A Method for the Synthesis of Deadlock Prevention Controllers in Systems Modeled by Petri Nets

Marian V. Iordache
Department of
Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556
iordache.1@nd.edu

John O. Moody
Lockheed Martin
Federal Systems
1801 State Rt. 17C, MD 0210
Owego, NY 13827-3998
john.moody@lmco.com

Panos J. Antsaklis
Department of
Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556
antsaklis.1@nd.edu

Deadlock is the faulty condition of a system in which no action can be performed.

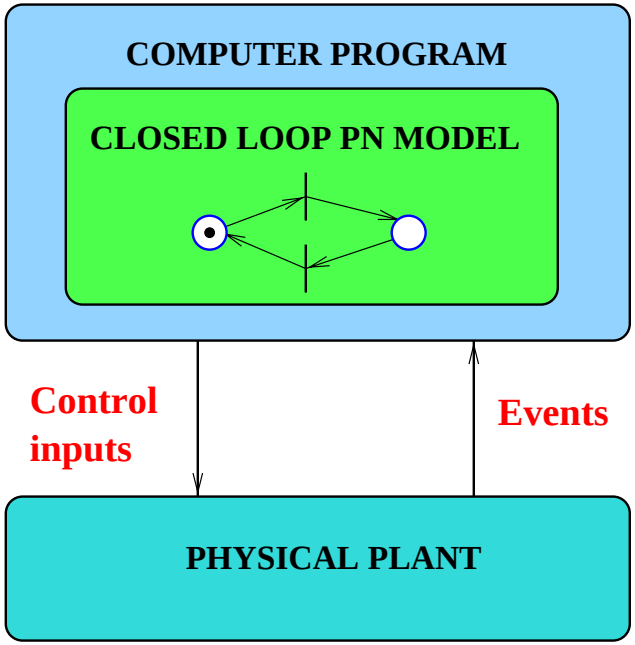
Deadlock prevention is a supervision goal of DES which is important in areas like:

- Manufacturing Systems and Robotics in general
- Operating Systems, Distributed Operating Systems
- Communications

Deadlock prevention means to avoid the deadlock states and the states which unavoidably lead to deadlock.

A **Petri net structure** is the tuple $\mathcal{N} = (P, T, F, W)$, where P is the finite **set of places**, T the finite **set of transitions**, $F \subseteq (P \times T) \cup (T \times P)$ is the **set of transition arcs**, and $W : F \to \mathbb{N}^*$ the **weight** of transition arcs. A **marking** of a Petri net is a map $\mu : P \to \mathbb{N}$. The **marking vector** is $\underline{\mu} = [\mu(p_1) \ldots \mu(p_n)]^T$, for $P = \{p_1, \ldots p_n\}$.

Closed loop PN model = Plant PN model + Controller PN model

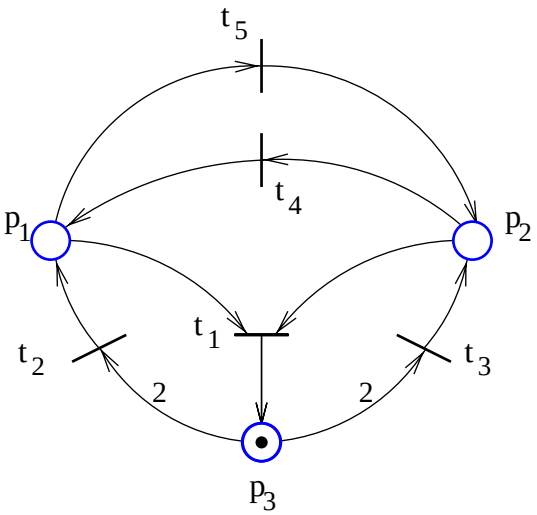


State $\longleftrightarrow$ Marking (μ)

