Hindawi
Discrete Dynamics in Nature and Society
Volume 2017, Article ID 1395142, 14 pages
https://doi.org/10.1155/2017/1395142

*Research Article*

# An Enhanced Supervisory Control Strategy for Periodicity Mutual Exclusions in Discrete Event Systems Based on Petri Nets

**Zhongyuan Jiang,[1] Huan Wang,[2] Xiaoliang Chen,[1] Mingwei Tang,[1] and Jianhong Ye[3]**

[1] *School of Computer and Software Engineering, Xihua University, Chengdu 610039, China*
[2] *School of Information Science and Technology, Southwest Jiaotong University, Chengdu 611756, China*
[3] *School of Computer Science and Technology, Huaqiao University, Xiamen 361021, China*

Correspondence should be addressed to Zhongyuan Jiang; jiangzydj@163.com

Received 2 September 2016; Accepted 8 December 2016; Published 30 January 2017

Academic Editor: Francisco R. Villatoro



Mutual exclusion problems widely exist in discrete event systems in which several processes will compete for the common resource for maintaining their normal running. This competition is mutually exclusive. However, a special behavior, that is, periodic mutual exclusion behavior, is important for many discrete event systems. Once a process obtains the common resource, it will consecutively obtain the common resource in the following several competitions. The other processes should wait for the release of the common resource. All processes will compete for the common resource again after the common resource is released. These competitions have obvious periodicity. In this paper, a methodology is proposed to design periodic mutual exclusion supervisors to control the periodic mutual exclusion behavior in discrete event systems. Moreover, two original structural conversion concepts, called $k$-derivation and $k$-convergence processes, are proposed to construct the periodic mutual exclusion supervisors. The discussion results show that many undesirable execution sequences are forbidden since the periodic mutual exclusion behavior is controlled by the proposed periodic mutual exclusion supervisors. Finally, an example is used to illustrate the proposed methodology.

## 1. Introduction

Discrete event systems [1] are classic dynamic systems, which widely exist in communication, manufacturing, traffic, and computer network fields, such as e-business systems, online game server systems, embedded systems, flexible manufacturing systems, and traffic management systems [2]. The concurrency behavior [3] of discrete event systems is to consider several events that simultaneously occur and potentially interact with each other. Concurrent events can be alternately executed on a single processor but can be also concurrently executed on several processors. The occurrence sequences of these concurrent events are important since system concurrency behavior can optimize system performances and promote the sufficient utilization for finite system resources [4].

The occurrence of some concurrent events may simultaneously occupy some special resources, such as a CPU, a printer, a robot, or a shared file. These special resources are finite because of the restriction of high implementation costs. Therefore, these concurrent events should share their special resources. This will form a competition among these concurrent events [5]. On the other hand, the competition will maximize the utilization of shared special resources. A most widespread and important phenomenon is that only one special resource can be used for several concurrent events. This results in a mutual exclusion behavior [6] for these concurrent events (only one of them can obtain the special resource at the same time and the other events should wait for the release of the occupied special resource). Mutual exclusion behaviors as an important character in discrete event systems are usually used for resource allocation (resource sharing) or marking constraint in Petri nets [7], such as the user request distribution in a single input cluster of computer servers, the raw material distribution of a robot for two production lines, and the database access control for multiple processes.

In discrete event systems, the special resource utilization rates are an important indicator to measure the performance of discrete event systems. It is necessary to design appropriate control strategies to control the special resource distribution for discrete event systems. The purpose is to maximize the utilization of shared special resources. The authors in [8] consider deterministic feasibility and time complexity of two fundamental tasks in the distributed computing of consensus and mutual exclusions. In [9], a mutual exclusion element is described using a reflective semiconductor optical amplifier and a simple scheme for contention resolution in arrayed waveguide grating router based optical switches in data centers. The authors in [10] propose an approach to compete for the privilege of passing the intersection that is a classical mutual exclusion problem via the communications among vehicles and infrastructure. In [11], two algorithms are presented to compete for exclusive access to a shared resource among geographically close nodes in a mobile ad hoc network.

Supervisory control theory [12, 13] is widely used to perform the controller syntheses of discrete event systems, which is proposed by Ramadge and Wonham based on automata [14, 15]. In order to avoid the numerous state representations of automata, the subsequent researchers use Petri nets to study the control strategies. The authors in [16] present a framework for supervisor synthesis for discrete event systems. The approach is based on compositional minimization by using concepts of process equivalence. For the studies of control strategies in discrete event systems, maximally permissive problems should be considered. The authors in [17] present a computational method to design optimal control places in order to obtain a maximally permissive liveness-enforcing supervisor by using a vector covering approach. The authors in [18] distinguish between the offline and the online computation that is required for the effective implementation of the maximally permissive deadlock avoidance policy. The authors in [19] study the maximally permissive abstractions for the hierarchical and decentralized control of large-scale discrete event systems. Moreover, the structure optimization problems of supervisors are also important. The authors in [20] present two iterative deadlock prevention policies for flexible manufacturing systems. The proposed methods can reduce the overall computational time. The authors in [21] propose a deadlock prevention method to obtain maximally permissive supervisor and minimize its structure. The authors in [22] propose a deadlock prevention method for a flexible manufacturing system with a minimal supervisory structure. Furthermore, researchers also concern the design and implementation problems of Petri net based supervisors. In [23], the authors present the design, generation, and implementation of coordinating discrete event control code using Petri nets for an operating flexible manufacturing system. The authors in [24] use Petri nets to model the operated behaviors and to synthesize the command filters in a command filtering framework. Automation Petri nets can be used to perform the design and implementation of discrete event control systems by converting automation Petri nets into ladder diagrams on programmable logic controllers [25].

