm{d},3}^{\mathrm{R}} = \{p_{12}, p_{13}, p_{14}\},\ P_{\mathrm{d},3}^{\mathrm{A}} = \{p_5, p_8, p_{10}, p_{11}\},$$
$$P_{\mathrm{u},3} = \{p_1, p_2, p_9\},\ P_{\mathrm{s},3} = \varnothing.$$

而且, 可得

$$P_{\mathrm{x},3}^{\mathrm{R}} = P_{\mathrm{x},3}^{\mathrm{A}} = P_{\mathrm{x},3} = P_{\mathrm{y},3}^{\mathrm{R}} = P_{\mathrm{y},3}^{\mathrm{A}} = P_{\mathrm{y},3} = \varnothing.$$

所以

$$P_{\min,3} = \varnothing,$$
$$S_{\min,3}^{*} = P_{\mathrm{d},3} \cup P_{\min,3} =$$
$$\{p_5, p_8, p_{10}, p_{11}, p_{12}, p_{13}, p_{14}\}.$$

由定义9, 得到

$$\mathrm{Th}(S_{\min,3}^{*}) = \{2p_1, 5p_2, 5p_3, 3p_4, p_6, 3p_7, 4p_9\}.$$

由 $\sum\limits_{p\in S_{\min,3}^{*}} I(p)(\max_{p^{\bullet}} - 1) = 4$, 可得$\xi_{\mathrm{S}_{\min,3}^{*}} = 5$, $M_0(V_{\mathrm{S}_{\min,3}^{*}}) = 21+3+5-5 = 24$. 由于$h_{\mathrm{S}_{\min,3}^{*}}(p_i) \geqslant 1 (i = 1, 2, 3, 4, 6, 7, 9)$, 则

$${}^{\bullet}CP_3 = \{2t_4, 3t_5, 3t_9, 4t_{11}\},$$
$$CP_3{}^{\bullet} = \{2t_1, 3t_2, t_7, 2t_8, t_{10}\}.$$

所以, 对$S_{\min,3}^{*}$添加一个一般控制库所$V_{\mathrm{S}_{\min,3}^{*}}(p_{19})$.

在第4次迭代, 应用MIP方法得到

$$S_{\max,4} = \{p_1, p_2, p_3, p_5, p_8, p_{10}, p_{11}, p_{12}\}.$$

由$(S_{\max,4} \cup {}^{\bullet}S_{\max,4} \cup S_{\max,4}{}^{\bullet})$导出的子网$N_{\mathrm{S}_{\max,4}}$如图7所示.

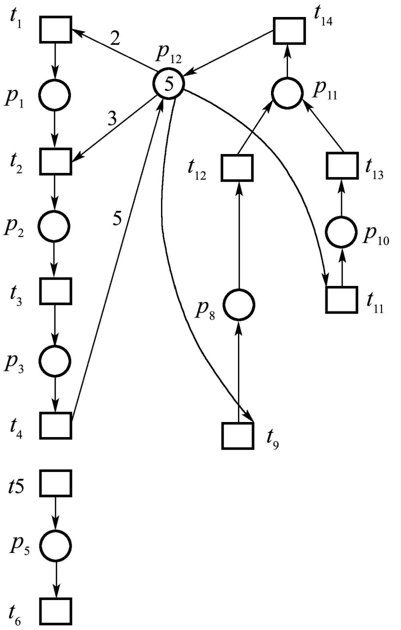

图 7 $(S_{\max,4} \cup {}^{\bullet}S_{\max,4} \cup S_{\max,4}{}^{\bullet})$导出的子网

Fig. 7 The subnet generated by $(S_{\max,4} \cup {}^{\bullet}S_{\max,4} \cup S_{\max,4}{}^{\bullet})$

由于$G^{\text{MIP}}(M_0) = 11 \neq |P| + |V| = 19$, 则算法1输出:

$$P'_4 = P''_4 = \{p_2, p_3, p_8, p_{10}, p_{11}, p_{12}\},$$

$$P^{\text{R}}_{\text{d},4} = \{p_{12}\},\ P^{\text{A}}_{\text{d},4} = \{p_2, p_3, p_8, p_{10}, p_{11}\},$$

$$P_{\text{u},4} = \{p_1, p_5\},\ P_{\text{s},4} = \varnothing.$$

而且, 得到

$$P^{\text{R}}_{\text{x},4} = P^{\text{A}}_{\text{x},4} = P_{\text{x},4} = P^{\text{R}}_{\text{y},4} = P^{\text{A}}_{\text{y},4} = P_{\text{y},4} = \varnothing.$$

所以$P_{\min,4} = \varnothing$, $S^*_{\min,4} = P_{\text{d},4} \cup P_{\min,4} = \{p_2, p_3, p_8, p_{10}, p_{11}, p_{12}\}$. 由定义9, 可得$\text{Th}(S^*_{\min,4}) = \{2p_1\}$, $h_{\text{S}^*_{\min,4}}(p_1) = 2$.

