We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

186,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS

BOOK CITATION INDEX

INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

A Computationally Improved Optimal Solution for Deadlocked Problems of Flexible Manufacturing Systems Using Theory of Regions

Yen-Liang Pan

Additional information is available at the end of the chapter

1. Introduction

While competing for a finite number of resources in a flexible manufacturing system (FMS), e.g., robots and machines, each part has a particular operational flow that determines the order in which such resources are needed. However, such competition for shared resources by concurrent job processes can lead to a system deadlock. It occurs when parts are blocked waiting for shared resources held by others that will never be granted. Its related blocking phenomena often incur unnecessary overhead cost, e.g., a long downtime and low utilization rate of some critical and expensive resources, possibly leading to a catastrophic outcome in some highly automated FMS. Therefore, an efficient deadlock control policy must be developed to ensure that deadlocks do not occur. Having received considerable attention in literature, deadlock is normally prevented by using an offline computational mechanism to control the resource requests in order to avert deadlocks. Fanti and Zhou[1] introduce three fundamental methods (i.e. prevention, detection and avoidance) to solve the deadlock problems. Deadlock prevention aims to impose system constraints to prevent a deadlock. Importantly, deadlock prevention algorithms do not require run-time costs since the problems are solved in system design and planning stages. This study belongs to the deadlock prevention field.

Petri nets (PN)[2] have been recognized as one of the most powerful formal methods for modeling FMS. The reason is that they are well suited to represent such FMS characteristics as precedence relations, concurrence, conflict and synchronization. Their analysis methods used for deadlock prevention in FMS include structural analysis and reachability graphs. Deadlock prevention and avoidance schemes have been developed for controlling FMS[3-8] by using the former. In particular, deadlock prevention problems are solved using the concept

of siphons[3-6]. Li & Zhou propose an elementary siphon control policy (ESCP) to reduce the redundant siphons to obtain structurally simpler controllers[9-10]. However, they cannot obtain optimal ones. Reachability graph methods are used to obtain the live system behavior[11-14]. Without confining to a certain class of FMS, they can provide an optimal deadlock controller by adopting the theory of regions[15]. The theory is originally developed for a transition system (TS). A state-based representation with arcs labeled with symbols from an alphabet of events in a TS can be mapped into a PN model. For an elementary TS (ETS) there exists a PN with minimum transition count (one transition for each label) with a reachability graph isomorphic to the original TS.

Uzam[12] follows the theory of regions[15] to define a deadlock-zone (DZ) and deadlock-free zone (DFZ) for preventing deadlocks. Hence, the concept of DZ and DFZ is used to solve ESSPs. An optimal controller can be obtained but suffers from many redundant control places. Ghaffari *et al.*[13] propose a unique interpretation of the theory of regions and define M_F (*forbidden marking*), M_D (*dangerous marking*), M_L (*legal marking*), and Ω (the set of *marking/transition-separation instances* or MTSI). An optimal PN controller synthesis method for FMS is proposed based on both MTSI and the theory of regions. Unfortunately, redundant MTSIs cannot be entirely avoided for large FMS cases.

To reduce redundant control places, Li *et al.*[16] adopt a combined algorithm based on siphon control and the theory of regions[15]. Its advantage is that the number of separation instances is significantly reduced after some sets of elementary siphons of a system are controlled. However, it fails to determine all sets of MTSIs and its application seems limited to some special nets only.

Uzam and Zhou propose an iterative control policy of liveness enforcement for PNs based on the theory of regions[17]. Less computation is required to obtain a controller. However, as indicated by Li et al[18], it requires the repeated calculation of reachability graphs. Piroddi *et al.* propose a combined selective siphons and critical markings in a reachability graph algorithm to obtain optimal controllers via iterations[19]. They successfully identify the critical uncontrolled siphons and control them to make a deadlock-prone PN live. However, their algorithm also requires the repeated calculation of reachability graphs. Eventually, the controllers are not ordinary (i.e. they contain weighted arcs).

This work in this chapter aims to develop a computationally more efficient optimal deadlock control policy by using the theory of regions. It focuses on dead markings in a reachability graph. The concept of a crucial MTSI (CMTSI) is proposed to synthesize optimal controllers. The proposed method can reduce the computational burden of the MTSI method[13] and redundant control places[12-13]. The experimental results indicate that it is the most efficient policy among all known ones[12-13, 16] that can design optimal controllers.

Section 2 presents the basic definitions and properties of PNs and the theory of regions. Section 3 describes the proposed policy. Section 4 presents the experimental results. Section 5 gives the comparisons. Conclusions are made in Section 6.

2. Preliminaries

2.1. Petri nets[2]

A Petri net (*PN*) is a 5-tuple $N = (P, T, F, W, M_0)$ where P is a finite set of places; T is a finite set of transitions, with $P \cup T \neq \varnothing$ and $P \cap T = \varnothing$; $F \subseteq (P \times T) \cup (T \times P)$ is the set of all directed arcs, $W: (P \times T) \cup (T \times P) \to \boldsymbol{N}$ is the weight function where $\boldsymbol{N} = \{0, 1, 2, \ldots\}$, and $M_0: P \to \boldsymbol{N}$ is the initial marking. A *PN* is said to be ordinary, denoted as (P, T, F), if $\forall f \in F, W(f) = 1$. $[N]^+(p, t) = W(p, t)$ is the input function that means the multiplicity of a directed arc from p to t if $(p, t) \in F$. $[N]^-(p, t) = W(t, p)$ is the output function that means the multiplicity of a directed arc from t to p if $(t, p) \in F$. The set of input (resp., output) transitions of a place p is denoted by $\bullet p$ (resp., $p \bullet$). Similarly, the set of input (resp., output) places of a transition t is denoted by $\bullet t$ (resp., $t \bullet$). A PN structure (P, T, F, W) is denoted by N. A PN with a given initial marking is denoted by (N, M_0).

A PN is said to be pure if no place is both input and output places of the same transition. The so-called incidence matrix $[N]$ of a pure Petri nets is defined as $[N] = [N]^- - [N]^+$. A transition t is said to be enabled at marking M, if $\forall p \in \bullet t, M(p) \geq W(p, t)$, or p is marked with at least $W(p, t)$ tokens, as denoted by $M[t>$. A transition may fire if it is enabled. In an ordinary net, it is enabled iff $\forall p \in \bullet t, M(p) \geq 1$. Firing t at M gives a new marking M' such that $\forall p \in P, M'(p) = M(p) - W(p, t) + W(t, p)$. It is denoted as $M[t>M'$. M indicates the number of tokens in each place, which means the current state of the modeled system. When M_n can be reached from M_0 by firing a sequence of transitions σ, this process is denoted by $M[\sigma > M_n$ and satisfies the *state equation* $M_n = M + [N]\vec{\sigma}$. Here, $\vec{\sigma}$ is a vector of non-negative integers, called a *firing vector*, and $\vec{\sigma}(t)$ indicates the algebraic sum of all occurrences of t in σ. The set of all reachable markings for a PN given M_0 is denoted by $R(N, M_0)$. Additionally, a definition of linearized reachability set (using the state equation) is defined as $\boldsymbol{R}(N, M_0) = \{M: M = M_0 + [N](\bullet \vec{\sigma})\}$. This definition is suitable for the incorporation of the state equation into a set of linear constraints. The markings in $\boldsymbol{R}(N, M_0) - R(N, M_0)$ are called *spurious ones* (with respect to the state equation)[20]. They may also be the solutions of the state equation but not reachable markings. In this work, ones just focus on the reachable markings.

A transition t is said to be *live* if for any $M \in R(N, M_0)$, there exists a sequence of transitions whose firing leads to M' that enables t. A PN is said to be *live* if all the transitions are live. A PN contains a *deadlock* if there is a marking $M \in R(N, M_0)$ at which no transition is enabled. Such a marking is called a dead marking. Deadlock situations are as a result of inappropriate resource allocation policies or exhaustive use of some or all resources. Liveness of a PN means that for each marking $M \in R(N, M_0)$ reachable from M_0, it is finally possible to fire t, $\forall t \in T$ through some firing sequence. (N, M_0) is said to be *reversible*, if $\forall M \in R(N, M_0)$, $M_0 \in R(N, M)$. Thus, in a reversible net it is always possible to go back to initial marking (state) M_0. A marking M' is said to be a *home state*, if for each marking $M \in R(N, M_0)$, 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.

2.2. Theory of regions and synthesis problem[13]

The theory of regions is proposed for the synthesis of pure nets given a finite TS[15], which can be adopted to synthesize the liveness-enforcing net supervisor (LENS) for a plant model[12-13]. For convenience, our method follows the interpretation of the theory of regions in[13].

First of all, let T be a set of transitions and G be a finite directed graph whose arcs are labeled by transitions in T. Assume that there exists a node v in G such that there exists a path from it to any node. The objective of the theory of regions is to find a pure PN (N, M_0), having T as its set of transitions and characterized by its incidence matrix $[N](p, t)$ and its initial marking M_0, such that its reachability graph is G and the marking of node v is M_0. In the following, M denotes both a reachable marking and its corresponding node in G.

Consider any marking M in net (N, M_0). Because (N, M_0) is pure, M can be fully characterized by its corresponding incidence vector $[N](p, \cdot)\vec{\Gamma}_M$ where $\vec{\Gamma}_M$ is the firing vector of path Γ_M. For any transition t that is enabled at M, i.e., t is the label of an outgoing arc of the node M in G

$$M'(p) = M(p) + [N](p, \cdot)\vec{\Gamma}_{M \to M'}, \ \forall (M, M') \in G \land M\,[\,t > M' \tag{1}$$

Consider now any oriented cycle γ of a reachability graph. Applying the state equation to a node in γ and summing them up give the following cycle equation:

$$\sum_{t \in T}[N](p,t)\vec{\gamma}(t) = 0, \forall \gamma \in C \tag{2}$$

where γ is an oriented cycle of G, $\bar{\gamma}(t)$ is a firing vector corresponding to γ, and C is the set of oriented cycles of G.