The main researches of supervisory control theory are divided into several classic problems based on Petri nets, that is, forbidden-state problems [26, 27], forbidden-string problems [28, 29], deadlock prevention problem [30], and liveness maintenance problems [31, 32]. The deadlock prevention problems and liveness maintenance problems can be transformed to forbidden-state problems directly [33, 34]. The authors in [35] address the forbidden-state problem of Petri nets by computing maximally permissive Petri net controllers. The design of control strategies for mutual exclusion mechanisms can be considered as resolving a forbidden-string problem. The authors in [36] use a colored generalized stochastic Petri net model to study the performance and correctness of a Lamport concurrent algorithm to solve the mutual exclusion problems. The authors in [37] discuss the simultaneous events in mutual exclusion transitions. The execution modes of sequential function charts are presented based on Petri nets. In [38], a method is proposed to control the occurrence of simultaneous events in mutual exclusion transitions by Petri nets based on supervisory control theory. The authors in [5] formulate two resource-sharing concepts, that is, parallel mutual exclusion and sequential mutual exclusion, and provide a theoretical basis for Petri net synthesis methods to model systems.

A class of specifications, that is, generalized mutual exclusion constraints, is studied in [39] for discrete event systems modeled using place/transition nets. According to these specifications, many constraints that deal with mutual exclusion problems between state and events or just events themselves can be transformed into generalized mutual exclusion constraints. This can be enforced by a set of places if all transitions are controllable. For the transitions that are uncontrollable, this specification is not always applicable. The authors in [40] provide a class of generalized mutual exclusion constraints on a class of forward-concurrent-free nets. In [41], a type of specification, called OR-AND generalized mutual exclusion constraints, is defined for place/transition nets. Such a specification consists of a disjunction of conjunction of several single generalized mutual exclusion constraints. The authors in [42] enforce the generalized mutual exclusion constraints on a Petri net plant by replacing the classical partition of event set into controllable and uncontrollable events from supervisory control theory. In [43], an algorithm is presented to transform a given generalized mutual exclusion constraint into an optimal admissible one for a class of Petri nets whose uncontrollable influence subnets are forward synchronization and backward conflict-free nets.

It is difficult to summarize the constraints for complex mutual exclusion mechanisms and special optimization problems in discrete event systems by using the existing control strategies, such as the following described periodic mutual exclusion problems. In many discrete event systems, one of the concurrent processes will occupy a common resource to maintain its normal running. To satisfy some special requirements, this process will consecutively obtain the common resource during the competition of the following finite times after it won a competition. The other processes only can wait for the release of the common resource. All concurrent processes will form a new competition again after this process finishes a periodical operation for the special requirements and the common resource is released. This mutually exclusive competition among these concurrent processes has periodicity, called periodic mutual exclusion behavior. It is necessary to design appropriate control strategies to distribute the common resources for these concurrent processes to control the periodic mutual exclusion problems.

In this paper, a methodology is proposed to design periodic mutual exclusion supervisors to control the periodic mutual exclusion behavior in discrete event systems. The structures of general mutual exclusion systems and periodic mutual exclusion systems are defined by using Petri nets. The proposal is convenient to construct formal models to analyze the properties of the two classes of mutual exclusion systems. Two original structural conversion concepts, called $k$-derivation process and $k$-convergence process, are proposed to construct the periodic mutual exclusion supervisors. The main idea is to derive a common resource to several virtual resources by the preliminary $k$-derivation processes if a process obtains the common resource and these derived virtual resources will be converged into a common resource by the final $k$-convergence processes after this process releases the common resource. The execution sequences of all processes are discussed in mutual exclusion systems with general mutual exclusion supervisors and periodic mutual exclusion supervisors, respectively. The discussion results show that many undesirable execution sequences are forbidden in the mutual exclusion systems with the periodic mutual exclusion supervisors because the periodic mutual exclusion behavior is controlled by these periodic mutual exclusion supervisors. Finally, a real example is used to illustrate the proposed methodology.

The main contributions of this paper are concluded as follows:

(1) The structures of general mutual exclusion systems and periodic mutual exclusion systems are defined by using Petri nets. The proposal is convenient for constructing formal models for these two classes of mutual exclusion systems.

(2) Two original structural conversion concepts, that is, $k$-derivation process and $k$-convergence process, are proposed to derive a common resource to several virtual resources in the preliminary stage and converge these virtual resources into a common resource in the final stage, respectively. The purpose is to enhance the execution permission for each process that obtains the common resource.