由$\sum\limits_{p \in S^*_{\min,4}} I(p)(\max_{p^{\bullet}} - 1) = 2$, 得到$\xi_{\text{S}^*_{\min,4}} = 3$, $M_0(V_{\text{S}^*_{\min,4}}) = 5 - 3 = 2$. 由于$h_{\text{S}^*_{\min,4}}(p_1) = 2$, 则${}^{\bullet}CP_4 = \{2t_2\}$, $CP_4{}^{\bullet} = \{2t_1\}$. 因此, 对$S^*_{\min,4}$添加一个一般控制库所$V_{\text{S}^*_{\min,4}}(p_{20})$.

在第5次迭代, 应用MIP方法得到$S_{\max,5} = S^*_{\min,5} = \varnothing$和$G^{\text{MIP}}(M_0) = |P| + |V| = 20$, 表明无最大死标识信标的存在以及迭代执行本死锁预防策略的终止. 所以, 最后得到了活性的受控网系统.

本死锁预防策略的迭代控制过程和结果见表1. 在Windows XP操作系统, 2.83GHz Intel(R) Dual CPU和3.48G内存的条件下, 它与文献[11]中的方法(表示为TGC$^+$)和文献[16]中的方法(表示为UZ)比较如表2所示. 添加了4个一般控制库所($V_{\text{S}^*_{\min,1}}, V_{\text{S}^*_{\min,2}}, V_{\text{S}^*_{\min,3}}, V_{\text{S}^*_{\min,4}}$)后, 所得到的活性Petri网受控系统如图8所示.

表 1 对图1所示S^{4}R网应用该死锁预防策略的处理结果

Table 1 Results of a deadlock prevention policy for a marked S^{4}R shown in Fig.1

j	$S_{\max,j}$	$S^*_{\min,j}$	$\text{Th}(S^*_{\min,j})$	${}^{\bullet}CP_j$	$CP_j{}^{\bullet}$	$M_0(CP_j)$	$G^{\text{MIP}}(M_0)$	$\|P\|+\|V\|$
1	$\{p_1, p_2, p_4, p_5, p_8, p_9, p_{10}, p_{11}, p_{12}, p_{13}, p_{14}\}$	$\{p_4, p_8, p_{10}, p_{11}, p_{12}, p_{13}\}$	$\{2p_1, 5p_2, 5p_3, 3p_7, 4p_9\}$				5	16
2	$\{p_1, p_2, p_5, p_7, p_8, p_9, p_{10}, p_{11}, p_{12}, p_{13}, p_{14}\}$	$\{p_5, p_7, p_9, p_{10}, p_{13}, p_{14}\}$	$\{3p_4, p_6\}$	$5t_4, 3t_9, 4t_{11}$	$2t_1, 3t_2, t_7, 2t_8, t_{10}$	21	6	17
3	$\{p_1, p_2, p_5, p_8, p_9, p_{10}, p_{11}, p_{12}, p_{13}, p_{14}\}$	$\{p_5, p_8, p_{10}, p_{11}, p_{12}, p_{13}, p_{14}\}$	$\{2p_1, 5p_2, 5p_3, 3p_4, p_6, 3p_7, 4p_9\}$	$3t_5, t_8$	$t_1, 2t_4, t_7$	21	8	18
4	$\{p_1, p_2, p_3, p_5, p_8, p_{10}, p_{11}, p_{12}\}$	$\{p_2, p_3, p_8, p_{10}, p_{11}, p_{12}\}$	$\{2p_1\}$	$2t_4, 3t_5, 3t_9, 4t_{11}$	$2t_1, 3t_2, t_7, 2t_8, t_{10}$	24	11	19
5	$\varnothing$	$\varnothing$	$\varnothing$	$2t_2$	$2t_1$	2	20	20

表 2 本策略, TGC$^+$方法以及UZ方法的比较

Table 2 Comparison of this policy with the methods of TGC$^+$ and UZ

评判标准	添加的控制库所数目	许可行为的状态数	计算时间/s	内存使用/M
本策略	4	49559	20	17.788
TGC$^+$	5	51386	22	19.780
UZ	8	48752	19	21.588