According to the definition of G, there exists an oriented path Γ_M from M_0 to M. Applying (1) along the path leads to $M(p) = M_0(p) + [N](p,\cdot)\,\vec{\Gamma}_M$. There are several paths from M_0 to M. Under the cycle equations, the product $[N](p,\cdot)\vec{\Gamma}_M$ is the same for all these paths. As a result, Γ_M can be arbitrarily chosen. The reachability of any marking M in G implies that

$$M(p) = M_0(p) + [N](p,\cdot)\vec{\Gamma}_M \geq 0, \forall M \in G \tag{3}$$

The above equation is called the reachability condition. Notably, (3) is necessary but not sufficient. Hence, spurious markings are beyond this paper.

It is clear that the cycle equations and reachability conditions hold for any place p. For each pair (M, t) such that M is a reachable marking of G and t is a transition not enabled at M, t should be prevented from happening by some place p. Since the net is pure, t is prevented from happening at M by a place p iff

$$M_0(p) + [N](p,\cdot)\vec{\Gamma}_M + [N](p,t) \leq -1 \tag{4}$$

The above equation (4) is called the event separation condition of (M, t). The set of all possible pairs (M, t) where M is a reachable marking and t is not enabled at M is called the *set of event separation instances* or *marking/transitions-separation instances* (MTSI)[13]. Symbol Ω is used to represent the set of MTSI in this paper. To solve the control problem, Ω is identified. The corresponding control places can then be found to prevent the transitions of the controlled system from firing in order to keep all legal markings only.

3. Controller synthesis method

In this section, an efficient controller synthesis method is developed based on the theory of regions. Please note that all transitions of the PN models are regarded as controllable ones.

3.1. Supervisory control problem

It is assumed that a deadlock-prone PN model contains at least a dead marking in its reachability graph at which no transition is enabled. Its reachability graph contains dead and live zones. Consequently, this study attempts to propose a method to prevent the controlled systems from entering a dead zone/marking.

A dead marking cannot enable any transition and thus cannot go to any other markings. We can formally define the dead marking M_D as follows.

Definition 1: The set of *dead markings* $M_D = \{M \in R(N, M_0) |$ at M, no transition is enabled$\}$.

Definition 2: A zone consisting of all dead markings is called a *dead zone*, denoted by Z_D.

Once a marking enters a dead zone, the system is dead. If there is no dead zone in a reachability graph, the system is called a live one.

The goal of the work is to control a deadlock-prone system such that it is live. All markings of a reachability graph can be divided into three groups: legal markings (M_L), quasi-dead markings (M_Q), and dead markings (M_D).

Definition 3: The set of *quasi-dead markings* $M_Q = \{M \in R(N, M_0) | M$ must eventually evolve to a dead one regardless of transition firing sequences$\}$.

Definition 4: A zone consisting of all quasi-dead markings is called a *quasi-dead zone*, denoted by Z_Q.

Definition 5: A zone consisting of all quasi-dead and dead markings, i.e., $Z_I = Z_D \cup Z_Q$, is called an *illegal zone*.

Markings except quasi-dead and dead markings are legal ones. Once a legal marking is enforced into the illegal zone, the net will eventually become deadlock.

Definition 6: A zone consisting of all legal markings is called a *legal zone*, i.e., $Z_L = R(N, M_0) - Z_I$.

Ramadge and Wonham show that a system has the maximally permissive behavior if the system behavior equals Z_L[21]. In other words, one must remove all the markings in illegal

zone (i.e. quasi-dead and dead markings) from $R(N, M_0)$ if one wants to obtain the maximally permissive behavior. Ghaffari *et al.* propose the MTSI method to achieve their deadlock prevention based on the theory of regions[13]. However, the set of all MTSIs from the reachability graph must be identified. As a result, we can conclude that their method is computationally inefficient. A more efficient method is thus needed as described next.

3.2. Crucial MTSI (CMTSI)

Two types of CMTSIs are defined as follows.

Definition 7: Type I CMTSI: $\Omega' = \{(M, t) | M \in M_L, t \in T$, and $\exists M' \in M_D, M'' \in M_L$, and $t' \in T$ such that $M[t > M'$ and $M[t' > M'']\}$. Denote the set of all the dead markings related to Ω' as M'_D, i.e., $M'_D = \{M' \in M_D | \exists (M, t) \in \Omega'$ such that $M[t > M'\}$. They are called type I deadlocks.

Definition 7 explains a legal marking that can evolve into a dead or legal zone as shown in Figure 1 through a single transition's firing. For those dead markings that are not type I deadlocks, we need to introduce Type II CMTSI and deadlocks.

Definition 8: A zone consisting of all type I deadlocks (M'_D) is called type I dead zone, denoted by Z'.

Definition 9: σ_k is defined as a transition firing sequence starting in a quasi-dead marking (M_Q) and ending in a deadlock marking in M_D where $i = |\sigma_k|$ is the number of transitions in σ_k, called its length. Denote a firing sequence with the shortest length (i.e., smallest i) from any quasi-dead marking to M' as $\sigma^*(M')$ given $M' \in M_D - M'_D$.

Definition 10: Type II CMTSI : $\Omega'' = \{(M, t) | M \in M_L, t \in T$, and $\exists M' \in M_Q, M'' \in M_L, M''' \in M_D, t' \in T$, and a firing sequence $\sigma = \sigma^*(M''')$ from M' to M''' such that $M[t > M', M[t' > M'']$, and $M'[\sigma > M''']\}$. The set of dead markings associated with Type II CMTSI is denoted as M''_D, called type II deadlocks. $M''_D = \{M''' \in M_D | \exists (M, t) \in \Omega'', M' \in M_Q$ and a firing sequence σ from M' to M''' such that $M[t > M'$ and $\sigma = \sigma^*(M''')\}$.
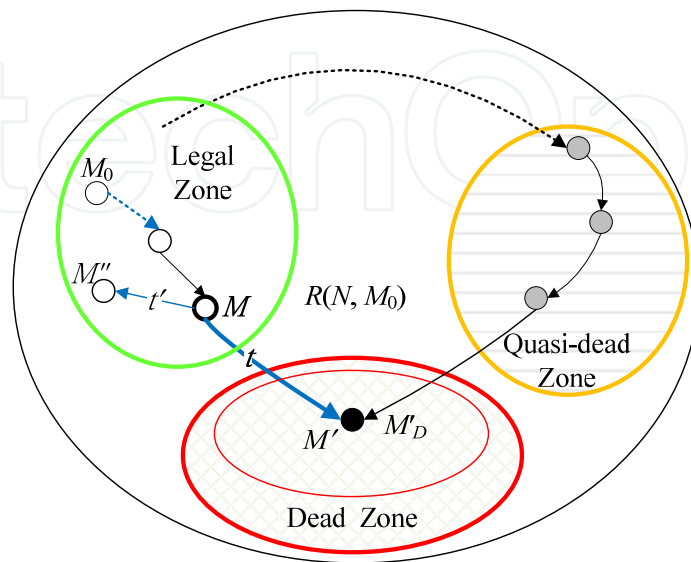

Figure 1. A structure of Type I CMTSI.

Definition 11: A zone consisting of all type II deadlocks (M''_D) is called type II dead zone, Z''.

A Type II CMTSI contains a legal marking that cannot reach a dead marking with one single transition's firing as shown in Figure 2. Given a dead marking in M''_D, the shortest transition firing sequence needs to be found. The main reason is based on the fact that, for a dead marking, the length of the firing sequence from the initial marking to CMTSI is the longest path than those from the initial marking to MTSIs. Hence, the solutions of MTSIs will be totally covered by the solution of CMTSI. For example, as shown in Figure 3, σ^* is the shorter path since $|\sigma^*| < |\sigma'|$ (i.e. $|\sigma^*| = 1$ and $|\sigma'| = 3$).

Remark 1: A dead marking is always with its corresponding CMTSI. As a result, the corresponding CMTSI is of either Type I or II. Type I may be viewed as a special case of Type II CMTSI by defining $\sigma^* = 0$ (no need to enter Z_Q but directly to Z_D). Type I CMTSI will be processed first in our proposed method. In the following, Theorems 1-3 will help readers to understand how to choose CMTSIs, which are with the same firing sequence of legal markings, from Types I and II.

Theorem 1: If a dead marking $M \in M'_D$ is associated with two different CMTSIs, only one CMTSI needs to be controlled.

Proof: Assume that a dead marking M is with both CMTSIs $\{M_i, t_m\}$ and $\{M_j, t_n\}$ as shown in Figure 4. According to the state equation, $M_i + [N](\bullet t_m) = M_j + [N](\bullet t_n) = M$. Arranging the above equation, $M_0 + [N](\bullet \vec{\sigma}_{M_0 \to M_i}) + [N](\bullet t_m) = M_0 + [N](\bullet \vec{\sigma}_{M_0 \to M_j}) + [N](\bullet t_n)$. According to (4), realizing either CMTSI, e.g., $\{M_i, t_m\}$, leads to $M_0 + [N](\bullet \vec{\sigma}_{M_0 \to M_i}) + [N](\bullet t_m) \leq -1$, which in turn implies $M_0 + [N](\bullet \vec{\sigma}_{M_0 \to M_j}) + [N](\bullet t_n) \leq -1$ and vice versa. Hence, only one CMTSI needs to be controlled.