(3) A methodology is proposed to design periodic mutual exclusion supervisors to control the periodic mutual exclusion behavior in discrete event systems. In the enhanced mutual exclusion systems, many undesirable execution sequences of all processes are forbidden by the periodic mutual exclusion supervisors.

The rest of this paper is organized as follows. Section 2 briefly recalls some basics of Petri nets. Section 3 introduces a class of mutual exclusion systems with resources and their periodic mutual exclusion behavior is discussed. Section 4 proposes the design method of periodic mutual exclusion supervisors. Section 5 gives the discussion of periodic mutual exclusion behavior in general mutual exclusion systems and periodic mutual exclusion systems. Section 6 introduces two examples to illustrate the proposed methods. Finally, the proposed methods are concluded in Section 7.

## 2. Preliminaries

We assume that the readers are familiar with the basics of Petri nets. Only some key concepts of Petri nets are provided. More details can be found in [44].

A Petri net is a 4-tuple $N = (P, T, F, W)$, where $P$ and $T$ are finite, $P \neq \emptyset$, $T \neq \emptyset$, and $P \cap T = \emptyset$. $P$ is a set of places and $T$ is a set of transitions. $F \subseteq (P \times T) \cup (T \times P)$ is called a flow relation represented by an arc with arrow from places to transitions and transitions to places. $W : F \to \mathbb{N} \setminus \{0\}$ is a mapping that assigns a weight to an arc, where $\mathbb{N}$ is the set of nonnegative integers. A Petri net $N = (P, T, F, W)$ is called an ordinary net if, $\forall f \in F, W(f) = 1$ (denoted as $N = (P, T, F)$). An ordinary net $N = (P, T, F)$ is called a state machine if $t \in T$ such that $t$ has only one input place and only one output place. A Petri net is self-loop-free if and only if $\nexists x, y \in P \cup T$ such that $f(x, y) \in F$ and $f(y, x) \in F$.

A marking $M$ of $N$ is a mapping from $P$ to $\mathbb{N}$. $(N, M_0)$ is called a net system, where $M_0$ is the initial marking of $N$. The preincidence matrix $\mathrm{Pre} : P \times T \to \mathbb{N}$ of $N$ is $\mathrm{Pre}(p, t) = W(p, t)$. The postincidence matrix $\mathrm{Post} : P \times T \to \mathbb{N}$ of $N$ is $\mathrm{Post}(p, t) = W(t, p)$. A self-loop-free Petri net $N = (P, T, F, W)$ can be represented by its incidence matrix $[N]$, where $[N]$ is a $|P| \times |T|$ integer matrix with $[N](p, t) = \mathrm{Post}(p, t) - \mathrm{Pre}(p, t)$. For $p \in P$, $M(p)$ denotes the sum of tokens contained in place $p$. Place $p$ is marked by a marking $M$ if and only if $M(p) > 0$. A subset $S \subseteq P$ is marked by $M$ if and only if $\exists p \in S$ such that $p$ is marked by $M$. The sum of tokens in all places in $S$ is denoted by $M(S)$; that is, $M(S) = \sum_{p \in S} M(p)$. The subset $S$ is said to be empty at $M$ if and only if $M(S) = 0$.

The preset of a node $x \in P \cup T$ is defined as ${}^\bullet x = \{y \in P \cup T \mid (y, x) \in F\}$ and the postset of a node $x \in P \cup T$ is defined as $x^\bullet = \{y \in P \cup T \mid (x, y) \in F\}$. For a set of nodes $X \subseteq P \cup T$, ${}^\bullet X = \bigcup_{x \in X} {}^\bullet x$ and $X^\bullet = \bigcup_{x \in X} x^\bullet$. $|X|$ represents the item count in $X$. A transition $t \in T$ is enabled at a marking $M$ if $p \in {}^\bullet t$ such that $M(p) > W(p, t)$, which is denoted as $M[t\rangle$. If $t$ is enabled, its firing yields another marking $M'$ such that, $\forall p \in P, M'(p) = M(p) - W(p, t) + W(t, p)$, denoted as $M[t\rangle M'$. Marking $M'$ is called reachable from $M$ if there exists a firing sequence $\sigma = t_1 t_2 \cdots t_n$ such that $M[t_1\rangle M_1 t_2\rangle M_2 \cdots M_{n-1} t_n\rangle M'$. This is denoted by $M[\sigma\rangle M'$.

A $P$-vector is a column vector $I : P \to \mathbb{Z}$ indexed by $P$, where $\mathbb{Z}$ is the set of integers. We denote a column vector where every entry equals 0(1) by $\mathbf{0(1)}$. $I^T$ and $[N]^T$ are the transposed versions of vector $I$ and matrix $[N]$, respectively. $P$-vector $I$ is a $P$-invariant if $I \neq \mathbf{0}$ and $I^T[N] = 0^T$. $\|I\| = \{p \in P \mid I(p) \neq 0\}$ is called the support of $I$. $\|I\|^+ = \{p \in P \mid I(p) > 0\}$ (resp., $\|I\|^- = \{p \in P \mid I(p) < 0\}$) denotes the positive (resp., negative) support of $I$. $P$-invariant $I$ is called a minimal $P$-invariant if $\|I\|$ is not a superset of the support