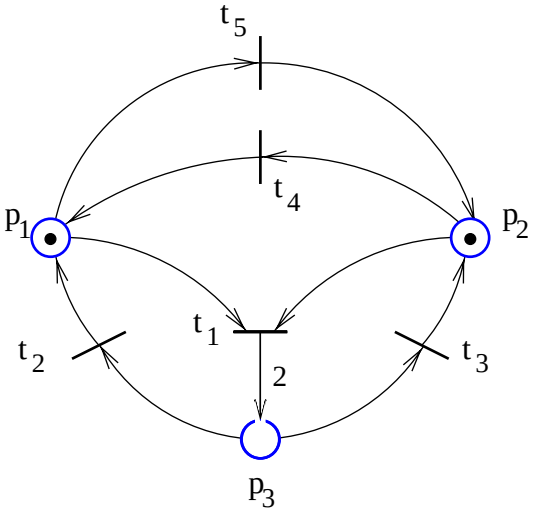
Event $\longleftrightarrow$ Transition firing (q)

Events $\longrightarrow$ State update

State $\longrightarrow$ Restriction of control inputs



Deadlock

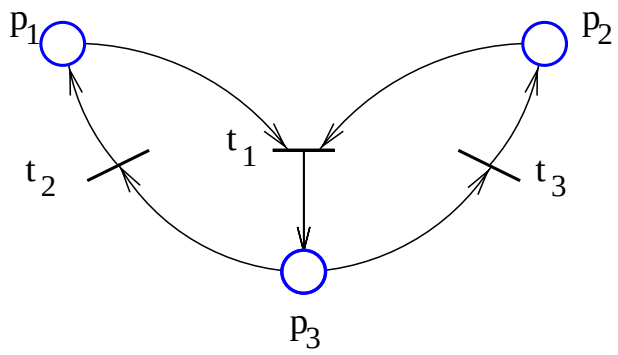


Live Petri net

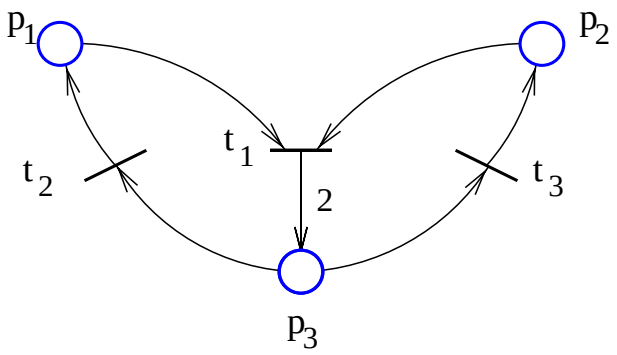
A Petri net $\mathcal{N}$ is said to be in **deadlock** for the marking μ, if no transition is enabled.

A Petri net $(\mathcal{N}, \mu_0)$ is said to be

- **live** if for all transitions t and for all reachable markings μ, μ enables some firing sequence that includes t.
- **deadlock-free** if for all reachable markings there is an enabled transition.



Not even partially repetitive



Repetitive

A reachable marking μ can expressed as: $\underline{\mu} = \underline{\mu}_0 + Dq$, where D is the **incidence matrix** and q the **firing count vector**.

A Petri net is **(partially) repetitive** iff $\exists x \neq 0,\ x > 0\ (x \geq 0)$ such that $Dx \geq 0$.

There are markings such that *liveness (deadlock-freedom)* is enforcible if and only if the Petri net is *(partially) repetitive*.

Siphons

Given the Petri net $\mathcal{N} = (P, T, F, W)$, for $p \in P$ and $S \subseteq P$:

$$\bullet p = \{t \in T : (t, p) \in F\},\ \bullet S = \bigcup_{p \in S} \bullet p \ (\textit{the preset operation})$$

$$p\bullet = \{t \in T : (p, t) \in F\},\ S\bullet = \bigcup_{p \in S} p\bullet \ (\textit{the postset operation})$$