Remark 2: Based on Theorem 1, if a dead marking $M \in M'_D$ is associated with more than two CMTSIs, only one of them needs to be controlled.
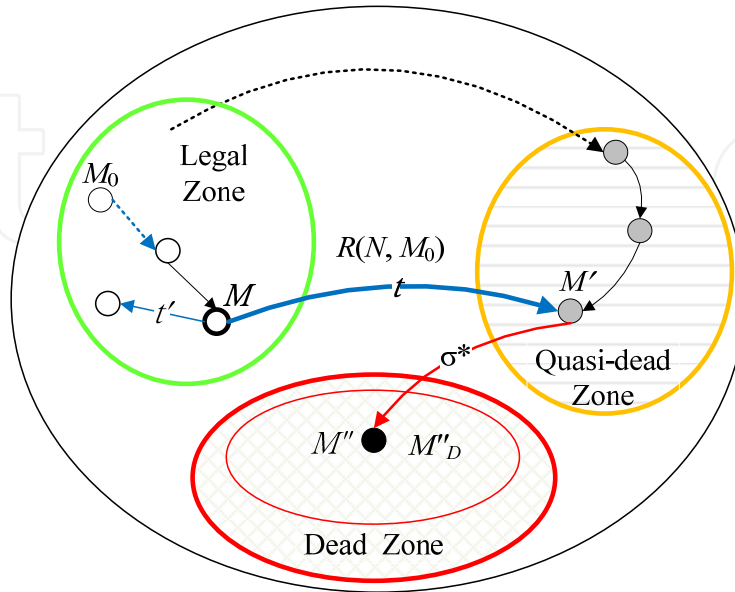

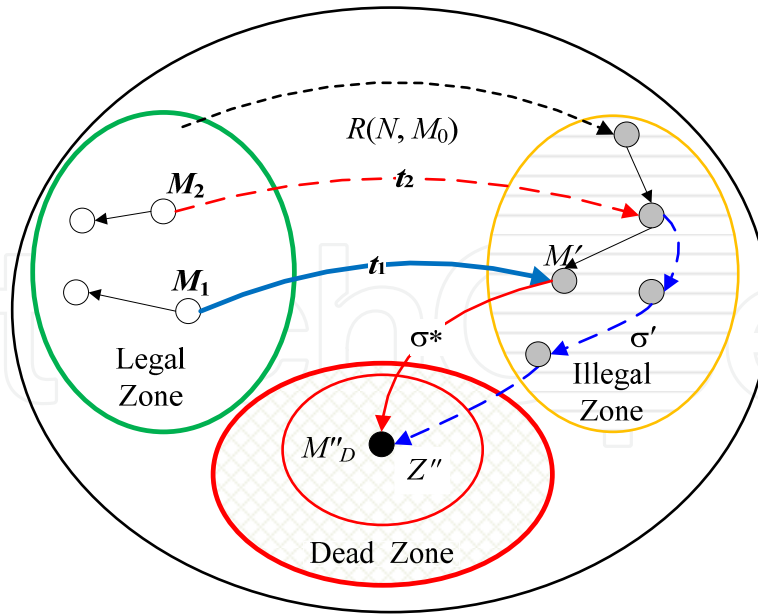**Figure 2.** A structure of Type II CMTSI.

Figure 3. The shorter path σ* in Type II CMTSI given a dead marking.

Theorem 2: If a dead marking $M \in M''_D$, is associated with two CMTSIs whose markings can reach a same quasi-dead marking M' via their respective single transition's firing, only one CMTSI needs to be controlled.

Proof: Assume that a dead marking M is associated with Type II CMTSIs $\{M_p, t_r\}$ and $\{M_q, t_s\}$. M_p and M_q reaches a quasi-dead markings M' via t_r and t_s's firing, respectively as shown in Figure 5.

According to the state equation, $M_p + [N](\bullet t_r) + [N](\bullet \vec{\sigma}^*) = M_q + [N](\bullet t_s) + [N](\bullet \vec{\sigma}^*)$ M. Arranging the above equation, one can realize that $M_p + [N](\bullet t_r) = M_q + [N](\bullet t_s)$. According to (4), realizing either CMTSI, e.g., $\{M_p, t_r\}$, leads to $M_0 + [N] (\bullet \vec{\sigma}_{M_0 \to M_p}) + [N](\bullet t_r) \leq -1$, which in turn implies $M_0 + [N] (\bullet \vec{\sigma}_{M_0 \to M_q}) + [N](\bullet t_s) \leq -1$ and vice versa. Hence, only one CMTSI needs to be controlled.

Theorem 3: A dead marking $M \in M''_D$, is associated with two CMTSIs whose markings can reach two different quasi-dead markings M'_p and M'_q via two different single transitions' firing. Both need to be controlled if $[N](\bullet \vec{\sigma}_r^*) \neq [N](\bullet \vec{\sigma}_s^*)$.

Proof: Assume that a dead marking M is associated with both Type II CMTSIs $\{M_p, t_r\}$ and $\{M_q, t_s\}$. M_p and M_q reach two different quasi-dead markings M'_p and M'_q via t_r and t_s's firing, respectively as shown in Figure 6.

According to the state equation, $M_p + [N](\bullet t_r) + [N](\bullet \vec{\sigma}_r^*) = M_q + [N](\bullet t_s) + [N](\bullet \vec{\sigma}_s^*) = M''_D$. Arranging the above equation, $M'_p + [N](\bullet \vec{\sigma}_r^*) = M'_q + [N](\bullet \vec{\sigma}_s^*)$. Since $[N](\bullet \vec{\sigma}_r^*) \neq [N](\bullet \vec{\sigma}_s^*)$, M'_p is not equal to M'_q. And also according to the definition of the event separation condition equation, the first set of CMTSI $\{M_p, t_r\}$ leads to the first event separation condition equation is $M_0 + [N](\bullet \vec{\sigma}_{M_0 \to M_p}) + [N](\bullet t_r) \leq -1$; and the second set of CMTSI $\{M_q, t_s\}$ leads to the another event separation condition equation is $M_0 + [N](\bullet \vec{\sigma}_{M_0 \to M_q}) + [N](\bullet t_s) \leq -1$. Hence,

one can infer that M'_p and M'_q are two different quasi-dead markings if $[N](\bullet \vec{\sigma}_r^{\,*}) \neq [N](\bullet \vec{\sigma}_s^{\,*})$. It hints the two event separation condition equations are different. As a result, both CMTSIs need to be controlled.
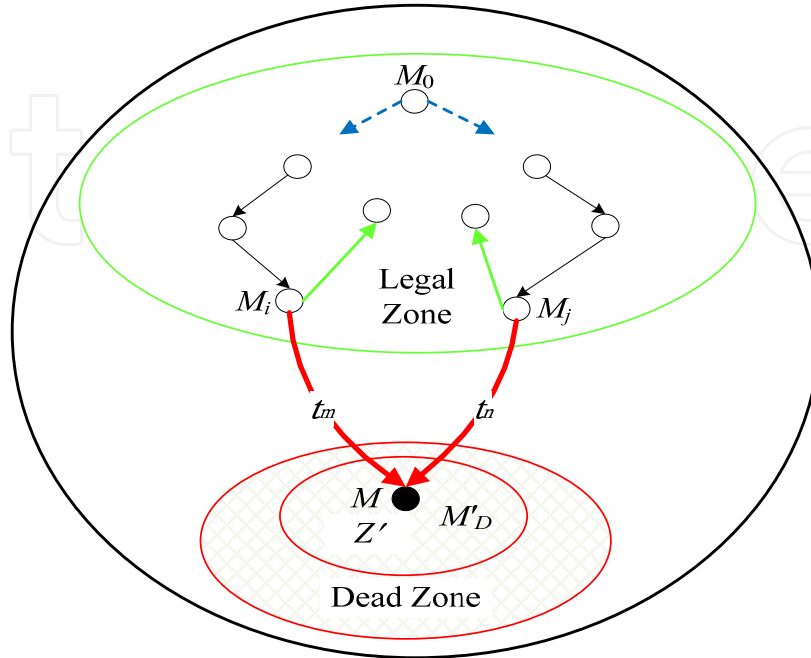

Figure 4. A type I deadlocks associated with two CMTSIs.

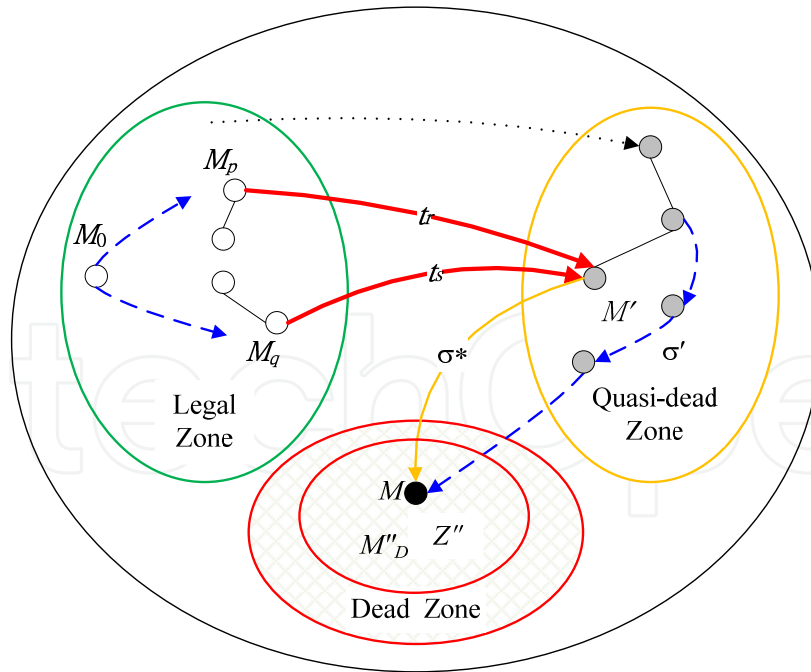**Figure 5.** Two CMTSIs connected to the same quasi-dead marking.

Definition 12: A legal marking $M \in M_L$ can be led to a quasi-dead marking M_q via a single transition firing. M_q must eventually evolve to a dead one M_d (i.e. $M_d \in M_D$) after a sequence $\sigma_n = t_1 t_2 \ldots t_n$ fires. Denote the set of all the markings on the path from M_q to M_d as M_{q-d}.
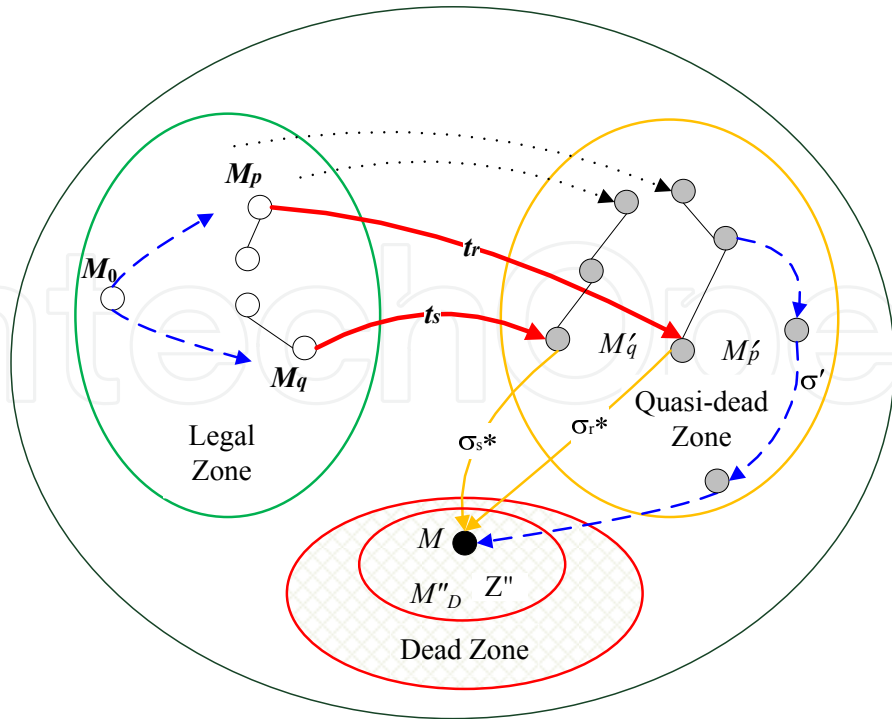
Figure 6. Two CMTSIs connected to two quasi-dead markings.