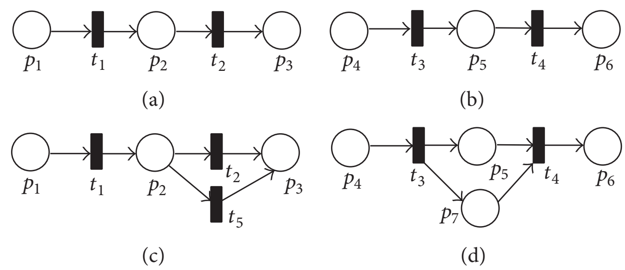


Figure 1: Illustration of linearity processes.

of any other one and its components are mutually prime. If $I$ is a $P$-invariant of $N$, then $M \in R(N, M_0)$ such that $I^T \cdot M = I^T \cdot M_0$.

## 3. Mutual Exclusion Systems with Resources

In this section, the model of mutual exclusion systems with resources (MESRs) is defined by using Petri nets with three classical patterns, that is, parallel pattern, dispersive pattern, and polymeric pattern. The details are shown as follows.

*Definition 1.* Let $N = (P, T, F, W)$ be a Petri net and $c \in P \cup T$ be a node of $N$. Node $c$ is called a root of $N$ if $^{\bullet}c = \emptyset$. Node $c$ is called a leaf of $N$ if $c^{\bullet} = \emptyset$.

For a Petri net, it may contain one or more roots and leaves. However, it is impossible to contain one or more roots or leaves in any closed-loop Petri net.

*Definition 2.* Let $N = (P, T, F, W)$ be a Petri net and $s_1, s_2, \ldots, s_n \in P \cup T$ be $n$ nodes such that $s_i \neq s_j$, where $i \neq j$, $\forall i, j \in \{1, 2, \ldots, n\}$, and $n \geq 2$. A transient process, denoted as $S = s_1 s_2, \ldots, s_n$, can be defined if $s_i \in {}^{\bullet}s_{i+1}$, where $i \in \{1, 2, \ldots, n-1\}$.

*Definition 3.* Let $N = (P_K \cup \{p_r\} \cup \{p_l\}, T, F)$ be a state machine and $s_1, s_2, \ldots, s_{|P_K \cup T|} \in P_K \cup T$ be $|P_K \cup T|$ nodes such that $s_i \neq s_j$, where $i \neq j$ and $\forall i, j = \{1, 2, \ldots, |P_K \cup T|\}$. $N$ is called a linearity process if

(1) $P_K$ is a set of places that represent the states of system;

(2) $N$ contains only a root and a leaf, where $p_r \notin P_K$ is the root of $N$, $p_l \notin P_K$ is the leaf of $N$, and $p_l \neq p_r$;

(3) $N$ only exhibits a transient process, denoted as $S_N = p_r s_1 s_2 \cdots s_{|P_K \cup T|} p_l$.

*Definition 4.* Let $N = (P_K \cup \{p_r\} \cup \{p_l\}, T, F)$ be a linearity process and $S$ be the transient process from $p_r$ to $p_l$. If all places are removed from $S$, the rest of the transient process $S$, denoted as $S^T$, is called the T-Process of $N$.

Figure 1(a) shows a linearity process $N^1 = (\{p_2\} \cup \{p_1\} \cup \{p_3\}, \{t_1, t_2\}, F)$. Place $p_1$ is a root and place $p_3$ is a leaf of $N^1$. $S_{N^1} = p_1 t_1 p_2 t_2 p_3$ is a transient process and $S_{N^1}^T = t_1 t_2$ is a T-Process of $N^1$. Figure 1(b) shows another linearity process $N^2 = (\{p_5\} \cup \{p_4\} \cup \{p_6\}, \{t_3, t_4\}, F)$. Place $p_4$ is a root and place $p_6$ is a leaf of $N^2$. $S_{N^2} = p_4 t_3 p_5 t_4 p_6$ is a transient process and $S_{N^2}^T = t_3 t_4$ is a T-Process of $N^2$. Figure 1(c) shows a Petri net that is not a linearity process since it contains two transient processes $S_{N^3} = p_1 t_1 p_2 t_2 p_3$ and $S'_{N^3} = p_1 t_1 p_2 t_5 p_3$. Figure 1(d) shows a Petri net that is also not a linearity process since it is not a state machine.

In a linearity process, the firing of its transitions may be restrained by some special resources, such as a CPU, a robot, a balancer, or a printer. Therefore, a linearity process with a special resource is defined as follows.

*Definition 5.* Let $N = (P_K \cup \{p_r\} \cup \{p_l\}, T, F)$ be a linearity process and place $p_s \notin (P_K \cup \{p_r\} \cup \{p_l\})$ be a resource. A linearity process with the resource $p_s$, denoted as $N = (P_K \cup \{p_r\} \cup \{p_l\} \cup \{p_s\}, T, F)$, is defined as follows:

(1) $p_r^{\bullet} = p_s^{\bullet}$ and $^{\bullet}p_l = {}^{\bullet}p_s$.