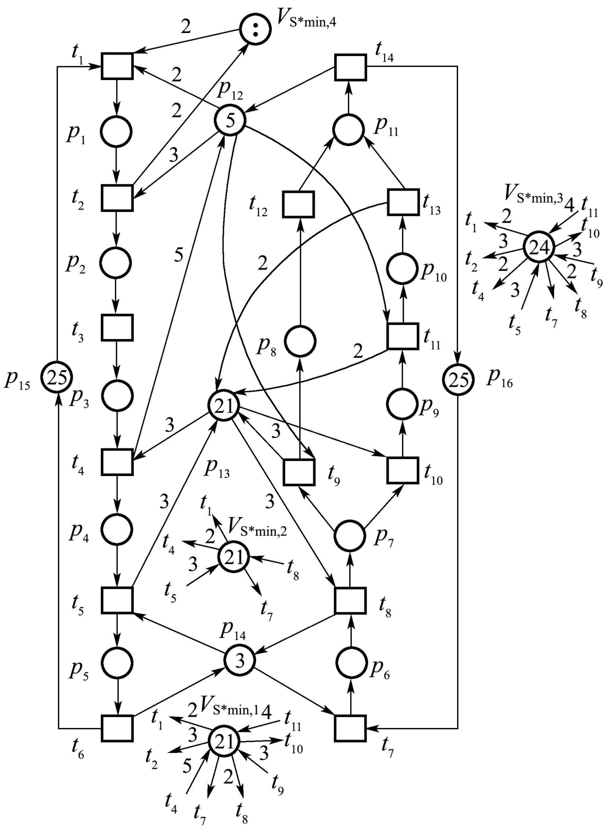

图 8 活性S^{4}R网受控系统

Fig. 8 A live controlled system of an S^{4}R

由上述的迭代控制过程可知, 本策略通过显式控制4个必需信标, 从而实现了对其他两个包含较多资源库所的信标的隐式控制, 最后得到了一个结构简单的活性被控网.

7 结论(Conclusions)

针对S^{4}R网的死锁问题, 本文提出了对最大死标识信标进行库所分类的算法, 表征死锁状态的必需信标的概念以及将混合整数规划算法(MIP)与必需信标控制相结合的死锁预防策略. MIP方法用以判定S^{4}R网中最大死标识信标的存在和活性的问题. 通过库所分类和迭代式信标的提取, 得到必需信标并且添加适当的控制库所, 使其满足最大可控, 直至获得活性的被控网系统. 由于通过显式控制必需信标, 从而实现了对其他包含较多资源库所的信标的隐式控制, 最后得到了一个结构简单的活性被控网. 理论分析和算例验证了该策略的正确性和有效性.

参考文献(References):

[1] MURATA T. Petri nets: properties, analysis, and applications[J]. *Proceedings of the IEEE*, 1989, 77(4): 541 – 580.

[2] ABDALLAH I B, ELMARAGHY H A. Deadlock prevention and avoidance in FMS: a Petri net based approach[J]. *International Journal of Advanced Manufacturing Technology*, 1998, 14(10): 704 – 715.

[3] EZPELETA J, COLOM J M, MARTINEZ J. A Petri net based deadlock prevention policy for flexible manufacturing systems[J]. *IEEE Transactions on Robotics and Automation*, 1995, 11(2): 173 – 184.

[4] FANTI M P, ZHOU M C. Deadlock control methods in automated manufacturing systems[J]. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 2004, 34(1): 5 – 22.

[5] LI Z W, ZHOU M C. Elementary siphons of Petri nets and their application to deadlock prevention in flexible manufacturing systems[J]. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 2004, 34(1): 38 – 51.

[6] VISWANADHAM N, NARAHARI Y, JOHNSON T L. Deadlock prevention and deadlock avoidance in flexible manufacturing systems using Petri net models[J]. *IEEE Transactions on Robotics and Automation*, 1990, 6(6): 713 – 723.

[7] PARK J, REVELIOTIS S A. Deadlock avoidance in sequential resource allocation systems with multiple resource acquisitions and flexible routings[J]. *IEEE Transactions on Automatic Control*, 2001, 46(10): 1572 – 1583.

[8] HUANG Y S, JENG M D, XIE X L, et al. Siphon-based deadlock prevention policy for flexible manufacturing systems[J]. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 2006, 36(6): 1248 – 1256.

[9] 吴哲辉. Petri网导论[M]. 北京: 机械工业出版社, 2006.
(WU Zhehui. *Introduction to Petri Nets*[M]. Beijing: China Machine Press, 2006.)

[10] LI Z W, ZHOU M C. Deadlock resolution in automated manufacturing systems[M] //*A Novel Petri Net Approach*. London, UK: Springer, 2009.

[11] TRICAS F, GARCIA-VALLES F, COLOM J M, et al. An iterative method for deadlock prevention in FMSs[C] //*Proceedings of 5th Workshop Discrete Event Systems.* Boston, American: Kluwer Academic, 2000: 139 – 148.

[12] CHAO D Y. Computation of elementary siphons for deadlock control[J]. *The Computer Journal*, 2006, 49(4): 470 – 479.

[13] LI Z W, ZHOU M C. On siphon computation for deadlock control in a class of Petri nets[J]. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 2008, 38(3): 667 – 679.

[14] HSIEH F S, CHANG S C. Dispatching-driven deadlock avoidance controller synthesis for flexible manufacturing systems[J]. *IEEE Transactions on Robotics and Automation*, 1994, 10(2): 196 – 209.

[15] BARKAOU K, PEYRE J F. On liveness and controlled siphons in Petri nets[C] //*Proceedings of 17th International Conference on Application and Theory of Petri Nets*. New York, American: Springer-Verlag, 1996, 1091: 57 – 72.

[16] UZAM M, ZHOU M C. An improved iterative synthesis method for liveness enforcing supervisors of flexible manufacturing systems[J]. *International Journal of Production Research*, 2006, 44(10): 1987 – 2030.

作者简介:

李绍勇 (1966—), 男, 博士研究生, 副教授, 主要从事离散事件系统监控的应用研究, E-mail: lishaoyong99@163.com;

王安荣 (1970—), 男, 博士, 副教授, 主要从事离散事件系统监督控制理论研究与实际应用, E-mail: arwang@mail.xidian.edu.cn.