Remark 3: Based on Theorem 3, both CMTSIs still need to be controlled even if $M_P = M_q$ for the case shown in Figure 7.
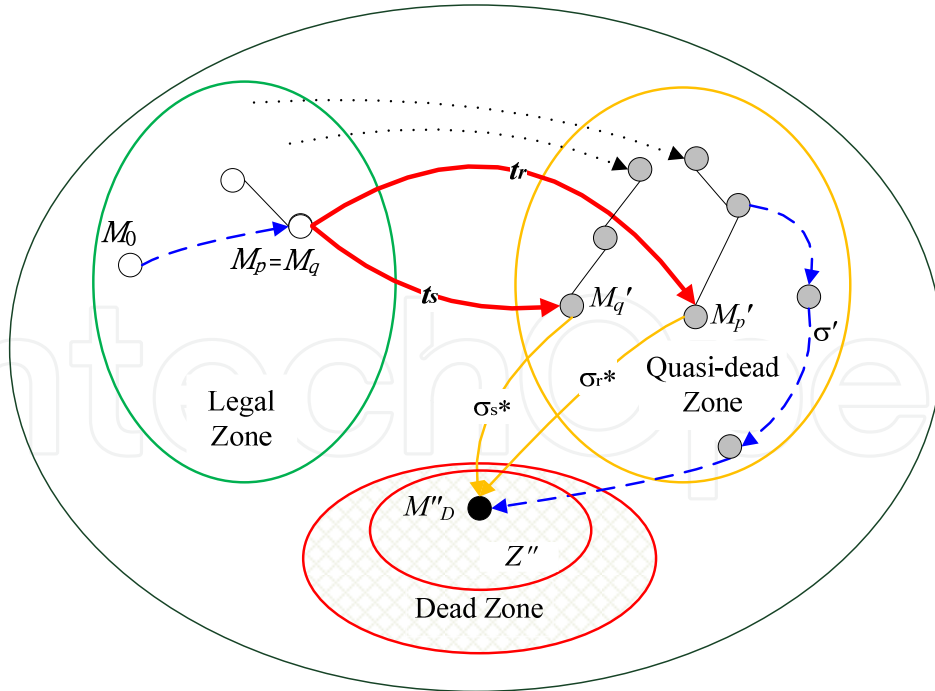

Figure 7. Two CMTSIs connected to two quasi-dead markings M'_p and M'_q with $\sigma_r{}^* = \sigma_s{}^* = \sigma^*(M''_D)$.

Control places are then found after CMTSIs. They are used to keep all markings of the controlled system within the legal zone.

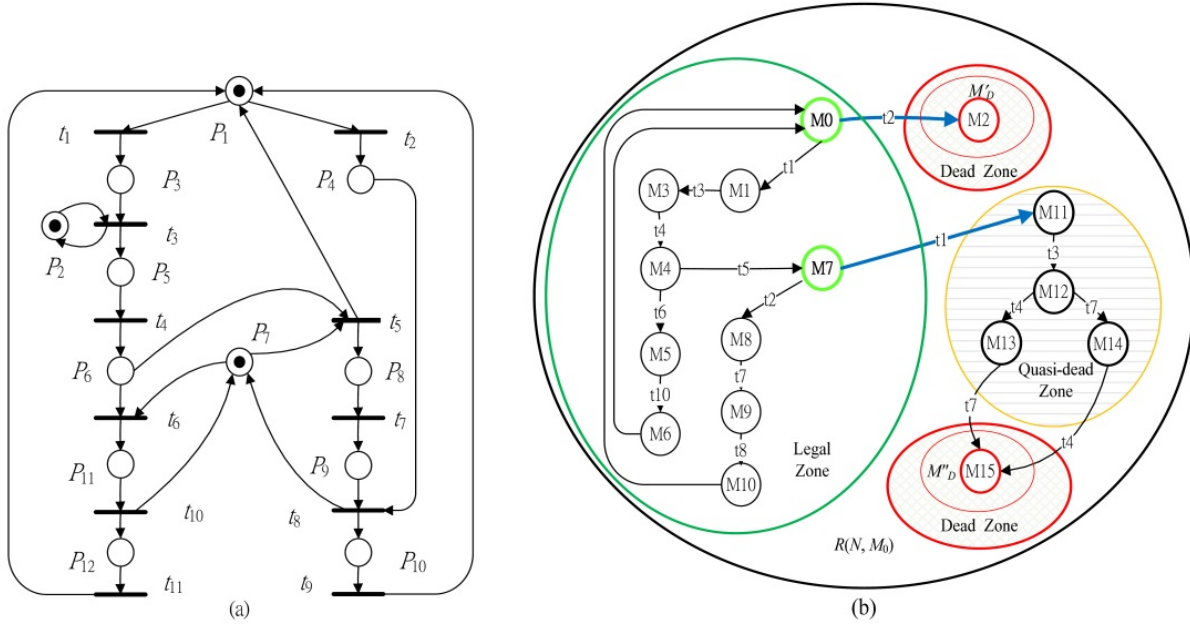*Theorem 4*: $(\Omega' \cup \Omega'') \subseteq \Omega$

Figure 8. (a) A Petri net model.[22] (b) Its reachability graph.

Here, Figure 8(a) taken from existing literatures[22] is used to demonstrate how to identify two types of CMTSIs from its reachability graph (i.e. Figure 8(b)). Assume that all transitions of PN models are immediately in this case. Therefore, one can easy identify there are two dead markings M'_D (i.e. M_2) and M''_D (i.e. M_{15}) and four quasi-dead markings (i.e. M_{11}, M_{12}, M_{13} and M_{14}). Additionally, the markings M_0, M_1, M_3-M_{10} are the legal markings. Based on the mentioned above, there are two sets of CTMSIs in the reachability graph system due to the two dead markings in the system. As a result, one can infer that $\{M_0, t_2\}$ belongs to type I CMTSI and $\{M_7, t_1\}$ belongs to type II. In this Petri net system model, there are only one type I CMTSI and only one type II.

3.3. Procedure of deadlock prevention policy

Next, quasi-dead, dead, and legal markings are identified. Based on[12-13], the maximally permissive behavior means all of legal markings (M_L) and the number of reachability condition equations equals $|M_L|$. Additionally, all CMTSIs can be obtained such that the legal markings do not proceed into the illegal zone. The proposed deadlock prevention algorithm is constructed as Figure 9.

Theorem 5: The proposed deadlock prevention policy is more efficient than the method proposed by Ghaffari et al.[13]

Proof: The theory of regions is used to prevent the system deadlocks by both our deadlock prevention policy and the conventional one. All MTSIs can be controlled by the two control policies. Since $(\Omega' \cup \Omega'') \subseteq \Omega$, the use of CMTSI can more efficiently handle the synthesis problem than that of MTSI[13].
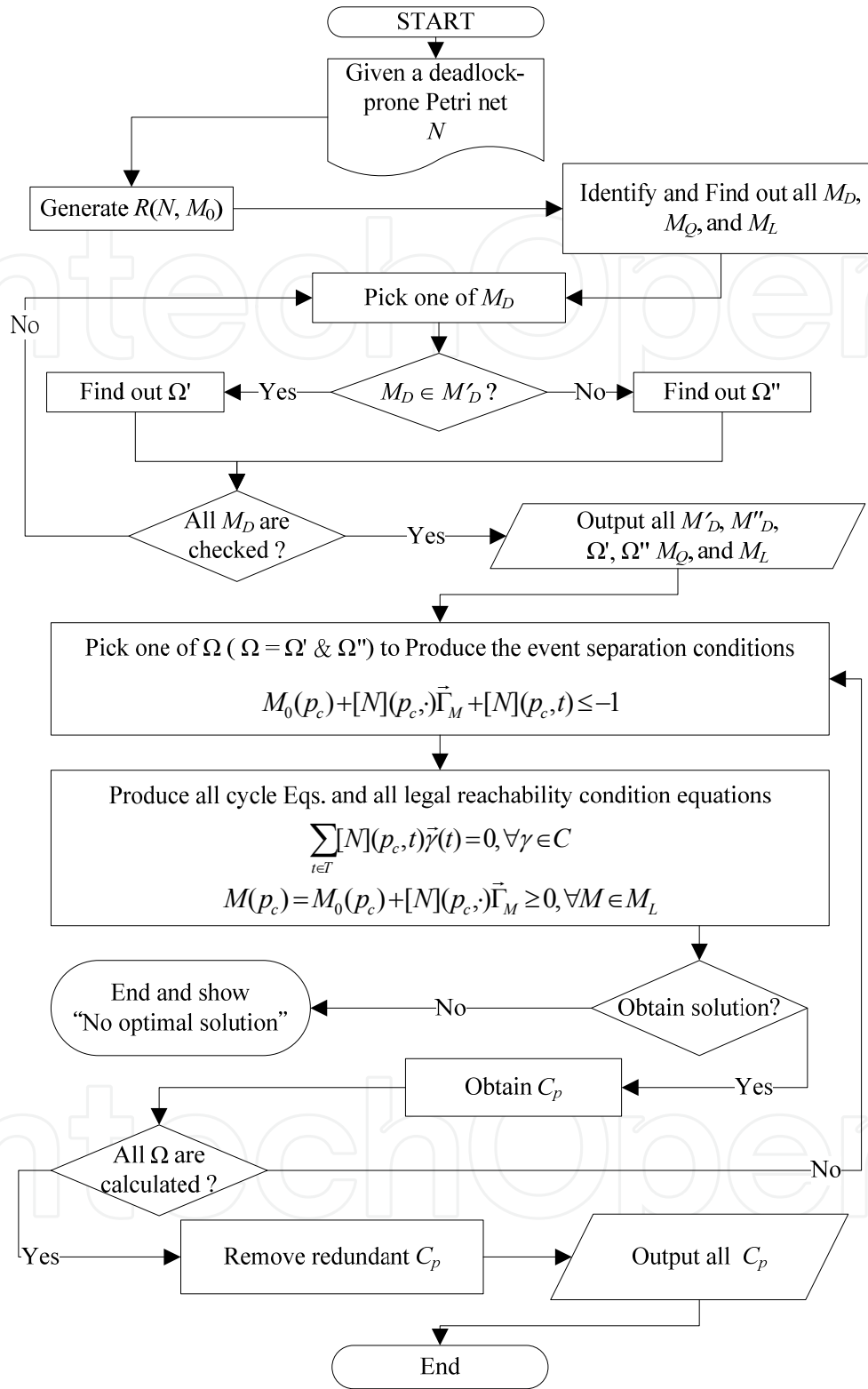
Figure 9. The Proposed Deadlock Prevention Flowchart.