(2) $t \in T \setminus (p_r^{\bullet} \cup {}^{\bullet}p_l)$ such that $p_s \notin {}^{\bullet}t$ and $p_s \notin t^{\bullet}$.

(3) $M_0(p_s) = 1$, where $M_0$ is the initial marking.

Let $N = (P_K \cup \{p_r\} \cup \{p_l\} \cup \{p_s\}, T, F)$ be a linearity process with the resource $p_s$. $\forall t \in p_r^{\bullet}$, the firing of $t$ will obtain the special resource $p_s$. Conversely, $\forall t' \in {}^{\bullet}p_l$, the firing of $t'$ will release the special resource $p_s$. Figure 2(a) shows two linearity processes with two special resources $p_s$ and $p'_s$; that is, $N_1 = (\{p_2\} \cup \{p_1\} \cup \{p_3\} \cup \{p_s\}, \{t_1, t_2\}, F)$ and $N_2 = (\{p_5\} \cup \{p_4\} \cup \{p_6\} \cup \{p'_s\}, \{t_3, t_4\}, F)$. In $N_1$, the special resource is occupied if $t_1$ fires and it is released if $t_2$ fires. This characteristic also exists in $N_2$.

In a discrete event system, it may contain several linearity processes that compete for a common special resource. These linearity processes of the discrete event system can construct a mutual exclusion system.

*Definition 6.* Let $N^1, N^2, \ldots$ and $N^n$ with $N^i = (P_K^i \cup \{p_r^i\} \cup \{p_l^i\} \cup \{p_s^i\}, T^i, F^i)$ ($\forall i \in \{1, 2, \ldots, n\}$) be $n$ ($n \geq 2$) linearity processes with resources such that $|P_K^1| = |P_K^2| = \cdots = |P_K^n|$, $|T^1| = |T^2| = \cdots = |T^n|$, $p_s^1 = p_s^2 = \cdots = p_s^n$. An MESR, denoted as $N = (P_K \cup P_R \cup P_L \cup \{p_s\}, T, F)$, is defined as follows:

(1) $P_K = \bigcup_{i=1}^{n} P_K^i$, $P_R = \{p_r^1, p_r^2, \ldots, p_r^n\}$, $P_L = \{p_l^1, p_l^2, \ldots, p_l^n\}$, $p_s = p_s^1$, $M_0(p_s) = 1$, where $M_0$ is the initial marking of $N$.

(2) $T = \bigcup_{i=1}^{n} T^i$.

(3) $F = \bigcup_{i=1}^{n} F^i$.

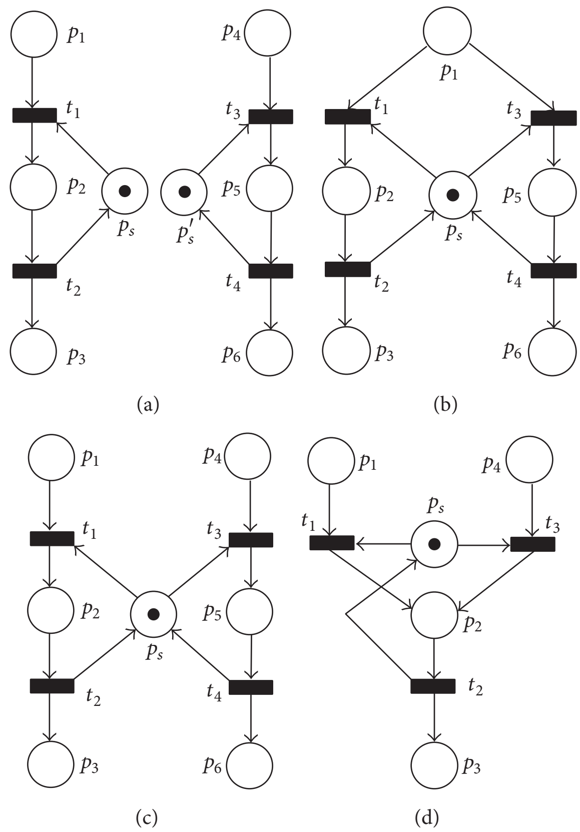


Figure 2: (a) Two linearity processes with two special resources, (b) an MESR with dispersive pattern, (c) an MESR with parallel pattern, and (d) an MESR with polymeric pattern.

The MESR $N$ can be represented with one of the following three classical patterns:

(i) $N$ is represented with dispersive pattern if $p_r^1 = p_r^2 = \cdots = p_r^n$, $\bigcap_{i=1}^n (P_K^i \cup \{p_l^i\}) = \emptyset$, and $\bigcap_{i=1}^n T_i = \emptyset$.

(ii) $N$ is represented with parallel pattern if $\bigcap_{i=1}^n (P_K^i \cup \{p_r^i\} \cup \{p_l^i\}) = \emptyset$ and $\bigcap_{i=1}^n T_i = \emptyset$.

(iii) $N$ is represented with polymeric pattern if $p_r^i \neq p_r^j$, $p_r^{i\bullet} \neq p_r^{j\bullet}$, $i \neq j$, $\forall i, j \in \{1, 2, \ldots, n\}$, $P_K^1 = P_K^2 = \cdots = P_K^n$, $p_l^1 = p_l^2 = \cdots = p_l^n$, and $T_1 \setminus p_r^{1\bullet} = T_2 \setminus p_r^{2\bullet} = \cdots = T_n \setminus p_r^{n\bullet}$.