$S \neq \emptyset$ is a **siphon** if $\bullet S \subseteq S\bullet$

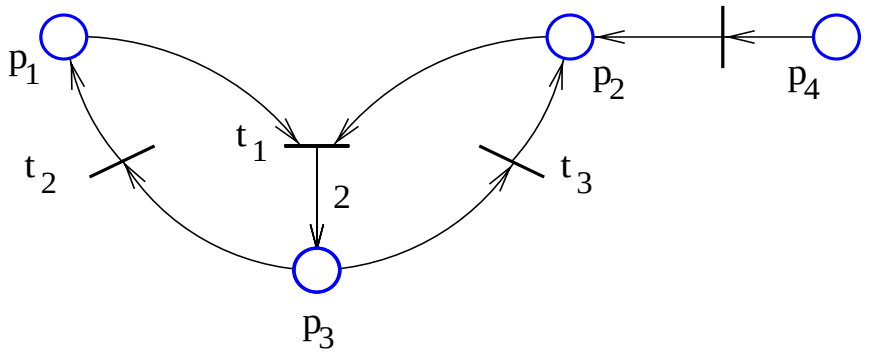
$\mathcal{N}$ is **PT-ordinary** if $W(x) = 1\ \forall x \in F \cap (P \times T)$

Proposition 1. *If μ is a deadlock marking of a PT-ordinary Petri net, there is an empty siphon S (i.e. $\mu(p) = 0\ \forall p \in S$.)*

Examples:

{p4}, {p1, p3}, {p2, p3, p4} and

{p1, p2, p3, p4} are siphons



Supervisory Control of Petri Nets

Note that the condition that a siphon is not empty is $\sum_{p \in S} \mu(p) \geq 1$.

Enforcing linear constraints of the form

$$L\underline{\mu} \geq b$$

on Petri nets has been studied by Giua, Yamalidou and Moody.

The supervision is achieved by enhancing the target Petri net with additional places, called **control places**, each enforcing one of the inequalities of $L\underline{\mu} \geq b$.

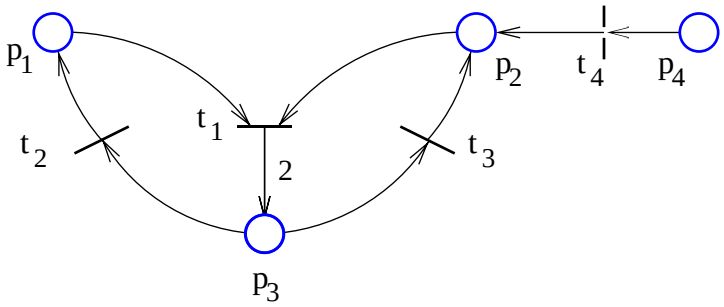
The closed loop Petri net is described by the incidence matrix $D_C = [D^T, D^T L^T]^T$.

Active Subnets

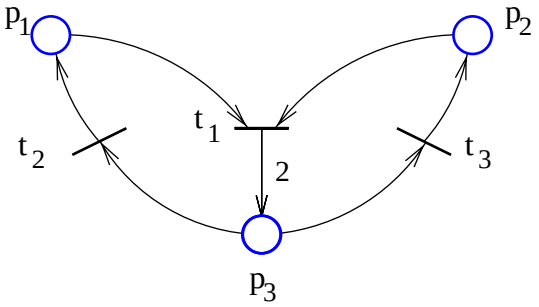
Theorem 1. *Consider a Petri net $\mathcal{N} = (P, T, F, W)$ which is not repetitive. At least one transition exists such that for any given initial marking it cannot fire infinitely often. Let T_D be the set of all such transitions. There are initial markings μ_0 and a supervisory policy Ξ such that $\forall \mu \in \mathcal{R}(\mathcal{N}, \mu_0, \Xi)$, no transition in $T \setminus T_D$ is dead.*

The proof shows that there is a nonnegative vector x such that $x(i) > 0$ for $t_i \in T \setminus T_D$ and $x(i) = 0$ for $t_i \in T_D$ and $Dx \geq 0$, where D is the incidence matrix.

In this context we define the **active subnet** of a Petri net as the subnet remaining after successively removing all transitions which contain source places in their preset and the source places.



Petri net



Active subnet

The deadlock prevention problem considered is:

> Given a target Petri net $\mathcal{N}_0$, find the marking constraints $L\underline{\mu} \geq b$ such that $\mathcal{N}_0$ supervised according to $L\underline{\mu} \geq b$ is deadlock-free for all initial markings μ_0 satisfying $L\underline{\mu}_0 \geq b$.

What is new:

- The initial marking is not assumed to be known. Rather it is considered to be a parameter.
- Although the reachability graph is not used, the procedure considers the Petri net types of previous approaches as well as the *partially repetitive* and/or *unbounded* Petri nets.