4. Experimental results

Two FMS examples are used to evaluate our deadlock prevention policy[12-23].
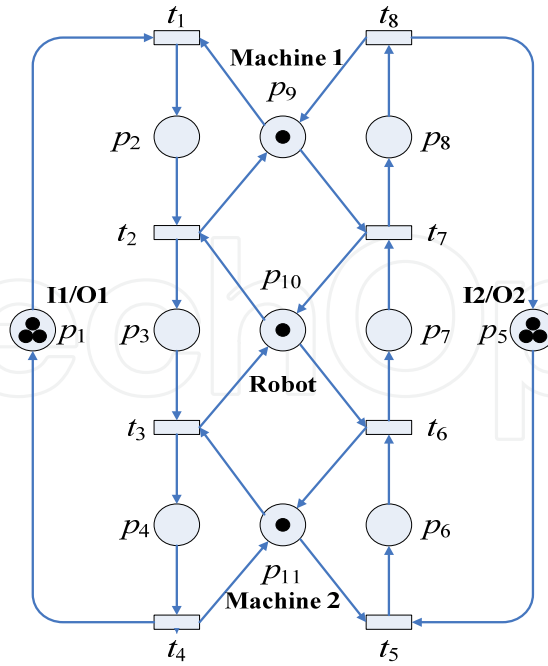
Figure 10. An FMS PN Model[12].

Example I: An FMS is shown in Figure 10[12]. This PN is a system of simple sequential processes with resources (S³PR), denoted by (N_1, M_0). To do our deadlock prevention policy, $R(N_1, M_0)$ of the PN system can be constructed as shown in Figure 11.

Two dead markings (i.e. M_7 and M_{12}) can then be identified. Next, $\Omega'_1 = \{(M_3, t_5)\}$ and $\Omega'_2 = \{(M_{15}, t_1)\}$ are obtained. The event separation condition equations can be obtained through them as follows.

$$M_7(p_c) = M_0(p_c) + 2[N](p_c, t_1) + [N](p_c, t_2) + [N](p_c, t_5) \leq -1 \tag{5}$$

$$M_{12}(p_c) = M_0(p_c) + [N](p_c, t_1) + 2[N](p_c, t_5) + [N](p_c, t_6) \leq -1 \tag{6}$$

Figure 11. $R(N, M_0)$ of Example I.

Two cycle equations are as follows.

$$[N](p_c, t_1) + [N](p_c, t_2) + [N](p_c, t_3) + [N](p_c, t_4) = 0 \qquad (7)$$

$$[N](p_c, t_5) + [N](p_c, t_6) + [N](p_c, t_7) + [N](p_c, t_8) = 0 \qquad (8)$$

After listing all reachability conditions, all legal markings can be determined. In detail, M_0-M_6, M_8, and M_{13}-M_{19} are legal. Hence, the following reachability conditions are obtained.

$$M_0(p_c) \geq 0 \qquad (9)$$

$$M_1(p_c) = M_0(p_c) + [N](p_c, t_1) \geq 0 \qquad (10)$$

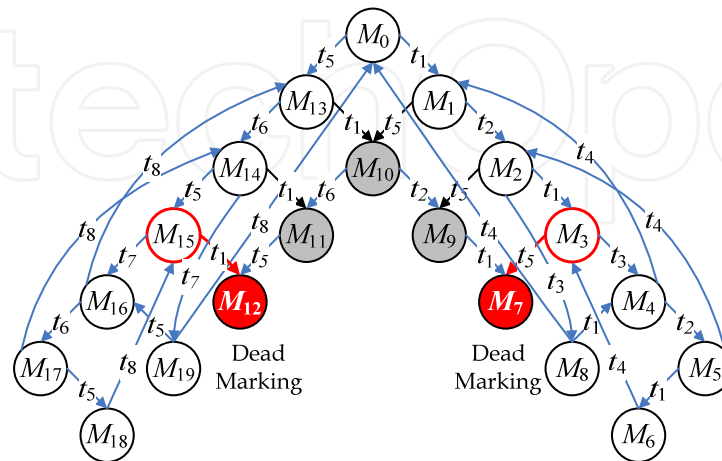$$M_2(p_c) = M_0(p_c) + [N](p_c, t_1) + [N](p_c, t_2) \geq 0 \qquad (11)$$

$$M_3(p_c) = M_0(p_c) + 2[N](p_c, t_1) + [N](p_c, t_2) \geq 0 \qquad (12)$$

$$M_4(p_c) = M_0(p_c) + 2[N](p_c, t_1) + [N](p_c, t_2) + [N](p_c, t_3) \geq 0 \qquad (13)$$

$$M_5(p_c) = M_0(p_c) + 2[N](p_c, t_1) + 2[N](p_c, t_2) + [N](p_c, t_3) \geq 0 \qquad (14)$$

$$M_6(p_c) = M_0(p_c) + 3[N](p_c, t_1) + 2[N](p_c, t_2) + [N](p_c, t_3) \geq 0 \qquad (15)$$

$$M_8(p_c) = M_0(p_c) + [N](p_c, t_1) + [N](p_c, t_2) + [N](p_c, t_3) \geq 0 \qquad (16)$$

$$M_{13}(p_c) = M_0(p_c) + [N](p_c, t_5) \geq 0 \qquad (17)$$

$$M_{14}(p_c) = M_0(p_c) + [N](p_c, t_5) + [N](p_c, t_6) \geq 0 \qquad (18)$$

$$M_{15}(p_c) = M_0(p_c) + 2[N](p_c, t_5) + [N](p_c, t_6) \geq 0 \qquad (19)$$

$$M_{16}(p_c) = M_0(p_c) + 2[N](p_c, t_5) + [N](p_c, t_6) + [N](p_c, t_7) \geq 0 \qquad (20)$$

$$M_{17}(p_c) = M_0(p_c) + 2[N](p_c, t_5) + 2[N](p_c, t_6) + [N](p_c, t_7) \geq 0 \qquad (21)$$

$$M_{18}(p_c) = M_0(p_c) + 3[N](p_c, t_5) + 2[N](p_c, t_6) + [N](p_c, t_7) \geq 0 \qquad (22)$$

$$M_{19}(p_c) = M_0(p_c) + [N](p_c, t_5) + [N](p_c, t_6) + [N](p_c, t_7) \geq 0 \qquad (23)$$

Furthermore, two optimal control places C_{P1} and C_{P2} can be obtained when (5) and (7)-(23) are solved. Their detailed information is: $M_0(C_{P1}) = 1$, $t_1 = t_5 = -1$, $t_2 = t_6 = 1$, $t_3 = t_4 = t_7 = t_8 = 0$; and $M_0(C_{P2}) = 1$, $t_2 = t_5 = -1$, $t_3 = t_6 = 1$, $t_1 = t_4 = t_7 = t_8 = 0$. By the same way, using (6) and (7)-(23), one can find two optimal control places, C_{p3} and C_{p4}. $M_0(C_{P3}) = 1$, $t_1 = t_6 = -1$, $t_2 = t_7 = 1$, $t_3 = t_4 = t_5 = t_8 = 0$; and $M_0(C_{P4}) = 1$, $t_1 = t_5 = -1$, $t_2 = t_6 = 1$, $t_3 = t_4 = t_7 = t_8 = 0$. Notably, C_{P1} and C_{P4} are the same. Therefore, a redundant control place (C_{P4}) can be removed. As a result, the system net can be controlled with the three control places C_{P1}, C_{P2} and C_{P3}. The optimally controlled system net (N_{1H}, M_0) is obtained as shown in Table 1.

It is worthy to emphasize that the three control places are obtained by using two CMTSIs and 36 equations under our control policy. However, six MTSIs/ESSPs and 108 equations have to be solved in two existing literatures[12, 24].