Figure 2(b) shows an MESR with dispersive pattern, which resulted from the composition of $N_1$ and $N_2$ depicted in Figure 2(a) via $p_s = p_s'$ and $p_1 = p_4$. Figure 2(c) shows an MESR with parallel pattern, which resulted from the composition of $N_1$ and $N_2$ via $p_s = p_s'$. Figure 2(d) shows an MESR with polymeric pattern, which resulted from the composition of $N_1$ and $N_2$ via $p_s = p_s'$, $p_2 = p_5$, $t_2 = t_4$, and $p_3 = p_6$.

*Property 1.* Let $N$ be an MESR with $N = (P_K \cup P_R \cup P_L \cup \{p_s\}, T, F)$ and $M_0$ be the initial marking of $N$. Then, $M \in R(N, M_0)$ satisfies

$$M(p_s) + \sum_{p_i \in P_K} M(p_i) = M_0(p_s) = 1. \tag{1}$$

*Proof.* It is easy to see that place $p_s$ and all places in $P_K$ can construct a $P$-invariant. The result naturally holds. □

For example, the MESR with dispersive pattern depicted in Figure 2(b) satisfies the restriction $M(p_2) + M(p_5) + M(p_s) = 1$. Similarly, the MESR with parallel pattern depicted in Figure 2(c) satisfies the restriction $M(p_2) + M(p_5) + M(p_s) = 1$. For the MESR with polymeric pattern depicted in Figure 2(d), it satisfies the restriction $M(p_2) + M(p_s) = 1$.

*Property 2.* Let $N = (P_K \cup P_R \cup P_L \cup \{p_s\}, T, F)$ be an MESR with dispersive pattern and $M \in R(N, M_0)$ be a marking of $N$, where $M_0$ is the initial marking of $N$. Then, $t \in P_R^\bullet$ such that $M[t\rangle$ holds if $\exists t' \in P_R^\bullet$ such that $M[t'\rangle$ holds.

*Proof.* Since $N$ is an MESR with dispersive pattern, we can assume $P_R = \{p_r\}$. Then, we have $P_R^\bullet = p_r^\bullet$. If $\exists t' \in p_r^\bullet$ such that $M[t'\rangle$ holds, we have $M(p_r) \geq 1$ and $M(p_s) = 1$. However, $\forall t \in p_r^\bullet$, we have ${}^\bullet t = \{p_r, p_s\}$ according to the structure of the MESR with dispersive pattern. Therefore, $\forall t \in p_r^\bullet$, $M[t\rangle$ holds

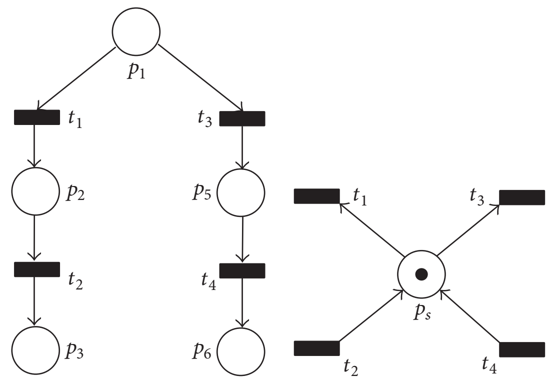


Figure 3: (a) A plant and (b) the corresponding supervisor obtained by generalized mutual exclusion constraints.

since $M(p_r) \geq 1$ and $M(p_s) = 1$. This means that, $\forall t \in P_R^\bullet$, $M[t\rangle$ holds. □

*Property 3.* Let $N = (P_K \cup P_R \cup P_L \cup \{p_s\}, T, F)$ be an MESR with parallel or polymeric pattern and $M \in R(N, M_0)$ be a marking of $N$, where $M_0$ is the initial marking of $N$. Then, $\forall t \in P_R^\bullet$, $M[t\rangle$ holds if $M(p_s) = 1$ and $\forall p_r \in P_R$ such that $M(p_r) \geq 1$.

*Proof.* $\forall t \in P_R^\bullet$, we have ${}^\bullet t = P_R \cup \{p_s\}$ according to the structure of the MESR with parallel or polymeric pattern. If $M(p_s) = 1$ and $\forall p_r \in P_R$ such that $M(p_r) \geq 1$, we have that, $\forall t \in P_R^\bullet$, $M[t\rangle$ holds. □

Let $N = (P_K \cup P_R \cup P_L \cup \{p_s\}, T, F)$ be an MESR. Then, all transitions contained in $P_R^\bullet$ are mutually exclusive. $\forall t_i \in P_R^\bullet$, $t_i$ will compete for the common resource for running its linearity process. If $t_i$ fires (its linearity process obtains the common resource), $\forall t_j \in (P_R^\bullet \setminus \{t_i\})$ cannot fire (they need to wait until they obtain the common resource) since $N$ only contains a common resource. $\forall t_i \in {}^\bullet P_L$, $t_i$ will release the common resource from its linearity process. This should particularly emphasize that all transitions contained in $P_R^\bullet$ will compete for the common resource again for their linearity processes if the common resource is released. For the MESR with dispersive pattern depicted in Figure 2(b), transitions $t_1$ and $t_3$ are both enabled or disenabled for all marking. If they are enabled, they will compete for the common resource $p_s$. Once the common resource $p_s$ is released by one of the transitions $t_2$ and $t_4$, transitions $t_1$ and $t_3$ will compete for the common resource again. Similar competitions also exist in the two MESRs depicted in Figures 2(c) and 2(d).