Given a target Petri net $\mathcal{N}_0$, the deadlock prevention procedure generates a sequence of PT-ordinary Petri nets, $\mathcal{N}_1$, $\mathcal{N}_2$, ... $\mathcal{N}_k$, increasingly enhanced for deadlock prevention, as well as their active subnets $\mathcal{N}_1^A$, $\mathcal{N}_2^A$, ... $\mathcal{N}_k^A$.

$\mathcal{N}_1$ is $\mathcal{N}_0$ transformed to be PT-ordinary.

In each iteration i the new minimal siphons S of $\mathcal{N}_i^A$ are controlled by enforcing the linear constraint $\sum_{p \in S} \mu(p) \geq 1$ in $\mathcal{N}_i$ with *control places*.

Each inequality associated to a siphon corresponds to a inequality in the target net $\mathcal{N}_0$. This is how the constraints $L\underline{\mu} \geq b$ are found.

The procedure terminates when after some iteration k, $\mathcal{N}_k^A$ has no new minimal siphons.

$\mathcal{N}_0$ supervised according to $L\underline{\mu} \geq b$ is deadlock-free for all initial markings μ_0 satisfying $L\underline{\mu}_0 \geq b$.

Deadlock Prevention Procedure

Input: The target Petri net $\mathcal{N}_0$
Output: Two sets of constraints (L, b) and (L_0, b_0).

repeat

1. **For** every new uncontrolled minimal siphon S of $\mathcal{N}_i^A$ **do**

   if S needs control enforcement with a *control place* **then**
   add control place to the Petri net and inequality in (L, b).

   else
   add inequality to (L_0, b_0).

2. Transform the current net to a PT-ordinary Petri net.

3. Update the active subnet (takes in account the new source places.)

until no new minimal active siphon appears.

Deadlock is prevented by supervising $\mathcal{N}_0$ with $L\underline{\mu} \geq b$, for all initial markings μ_0 such that $L\underline{\mu}_0 \geq b$ and $L_0\underline{\mu}_0 \geq b_0$.

Main Results

A **siphon control failure** is said to occurr when the control place C added to control the siphon S results such that $\exists t \in S\bullet \setminus \bullet S : W(C,t) > 1$.