Additional Control Places	$M_0(C_{pi})$	$\bullet(Cp)$	$(Cp)\bullet$
Cp_1	1	t_2, t_6	t_1, t_5
Cp_2	1	t_3, t_6	t_2, t_5
Cp_3	1	t_2, t_7	t_1, t_6

Table 1. Control Places of the Net (N_{1H}, M_0)

Example II: This example is taken from[23] and is used in [12, 16, 24]. Here, the PN model of the system, denoted as (N_2, M_0), is shown in Figure 12.
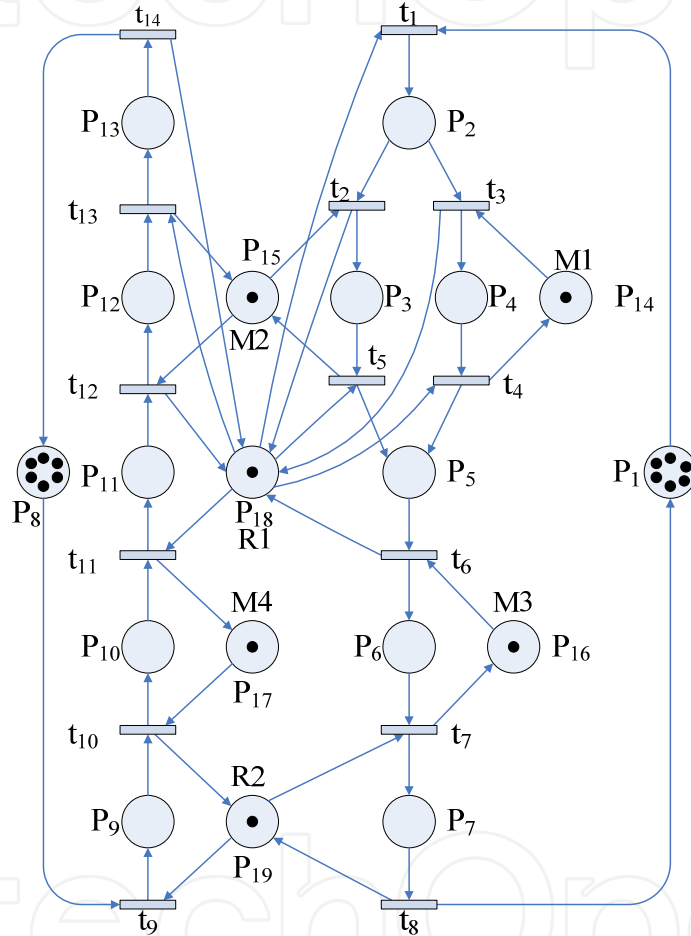
Figure 12. The Petri nets model of example II.

To prevent deadlock, 282 reachable markings (M_1 to M_{282}) are identified according to the software INA[25]. 16 dead markings M_9, M_{19}, M_{70}, M_{71}, M_{76}, M_{77}, M_{78}, M_{83}, M_{84}, M_{94}, M_{99}, M_{100}, M_{105}, M_{106}, M_{112}, and M_{113} are then located. Next, 61 quasi-dead markings M_6, M_7, M_8, M_{14}, M_{15}, M_{16}, M_{17}, M_{18}, M_{25}, M_{48}, M_{66}, M_{67}, M_{68}, M_{69}, M_{72}, M_{73}, M_{74}, M_{75}, M_{79}, M_{80}, M_{81}, M_{82}, M_{87}, M_{101}, M_{102}, M_{103}, M_{104}, M_{108}, M_{109}, M_{110}, M_{111}, M_{124}, M_{130}, M_{135}, M_{136}, M_{141}, M_{142}, M_{143}, M_{150}, M_{162}, M_{198}, M_{203}, M_{204}, M_{210}, M_{250}, M_{251}, M_{257}, M_{258}, M_{259}, M_{260}, M_{261}, M_{262}, M_{263}, M_{265}, M_{269}, M_{270}, M_{272}, M_{273}, M_{274}, M_{277}, and M_{278} are found based on Definition 3 in this paper. Hence, the number of legal markings (i.e. 288 – (16 + 61) = 205) can be determined. Type I and II CMTSIs can be obtained as shown in Table 2. Notice that {(M_{56}, t_9)} in M_{77} is a redundant one.

M'_D	Ω'_C	M''_D	Ω''_C
M_9	$\{(M_{65}, t_1)\}$	M_{19}	$\{(M_{56}, t_1)\}$
			$\{(M_{56}, t_9)\}$
M_{70}	$\{(M_{43}, t_9)\}$	M_{76}	$\{(M_{65}, t_{11})\}$
M_{71}	$\{(M_{51}, t_9)\}$	M_{77}	$\{(M_{56}, t_9)\}$
			$\{(M_{56}, t_{11})\}$
M_{78}	$\{(M_{171}, t_9)\}$	M_{83}	$\{(M_{128}, t_{11})\}$
M_{94}	$\{(M_{93}, t_1)\}$	M_{84}	$\{(M_{60}, t_9)\}$
			$\{(M_{60}, t_{11})\}$
M_{99}	$\{(M_{98}, t_1)\}$	M_{105}	$\{(M_{93}, t_{11})\}$
M_{100}	$\{(M_{98}, t_4)\}$	M_{106}	$\{(M_{98}, t_{11})\}$
		M_{112}	$\{(M_{122}, t_{11})\}$
		M_{113}	$\{(M_{96}, t_{11})\}$

Table 2. The Dead Markings and Their Relative CMTSIs.

M'_D	Event Separation Condition Equations
M_9	$M_0 + 3t_1 + t_2 + t_3 + 2t_9 + t_{10} \leq -1$
M_{70}	$M_0 + 3t_1 + t_2 + 2t_3 + 2t_4 + t_6 + 2t_9 + t_{10} \leq -1$
	$M_0 + 3t_1 + 2t_2 + t_3 + t_4 + t_5 + t_6 + 2t_9 + t_{10} \leq -1$
M_{71}	$M_0 + 3t_1 + t_2 + 2t_3 + t_4 + t_5 + t_6 + 2t_9 + t_{10} \leq -1$
	$M_0 + 3t_1 + 2t_2 + t_3 + 2t_5 + t_6 + 2t_9 + t_{10} \leq -1$
M_{78}	$M_0 + 2t_1 + t_2 + t_3 + t_4 + t_5 + t_6 + 2t_9 + t_{10} \leq -1$
	$M_0 + 2t_1 + 2t_2 + 2t_5 + t_6 + 2t_9 + t_{10} \leq -1$
	$M_0 + 2t_1 + 2t_3 + 2t_4 + t_6 + 2t_9 + t_{10} \leq -1$
M_{94}	$M_0 + 2t_1 + t_3 + 3t_9 + 2t_{10} + t_{11} + t_{12} \leq -1$
M_{99}	$M_0 + 3t_1 + 2t_3 + t_4 + t_6 + 3t_9 + 2t_{10} + t_{11} + t_{12} \leq -1$
	$M_0 + 3t_1 + t_2 + t_3 + t_5 + t_6 + 3t_9 + 2t_{10} + t_{11} + t_{12} \leq -1$

Table 3. The Dead Markings and Relative Event Separation Condition Equations of Type I CMTSI.

Tables 3-4 show the event separation condition equations based on 18 CMTSIs. Here, the procedure of our method is introduced as follows. Due to the space limitation, we use only one example (i.e. dead marking M_9) to illustrate how to prevent legal markings from leading to dead one M_9 by using a CMTSI. To do so, $\Omega'_{C1} = \{(M_{65}, t_1)\}$ can be located due to M_9. The event separation condition equation can then be identified as follows.

$$M_9(C_{P1}) = M_0(C_{P1}) + 3t_1 + t_2 + t_3 + 2t_9 + t_{10} \leq -1 \tag{24}$$

Next, three different cycle equations are:

$$t_1 + t_2 + t_5 + t_6 + t_7 + t_8 = 0 \tag{25}$$

$$t_1 + t_3 + t_4 + t_6 + t_7 + t_8 = 0 \tag{26}$$

$$t_9 + t_{10} + t_{11} + t_{12} + t_{13} + t_{14} = 0 \tag{27}$$

Finally, 205 reachability condition equations can be listed. They represent the sequence of all legal markings from the initial one. Moreover, control place C_{P1} can be computed, i.e., M_0 (C_{P1}) = 2, $\bullet C_{P1}$ = {t_6, t_{13}}, and $C_{P1}\bullet$ = {t_1, t_{11}}. Similarly, other control places are obtained as shown in Tables 5-6.

M''_D	Event Separation Condition Equations
M_{19}	$M_0 + 4t_1 + t_2 + 2t_3 + t_4 + t_6 + t_9 + t_{10} \leq -1$
	$M_0 + 4t_1 + 2t_2 + t_3 + t_5 + t_6 + t_9 + t_{10} \leq -1$
	$M_0 + 3t_1 + t_2 + 2t_3 + t_4 + t_6 + 2t_9 + t_{10} \leq -1$
	$M_0 + 3t_1 + 2t_2 + 2t_3 + t_5 + t_6 + 2t_9 + t_{10} \leq -1$
M_{76}	$M_0 + 2t_1 + t_2 + t_3 + 2t_9 + t_{10} + t_{11} \leq -1$
M_{77}	$M_0 + 3t_1 + t_2 + 2t_3 + t_4 + t_6 + t_9 + t_{10} + t_{11} \leq -1$
	$M_0 + 3t_1 + 2t_2 + t_3 + t_5 + t_6 + t_9 + t_{10} + t_{11} \leq -1$
M_{83}	$M_0 + t_1 + t_2 + 2t_9 + t_{10} + t_{11} \leq -1$
M_{84}	$M_0 + 2t_1 + t_2 + t_3 + t_4 + t_6 + t_9 + t_{10} + t_{11} \leq -1$
	$M_0 + 2t_1 + 2t_2 + t_5 + t_6 + t_9 + t_{10} + t_{11} \leq -1$
	$M_0 + 3t_1 + t_2 + 2t_3 + t_4 + t_6 + 2t_9 + t_{10} \leq -1$
	$M_0 + 3t_1 + 2t_2 + t_3 + t_5 + t_6 + 2t_9 + t_{10} \leq -1$
M_{105}	$M_0 + t_1 + t_3 + 3t_9 + 2t_{10} + 2t_{11} + t_{12} \leq -1$
M_{106}	$M_0 + 2t_1 + 2t_3 + t_4 + t_6 + 3t_9 + 2t_{10} + 2t_{11} + t_{12} \leq -1$
	$M_0 + 2t_1 + t_2 + t_3 + t_5 + t_6 + 3t_9 + 2t_{10} + 2t_{11} + t_{12} \leq -1$
M_{112}	$M_0 + 3t_9 + 2t_{10} + 2t_{11} + t_{12} \leq -1$
M_{113}	$M_0 + t_1 + t_3 + t_4 + t_6 + 3t_9 + 2t_{10} + 2t_{11} + t_{12} \leq -1$
	$M_0 + t_1 + t_2 + t_5 + t_6 + 3t_9 + 2t_{10} + 2t_{11} + t_{12} \leq -1$

Table 4. The Dead Markings and Relative Event Separation Condition Equations of Type II CMTSI.