For an MESR, it may have some special requirements that are concluded as follows:

(i) One of its linearity processes will be consecutively executed $k$ times once this linearity process obtains the common resource, where $k \geq 1$ is an integer. The other linearity processes cannot be executed and need to wait until they obtain the common resource.

(ii) Once this linearity process is consecutively executed $k$ times and the common resource is released, all linearity processes will compete for the common resource again.

These linearity processes are mutually exclusive with these special requirements. Their mutual exclusion characteristics are called periodic mutual exclusion behavior that widely exists in discrete event systems, such as the online team game server systems, flexible manufacturing systems, and intelligence traffic dispatch systems.

For the traditional mutual exclusion behavior, the methods of generalized mutual exclusion constraints presented in [39] can be used to design supervisors to perform the control of mutual exclusion mechanisms. However, it is difficult to use generalized mutual exclusion constraints to control the periodic mutual exclusion behavior. Therefore, the general mutual exclusion supervisors should be extended to handle these special problems.

## 4. Design of Enforced Supervisors

In this section, the methods of generalized mutual exclusion constraints are simply reviewed, which can be used to design general mutual exclusion supervisors. Furthermore, two important concepts, that is, $k$-derivation process and $k$-convergence process, are defined to extend the general mutual exclusion supervisors. Finally, a method is proposed to design periodic mutual exclusion supervisors to control periodic mutual exclusion problems.

The mutually exclusive competition for a common resource is a classical character in MESRs. Generalized mutual exclusion constraints can be used to design supervisors to perform the mutual exclusion mechanisms. For example, Figure 3(a) shows a plant that is composed of two linearity processes. If we assume that only one of the competition transitions $t_1$ and $t_3$ can fire in the plant, we have a restriction $M(p_2) + M(p_5) \leq 1$, where $M$ is a marking of the plant. A supervisor of the plant can be obtained by using the methods of generalized mutual exclusion constraints, as shown in Figure 3(b). The details about the methods of generalized mutual exclusion constraints can be found in [39]. The controlled system that is combined by the plant

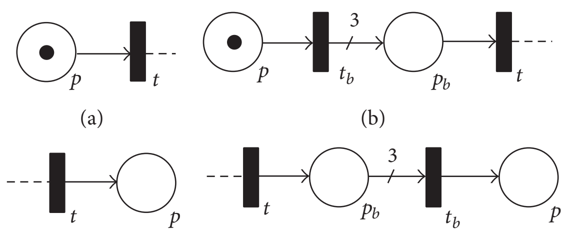


Figure 4: (a) A Petri net, (b) a three-derivation process, (c) a Petri net, and (d) a three-convergence process.

and the supervisor is shown in Figure 2(b). It can satisfy the restriction $M(p_2) + M(p_5) \leq 1$.

In order to design enhanced supervisors to control the periodic mutual exclusion behavior for MESRs, two important concepts, $k$-derivation process and $k$-convergence process, are defined as follows.

*Definition 7.* Let $N = (\{p\}, \{t\}, F, W)$ be a Petri net such that $p^{\bullet} = \{t\}$, ${}^{\bullet}t = \{p\}$, and $W(p, t) = 1$, $p_b \neq p$ be a place, $t_b \neq t$ be a transition, and $k \geq 1$ be an integer. Then, a $k$-derivation process from $p$ to $t$, denoted as $N_d = (\{p\} \cup \{p_b\}, \{t\} \cup \{t_b\}, F_d, W_d)$, is a Petri net if ${}^{\bullet}t_b = \{p\}$, $t_b^{\bullet} = \{p_b\}$, ${}^{\bullet}p_b = \{t_b\}$, $p_b^{\bullet} = \{t\}$, $W_d(p, t_b) = 1$, $W_d(t_b, p_b) = k$, and $W_d(p_b, t) = 1$. Transition $t_b$ is called the derivative transition of $N_d$. Place $p_b$ is called the derivation place of $N_d$. $k$ is called the derivation coefficient of $N_d$.

In the Petri net depicted in Figure 4(a), transition $t$ can fire only once since place $p$ contains only one token. Figure 4(b) shows a three-derivation process from $p$ to $t$, which is transformed from the Petri net depicted in Figure 4(a). Transition $t_b$ is the derivative transition. Place $p_b$ is the derivative place and the derivation coefficient is three. Transition $t$ can fire three times after $t_b$ fired. The firing time of $t$ is derived from one to three.