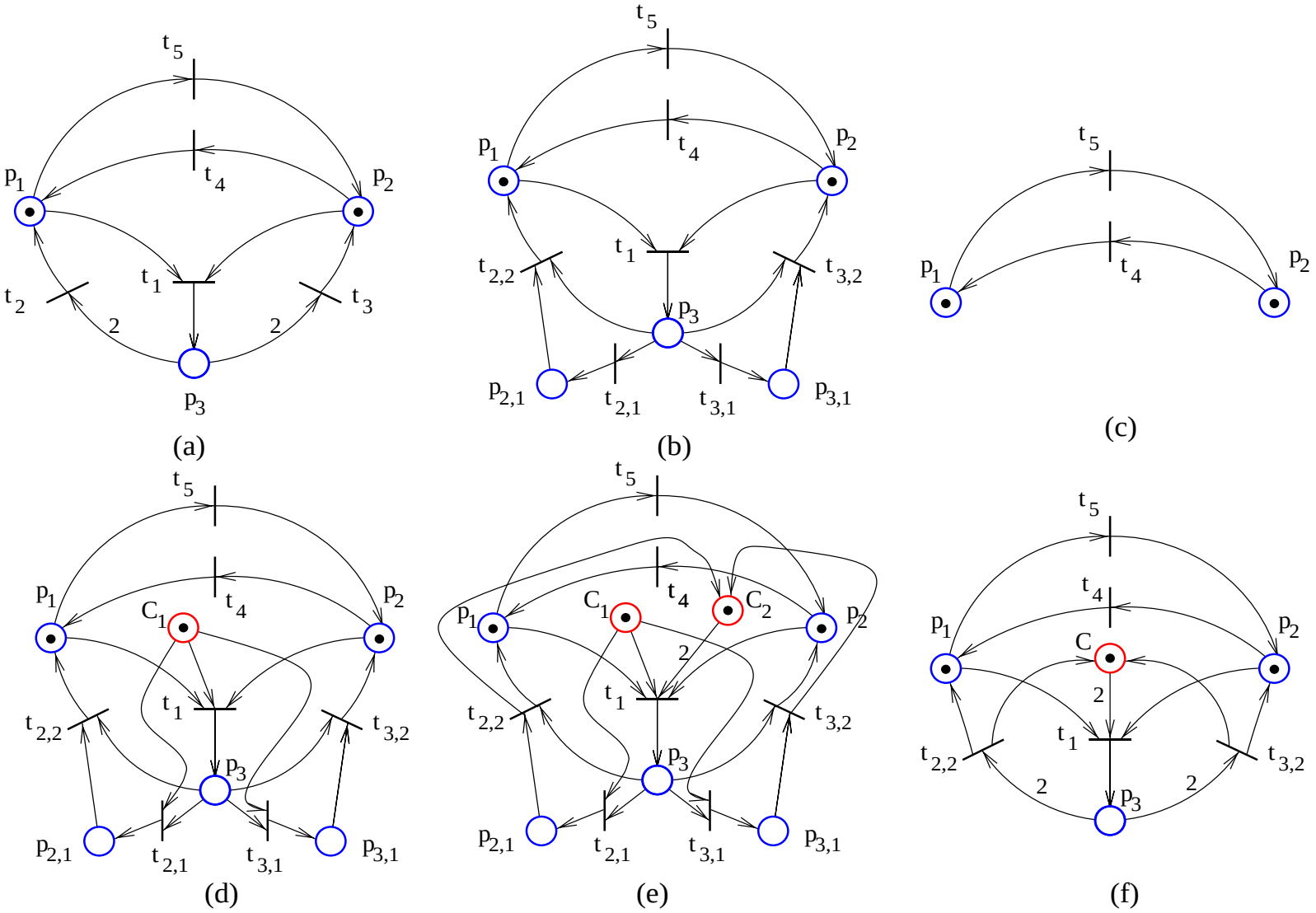
Theorem 2. **Deadlock Prevention.** *Assume that the procedure terminates and that no siphon control failure occurred. Let $\mathcal{N}_0$ be the target Petri net and $\mathcal{N}_k$ the net produced by the last iteration. If $\mathcal{N}_k^A$, the active subnet of the last iteration, is nonempty, then $\mathcal{N}_0$ in closed loop with the supervisor enforcing $L\mu \geq b$ is deadlock-free for all initial markings μ_0 of $\mathcal{N}_0$ such that $L\mu_0 \geq b$ and $L_0\mu_0 \geq b_0$.*

Proposition 2. *Deadlock cannot be prevented under any circumstances if $\mathcal{N}_0^A$ is empty.*

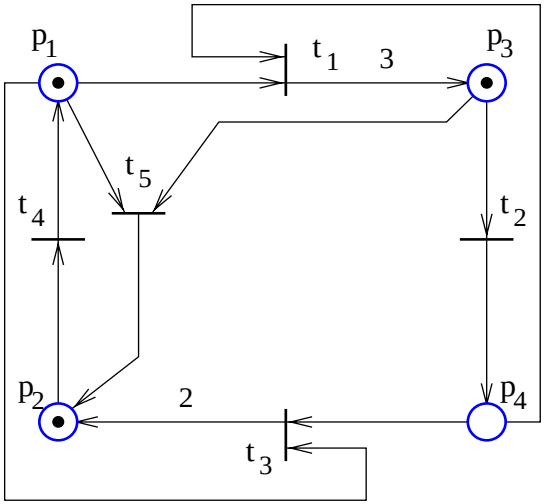
Theorem 3. **Permissivity.** *Then the deadlock prevention method provides a supervisor no more restrictive than any liveness enforcing supervisor, if any exists.*