M'_D	Ω'_C	$M_0(C_{pi})$	$\bullet(C_{pi})$	$(C_{pi})\bullet$
M_9	{(M_{65}, t_1)}	2	t_6, t_{13}	t_1, t_{11}
M_{70}	{(M_{43}, t_9)}	3	t_6, t_{11}	t_2, t_4, t_9
		3	t_5, t_7, t_{11}	t_2, t_6, t_9
		3	t_7, t_{11}	t_4, t_5, t_9
M_{71}	{(M_{51}, t_9)}	3	t_7, t_{11}	t_4, t_5, t_9
M_{78}	{(M_{171}, t_9)}	3	t_7, t_{11}	t_4, t_5, t_9
M_{94}	{(M_{93}, t_1)}	2	t_6, t_{13}	t_1, t_{11}
M_{99}	{(M_{98}, t_1)}	4	t_7, t_{11}	t_1, t_9
M_{100}	{(M_{98}, t_4)}	3	t_7, t_{11}	t_4, t_5, t_9

Table 5. Control Places from TYPE I CMTSI.

Finally, the controlled net is obtained by adding the six control places as shown in Table 7. It is live and maximally permissive with 205 reachable markings. However, 59 ESSPs and nine control places are required in [12]. It hints that 59 sets of inequalities are needed in [12], while only 18 sets of inequalities suffice using our algorithm. Hence, our policy is more efficient than that in [12].

Li *et al*. solve this problem by using elementary siphons controlled policy (ESCP) and the theory of region [16]. A two-stage deadlock prevention method is used. First, ESCP is used to replace a siphon control method[5]. Therefore, the number of dead markings is reduced. Second, the theory of regions is used to obtain the optimal solution. Three elementary siphons (i.e. $S_1 = \{p_2, p_5, p_{13}, p_{15}, p_{18}\}$, $S_2 = \{p_5, p_{13}, p_{14}, p_{15,} p_{18}\}$ and $S_3 = \{p_2, p_7, p_{11}, p_{13}, p_{16}–p_{19}\}$) can be identified. As a result, three control places $V_{S1}–V_{S3}$ as shown in Table 8 are needed to handle three elementary siphons. Then a partially controlled PN system, denoted as (N_{2L1}, M_0), can be obtained after the first stage.

M''_D	Ω''_C	$M_0(C_{pi})$	$•(C_{pi})$	$(C_{pi})•$
M_{19}	$\{(M_{56}, t_1)\}$	2	t_6, t_{13}	t_1, t_{11}
		4	t_7, t_{11}	t_1, t_9
	$\{(M_{56}, t_9)\}$	3	t_5, t_7, t_{11}	t_2, t_6, t_9
		4	t_7, t_{11}	t_1, t_9
M_{76}	$\{(M_{65}, t_{11})\}$	1	t_5, t_{13}	t_2, t_{11}
M_{77}	$\{(M_{56}, t_{11})\}$	1	t_5, t_{13}	t_2, t_{11}
M_{83}	$\{(M_{128}, t_{11})\}$	1	t_5, t_{13}	t_2, t_{11}
M_{84}	$\{(M_{60}, t_9)\}$	3	t_5, t_7, t_{11}	t_2, t_6, t_9
	$\{(M_{60}, t_{11})\}$	1	t_5, t_{13}	t_2, t_{11}
M_{105}	$\{(M_{93}, t_{11})\}$	1	t_5, t_{13}	t_2, t_{11}
M_{106}	$\{(M_{98}, t_{11})\}$	1	t_5, t_{13}	t_2, t_{11}
M_{112}	$\{(M_{122}, t_{11})\}$	1	t_5, t_{13}	t_2, t_{11}
M_{113}	$\{(M_{96}, t_{11})\}$	1	t_5, t_{13}	t_2, t_{11}

Table 6. Control Places from TYPE II CMTSI.

Additional Control Places	$M_0(C_{pi})$	$•(C_{pi})$	$(C_{pi})•$
Cp_1	1	t_5, t_{13}	t_2, t_{11}
Cp_2	2	t_6, t_{13}	t_1, t_{11}
Cp_3	3	t_6, t_{11}	t_2, t_4, t_9
Cp_4	3	t_5, t_7, t_{11}	t_2, t_6, t_9
Cp_5	3	t_7, t_{11}	t_4, t_5, t_9
Cp_6	4	t_7, t_{11}	t_1, t_9

Table 7. Control Places for the Net (N_{2H}, M_0).

However, (N_{2L1}, M_0) has a dead marking. Figure 13 shows a partial reachability graph and the deadlock marking M_{57} is included. M_{57} is one of the 210 reachable markings (i.e. the reachable markings M_1-M_{210} as denoted in [16]. In [16], 210 reachable markings are divided into two categories: legal and illegal zones. The illegal zone consists of quasi-dead markings (i.e. M_{54}, M_{55}, M_{56} and M_{60}) and a dead marking (i.e. M_{57}). Obviously, some legal markings (i.e. M_{43}, M_{44}, M_{47}, M_{48}, M_{49}, M_{53}, M_{59} and M_{74}) can enter the illegal zone (i.e. $Z_I = Z_Q \cup Z_D$). One can realize that they use the theory of regions to prevent these legal markings from entering Z_I. Therefore, they must resolve 8 MTSIs (i.e. $\{(M_{43}, t_9)\}$, $\{(M_{44}, t_9)\}$, $\{(M_{47}, t_9)\}$, $\{(M_{48}, t_9)\}$, $\{(M_{49}, t_9)\}$, $\{(M_{53}, t_4)\}$, $\{(M_{59}, t_1)\}$, $\{(M_{74}, t_2)\}$) and many equations in this example. Then the additional three control places can be obtained as shown in Table 9. Eight MTSIs are needed. Hence,

their control policy[16] does not seem efficiently enough when the MTSI at the second stage is used to obtain control places.

To compare the efficiency of the deadlock prevention methods, the proposed one is examined in the system net (N_{2L1}, M_0). One can realize that M_{57} is the only deadlock marking in $R(N_{2L1}, M_0)$. Based on our method, only the dead marking M_{57} needed to be controlled. Here, only one $\Omega' = \{(M_{44}, t_9)\}$ is needed. Obviously, the involved necessary equations are much less than those of the conventional one. We can obtain the same controlled net as that in [16]. Hence, the proposed concept of CMTSIs can be used in their approach [16] to improve its computational efficiency significantly as well.
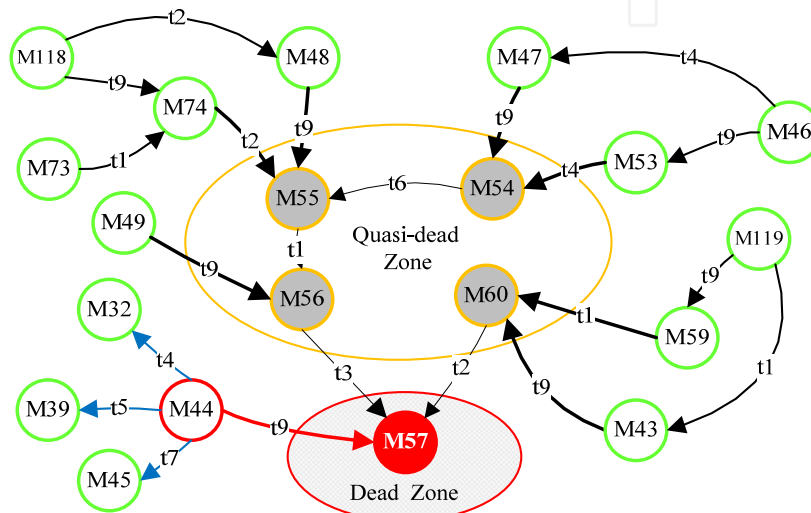

Figure 13. A Partial Reachability Graph of the Net (N_{2L1}, M_0)[16].

Additional Control Places	$M_0(C_{pi})$	$\bullet(C_{pi})$	$(C_{pi})\bullet$
V_{S1}	1	t_5, t_{13}	t_2, t_{11}
V_{S2}	2	t_4, t_5, t_{13}	t_1, t_{11}
V_{S3}	3	t_7, t_{11}	t_4, t_5, t_9

Table 8. Additional Control Places for the Net (N_{2L1}, M_0)

Additional Control Places	$M_0(C_{pi})$	$\bullet(C_{pi})$	$(C_{pi})\bullet$
C_{p1}	3	t_6, t_{11}	t_2, t_4, t_9
C_{p2}	3	t_5, t_7, t_{11}	t_2, t_6, t_9
C_{p3}	4	t_2, t_4, t_7, t_{11}	t_1, t_6, t_9

Table 9. Control Places for (N_{2L1}, M_0) by Two-Stage Method.

5. Comparison with existing methods

One can attempt to make a comparison with the previous methods[12, 16, 24] in terms of efficiency. The first one proposed by Uzam[12], called *Algorithm U*, is totally based on the theory of regions. It solves six ESSPs in Example I. Then three control places are added on the net such that the controlled net is live and reversible. As for Example II, it solves 59 MTSIs. Nine control places are obtained. However, the proposed deadlock prevention policy called *Algorithm P* solves only two and 18 CMTSIs in Examples I and II, respectively.

The other one is proposed by Li *et al.*[16, 24] called *Algorithm L* in which only the theory of regions is used in Example I. Notice that both the controlled results of *Algorithms L* and *U* are the same in Example I. In Example II, using *Algorithm L*, eight MTSIs are solved and six control places are computed. However, under the two-stage control policy, only one set of MTSI is needed by using our new policy to obtain the controlled result that is as the same as *Algorithm L* in Example II. Note that both the definitions of ESSP and MTSI are the same. Hence, ESSP and CMTSI can be regarded as MTSI for the comparison purpose. The detailed comparison results are given in Table 10. However, only 18 MTSIs among 59 MTSIs are needed by using *Algorithm P*.

Example	# of Places	# of Resource Places	MTSI U, L, P	Control Places U, L, P	Reachable Markings
I	11	3	6, 6, 2	3, 3, 3	15
II	19	6	59,／,18	9,／, 6	205
II (two stages)			／, 8, 1	／, 6, 6	

Table 10. Comparison of the Controlled Systems.

For Example II, eight MTSIs are required to obtain the six control places under *Algorithm L*. Hence, one can infer that its performance is better than that of *Algorithm U*. Only one set of CMTSI is needed to obtain the same control result by *Algorithm P*. As a result, one can conclude that our proposed policy is more efficient than the other two methods.

To examine and compare the efficiency of the proposed method with those in[16, 24] in a system with large reachability graphs, one can use eight different markings of p_1, p_8, p_{15}, p_{18}, and p_{19}: $[6, 5, 1, 1, 1]^T$, $[7, 6, 2, 1, 1]^T$, $[7, 6, 1, 2, 1]^T$, $[7, 6, 1, 1, 2]^T$, $[9, 8, 2, 2, 2]^T$, $[12, 11, 3, 3, 3]^T$, $[15, 14, 4, 4, 4]^T$, and $[18, 17, 5, 5, 5]^T$. Tables 11 and 12 show various parameters in the plant and partially controlled net models, where $M(p_{15})$, $M(p_{18})$, and $M(p_{19})$ vary; $|R|$, $|M_L|$, $|R_D|^U$, $|R_D|^L$, indicate the number of reachable markings (states), legal markings, and dead markings under *Algorithms U* and *L*, respectively. Additionally, MTSIs of *Algorithms U, L*, and *P* are symbolized by $|\psi|^U$, $|\psi|^L$ and $|\psi|^P$, respectively. The last column is $r_a = |\psi|^P / |\psi|^U$ in Table 11, and $r_b = \psi|^P / |\psi|^L$ in Table 12. Notably, *Algorithm G*[13] can be regarded as

Algorithm U in Table 11 since the number of MTSIs and ESSPs are the same. In table 11, here, N_{sep} represents the number of MTSIs, and the N_{sep}/U, N_{sep}/L and N_{sep}/P represent the number of MTSIs of *Algorithms U, L*, and *P*, respectively. Obviously, the number of $|\psi|^{U}$ in the plant model grows quickly from cases 1 to 8. For instance, when $M(p_{15}) = M(p_{18}) = M(p_{19}) = 5$, $|\psi|^{U} = 4311$, meaning that one must solve 4311 MTSIs when *Algorithm U* is used. However, since $|\psi|^{P} = 228$, only 228 equations (MTSIs) need to be solved under *Algorithm P*. As a result, *Algorithm P* is more efficient than *Algorithm U* in a large system.

CASES	$|R|$	$|M_L|$	$|R_D|^{U}$	$|\psi|^{U}$	$|\psi|^{P}$	r_a
1	282	205	16	59	18	30.5%
2	600	484	27	95	28	29.5%
3	972	870	26	103	26	25.2%
4	570	421	16	107	19	17.8%
5	4011	3711	42	288	42	14.6%
6	27152	26316	84	886	84	9.5%
7	124110	122235	145	2115	145	6.9%
8	440850	437190	228	4311	228	5.3%

Table 11. Parametersin the Plant and Partially Controlled Models with Varying Markings: *U* vs. *P*.

CASE	$|R|$	$|M_L|$	$|R_D|^{L}$	$|\psi|^{L}$	$|\psi|^{P}$	r_b
1	282	205	1	8	1	12.5%
2	600	484	1	8	1	12.5%
3	972	870	6	10	6	60.0%
4	570	421	1	8	1	12.5%
5	4011	3711	9	15	9	60.0%
6	27152	26316	28	48	28	58.3%
7	124110	122235	60	105	60	57.1%
8	440850	437190	108	192	108	56.3%

Table 12. Parametersin the Plant and Partially Controlled Models with Varying Markings: *L* vs. *P*.

In Table 12, the number of MTSIs calculated by *Algorithm L* can be controlled, but *Algorithm P* is more efficient in these cases. For instance, when $M(p_{15}) = M(p_{18}) = M(p_{19}) = 5$, $|\psi|^{L} = 192$, meaning that one still has to solve 192 MTSIs when *Algorithm L* is used. However, $|\psi|^{P} = 108$. Only 108 MTSIs need to be solved by using *Algorithm P*. Importantly, the computational cost can be reduced by using our proposed method when it is compared with those in[12, 16]. In conclusion, *Algorithm P* is more efficient in large reachability graph cases than those in[12-13].

6. Conclusion

The proposed policy can be implemented for FMSs based on the theory of regions and Petri nets, where the dead markings are identified in its reachability graph. The

underlying notion of the prior work is that many inequalities (i.e. MTSIs) must be solved to prevent legal markings from entering the illegal zone in the original PN model. One must generate all MTSIs in a reachability graph and require high computation. This work proposes and uses CMTSI to overcome the computational difficulty. The detail information is also obtained in existing literatures.[26-29] The proposed method can reduce the number of inequalities and thus the computational cost very significantly since CMTSIs are much less than MTSIs in large models. Consequently, it is optimal with much better computational efficiency than those existing optimal policies[12-13, 16]. More benchmark studies will be desired to establish such computational advantages of the proposed one over the prior ones. It should be noted that the problem is still NP-hard the same as other optimal policies due to the need to generate the reachability graph of a Petri net. The future research is thus much needed to overcome the computational inefficiency of all these methods.

Author details

Yen-Liang Pan
Department of Avionic Engineering, R.O.C. Air Force Academy, Taiwan, R.O.C.

Acknowledgement

7. References

[1]   Fanti MP, Zhou MC. Deadlock control methods in automated manufacturing systems. IEEE Trans Syst Man Cybern A Syst Humans, 2004;34(1):5-22.

[2]   Murata T. Petri nets: Properties, analysis and applications. In Proc IEEE. 1989;77(4):541-580.

[3]   Ezpeleta J, Colom JM, Martinez J. A Petri net based deadlock prevention policy for flexible manufacturing systems. IEEE Trans Robot Autom. 1995;11(2):173-184.

[4]   Jeng MD. A Petri net synthesis theory for modeling flexible manufacturing systems. IEEE Trans Syst Man Cybern B Cybern. 1997;27(2):169-183.

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

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

[7]   Iordache MV, Moody JO, Antsaklis PJ. A method for the synthesis liveness enforcing supervisors in Petri nets. Procceedings of the 2001 American Control Conference; 2011 Jun 25-27; Arlington, VA, USA. P. 4943-8. (ISBN:0-7803-6495-3)

[8]   Park J, Reveliotis SA. Algebraic synthesis of efficient deadlock avoidance policies for sequential resource allocation systems IEEE Trans Autom Control. 2000;16(2):190-195.

[9]   Li ZW, Zhou MC. Elementary siphons of Petri nets and their application to deadlock prevention in flexible manufacturing systems. IEEE Trans Syst Man Cybern A Syst Humans. 2004 Jan;34(1):38-51.

[10] Li ZW, Zhou MC. On siphon computation for deadlock control in a class of Petri nets. IEEE Trans Syst Man Cybern A Syst Humans. 2008;38(3):667-679.

[11] Cho H, Kumaran TK, Wysk RA. Graph-theoretic deadlock detection and resolution for flexible manufacturing systems. IEEE Trans Robot Autom. 2000;11(2):190-195.

[12] Uzam M. 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. 2002;19(3):192-208.

[13] Ghaffari A, Rezg N, Xie XL. Design of a live and maximally permissive Petri net controller using the theory of regions IEEE Trans Robot Autom. 2003;19(1):137-142.

[14] Reveliotis SA, Choi JY. Designing reversibility-enforcing supervisors of polynomial complexity for bounded Petri nets through the theory of regions. Lecture Notes in Computer Science. 2006;4024:322-341.

[15] Badouel E, Darondeau P. Theory of Regions. Third Advance Course on Petri Nets. Springer-Verlag, 1998.

[16] Li ZW, Zhou MC, Jeng MD. A maximally permissive deadlock prevention policy for FMS based on Petri net siphon control and the theory of regions. IEEE Trans Autom Sci Eng. 2008;5(1):183-188.

[17] Uzam M, Zhou MC. An iterative synthesis approach to Petri net-based deadlock prevention policy for flexible manufacturing systems. IEEE Trans Syst Man Cybern A Syst Humans. 2007;37(3):362-371.

[18] Li ZW, Zhou MC, Wu NQ. A survey and comparison of petri net-based deadlock prevention policies for flexible manufacturing systems. IEEE Trans Syst Man Cybern C Appl Rev. 2008 ;38(2):173-188.

[19] Piroddi L, Cordone R, Fumagalli I. Selective siphon control for deadlock prevention in Petri nets. IEEE Trans Syst Man Cybern A Syst Humans. 2008;38(6):1337-1348.

[20] Silva M, Teruel E, Colom JM. Linear algebraic and linear programming techniques for the analysis of P/T net systems. LNCS, 1998;1491:303-373.

[21] Ramadge PJ, Wonham WM. The control of discrete event systems. Proc IEEE. 1989;77(1):81-98.

[22] Viswanadham N, Narahari Y, Johnson TL. Deadlock prevention and deadlock avoidance in flexible manufacturing systems using Petri net models. IEEE Trans Robot Autom. 1990;6(6):713-723.

[23] Abdallah IB, ElMaraghy HA. Deadlock prevention and avoidance in FMS: A Petri net based approach. Int J Adv Manuf Technol. 1998;14(10):704-715.

[24] Li ZW, Wang A, Lin H. A deadlock prevention approach for FMS using siphon and the theory of regions. Proceeding of the IEEE International Conference on Systems, Man and Cybernetics; 2004 Oct 10-13; The Hague, Netherlands. P. 5079-84. (ISBN:0-7803-8566-7)

[25] INA. (Integrated Net Analyzer), A Software Tool for Analysis of Petri Nets. Version 2.2, 31.07. 2003. [Online]. Available: http://www.informatik.hu-berlin.de/~starke/ina.html.

[26] Huang YS, Pan YL. Enhancement of An Efficient Liveness-Enforcing Supervisor for Flexible Manufacture Systems. Int J Adv Manuf Technol. 2010;48:725-737. (DOI: 10.1007/s00170-009-2299-x)

[27] Huang YS, Pan YL. An Improved Maximally Permissive Deadlock Prevention Policy Based on the Theory of Regions and Reduction Approach. IET Control Theory Appl. 2011;5(9):1069-1078. (DOI: 10.1049/iet-cta.2010.0371.)

[28] Pan YL, Huang YS. Solutions for Deadlocked Problem of FMSs Using Theory of Regions. Adv Mat Res. 2011 ;314-346 :535-538. (DOI:10.4028/www.scientific.net/AMR.314-316.535.)

[29] Huang YS, Pan YL, Zhou MC. Computationally Improved Optimal Deadlock Control Policy for Flexible Manufacturing Systems. IEEE Trans Syst Man Cybern A Syst Humans. 2012; 42(2):404-415.