Outline of Main Results

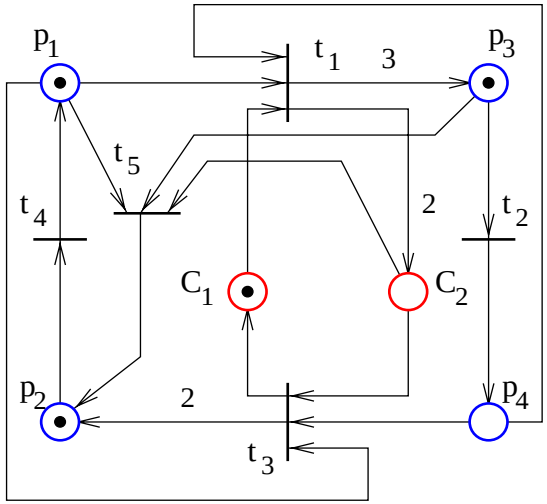
- The target Petri net $\mathcal{N}_0$, supervised according to the constraints (L, b), is deadlock-free for all initial markings μ_0 such that $L_0\underline{\mu}_0 \geq b_0$ and $L\underline{\mu}_0 \geq b$.
- It is possible to detect situations in which the structure of $\mathcal{N}_0$ does not allow deadlock to be prevented for any initial marking.
- The procedure does not provide the least restrictive supervisor that prevents deadlock. *In case that there exist initial markings μ_0 such that liveness can be enforced in $(\mathcal{N}_0, \mu_0)$, the supervisor (for deadlock-freedom) is no more restrictive than any supervisor enforcing liveness*.
- There are particular cases in which the supervisor of our algorithm enforces as well liveness. When liveness is enforced, the algorithm provides the least restrictive liveness enforcing supervisor.
- With additional modification, the procedure can be guaranteed to terminate. The termination result has practical importance for bounded Petri nets, assuming that a bounded marking region is known.



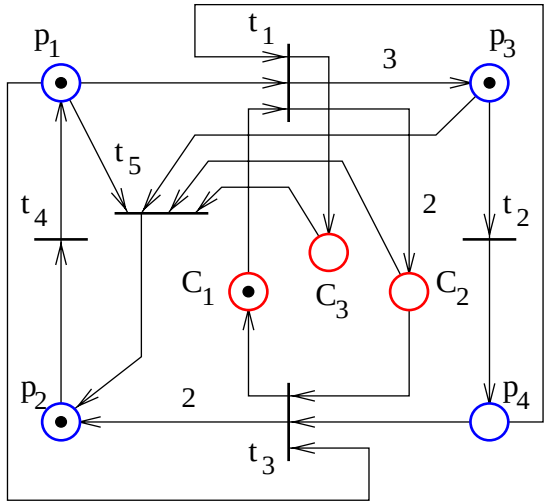
$L = [1, 1, 0]$, $b = 1$, $L_0 = [\,]$ and $b_0 = [\,]$



(a)

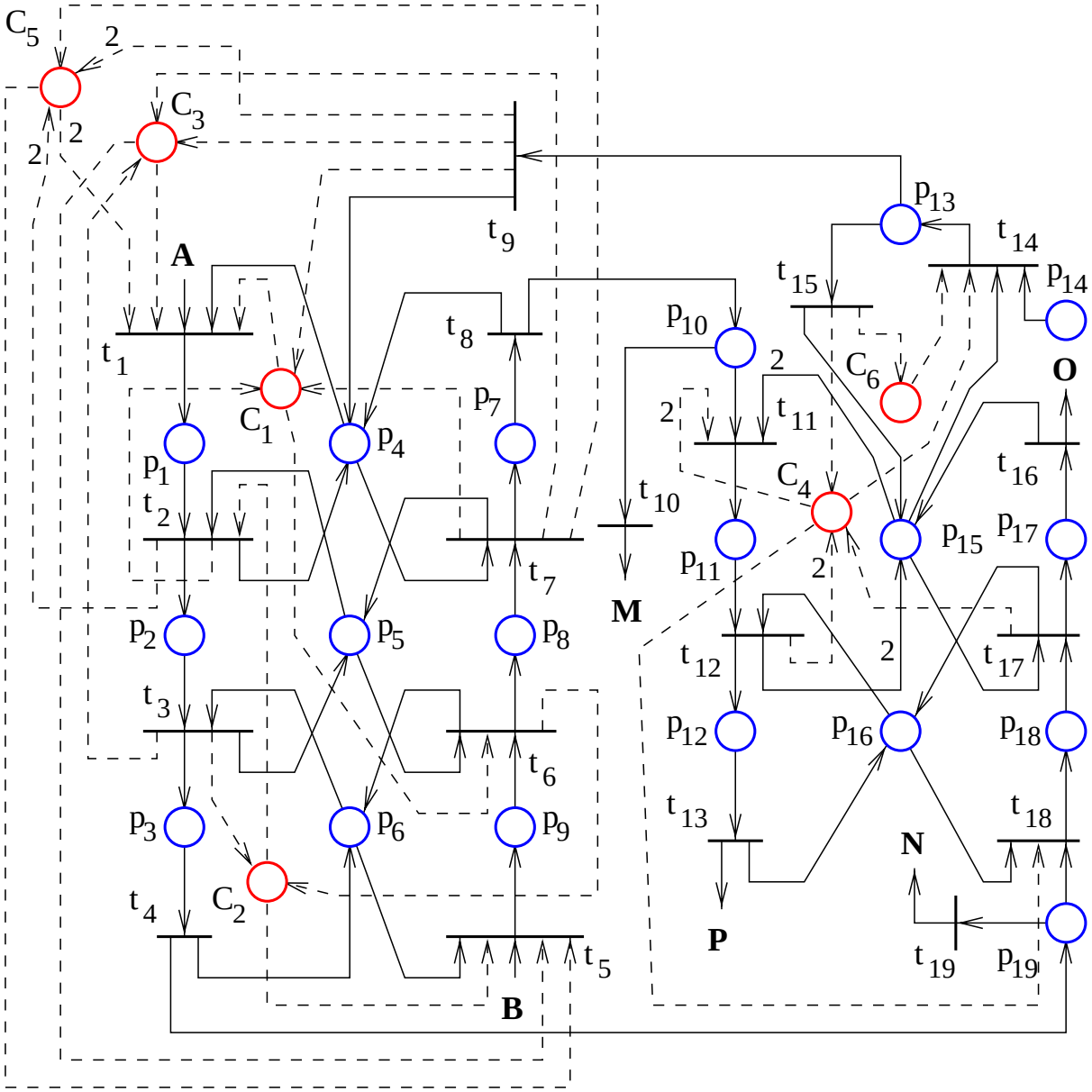


(b)



(c)

$$L = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \qquad b = \begin{bmatrix} 1 \\ 1 \\ 3 \end{bmatrix} \qquad L_0 = [\,] \qquad b_0 = [\,]$$


C5
2
2
2
C3
t9
A
t1
C1
p4
p7
t8
p1
t2
t7
p2
p5
p8
t3
t6
p3
p6
p9
t4
C2
t5
B
M
p10
2
t11
C6
C4
t10
p11
2
p15
p13
t14
t15
p14
O
t16
p17
2
t12
2
t17
p12
p16
p18
t13
t18
N
P
t19
p19

Conclusions

+ The procedure does not assume the initial marking to be known, but rather provides the constraints that a initial marking must satisfy to guarantee that the supervisor is effective.
+ The procedure is applicable to Petri nets which may not be repetitive and/or bounded.
+ Since the good markings are characterized as the feasible region of linear inequalities, optimization problems can be solved in those problems in which a part of the marking reflects available resources.
+ The problem which is solved by the procedure cannot be solved with finite automaton based approaches and in general the deadlock problem approached by the latter methods can be approached with our procedure.
+ The procedure makes no assumptions on the structure of the Petri net.
+ An excellent permissivity property has been proved (Theorem 3.)
+ Deadlock prevention is guaranteed by Theorem 2.

Conclusions

+ The procedure allows fully automated implementation (it has been impleemented as a computer program.)
- The supervisor may forbid markings that allow supervision for deadlock-freedom.
- We identified two situations in which the procedure does not terminate.
- The modification of the procedure for guaranteed termination has the disadvantage that it is not guaranteed to satisfy Theorem 2.
- Every iteration may require operations that check the feasibility of an integer program. This may be computationally expensive.
+ All computations are performed off-line. The supervisor is very simple, and so it is a true real time solution.

For More Information ...

The full length paper is available at

http://www.nd.edu/~isis/tech.html

Up to date computer implementation is available at

http://www.nd.edu/~miordach/dp