*Definition 8.* Let $N = (\{p\}, \{t\}, F, W)$ be a Petri net such that $t^{\bullet} = \{p\}$, ${}^{\bullet}p = \{t\}$, and $W(t, p) = 1$, $p_b \neq p$ be a place, $t_b \neq t$ be a transition, and $k \geq 1$ be an integer. Then, a $k$-convergence process from $t$ to $p$, denoted as $N_c = (\{p\} \cup \{p_b\}, \{t\} \cup \{t_b\}, F_c, W_c)$, is a Petri net if ${}^{\bullet}p_b = \{t\}$, $p_b^{\bullet} = \{t_b\}$, ${}^{\bullet}t_b = \{p_b\}$, $t_b^{\bullet} = \{p\}$, $W_c(t, p_b) = 1$, $W_c(p_b, t_b) = k$, and $W_c(t_b, p) = 1$. Transition $t_b$ is called the emergence transition of $N_c$. Place $p_b$ is called the emergence place of $N_c$. $k$ is called the emergence coefficient of $N_c$.

A $k$-convergence process from $t$ to $p$ is an inverse process of a $k$-derivation process from $p$ to $t$. In the Petri net depicted in Figure 4(c), each firing of transition $t$ will add a token to place $p$. Figure 4(d) shows a three-convergence process from $t$ to $p$, which is transformed from the Petri net depicted in Figure 4(c). Transition $t_b$ is the convergence transition. Place $p_b$ is the convergence place and the convergence coefficient is three. Only one token is added to place $p$ after transition $t$ fires three times. The firing time of $t$ is converged from three to one.

*Definition 9.* Let $N$ be an MESR with $N = (P_K \cup P_R \cup P_L \cup \{p_s\}, T, F)$. Petri net $N_M$ with $N_M = (\{p\}, T_M^I \cup T_M^O, F_M)$ is called the mutual exclusion supervisor of $N$ if $p = p_s$, $T_M^I = P_R^{\bullet}$, $T_M^O = {}^{\bullet}P_L$, and $F_M = \bigcup_{t_i \in T_M^I} \{(p_s, t_i)\} \cup \bigcup_{t_i \in T_M^O} \{(t_i, p_s)\}$.

For example, Petri net $N_M = (\{p_s\}, \{t_1, t_3\} \cup \{t_2, t_4\}, F_M)$ depicted in Figure 3(b) is the mutual exclusion supervisor of the two MESRs depicted in Figures 2(b) and 2(c), respectively.

Actually, a $k$-derivation process is a virtual resource expansion process. It can derive a resource to $k$ ($k > 0$) resources. The common resource can be virtually derived to $k$ virtual resources by the $k$-derivation process. Similarly, a $k$-convergence process is a virtual resource constriction process. It can converge $k$ resources to a resource. The $k$ virtual resources can be converged to a common resource by the $k$-convergence process. Therefore, the methods of $k$-derivation process and $k$-convergence process can be used to extend general mutual exclusion supervisors to control periodic mutual exclusion behavior.

*Definition 10.* Let $N$ be an MESR, $N_M = (\{p\}, T_M^I \cup T_M^O, F_M)$ be the mutual exclusion supervisor of $N$ with $T_M^I = \{t_1, t_2, \ldots, t_n\}$ and $T_M^O = \{t_1', t_2', \ldots, t_n'\}$, $P_B^D = \{p_b^1, p_b^2, \ldots, p_b^n\}$ and $P_B^C = \{p_b'^1, p_b'^2, \ldots, p_b'^n\}$ be two sets of places such that $p_b^i \neq p_b^j$ and $p_b'^i \neq p_b'^j$, and $T_B^D = \{t_b^1, t_b^2, \ldots, t_b^n\}$ and $T_B^C = \{t_b'^1, t_b'^2, \ldots, t_b'^n\}$ be two sets of transitions such that $t_b^i \neq t_b^j$ and $t_b'^i \neq t_b'^j$, where $i \neq j$, $i, j \in \{1, 2, \ldots, n\}$, and $n \geq 2$. Then, Petri net $N_S = (\{p\} \cup P_B^D \cup P_B^C, T_M^I \cup T_M^O \cup T_B^D \cup T_B^C, F_S, W_S)$ is called the periodic mutual exclusion supervisor of $N$ if, $\forall i \in \{1, 2, \ldots, n\}$, $N_d^i$ and $N_c^i$ satisfy the notion that

(i) Petri net $N_d^i = (\{p\} \cup \{p_b^i\}, \{t_i\} \cup \{t_b^i\}, F_d^i, W_d^i)$ is a $k$-derivation process from $p$ to $t_i$;

(ii) Petri net $N_c^i = (\{p\} \cup \{p_b'^i\}, \{t_i'\} \cup \{t_b'^i\}, F_c^i, W_c^i)$ is a $k$-convergence process from $t_i'$ to $p$.

$k$ ($k \geq 1$) is the derivation coefficient and convergence coefficient of $N_d^i$ and $N_c^i$, respectively.

The main idea of designing periodic mutual exclusion supervisors is to derive a common resource to several virtual resources by the preliminary $k$-derivation processes and these derived virtual resources will be converged to a common resource by the final $k$-convergence processes. Note that the $k$-derivation processes and the $k$-convergence processes

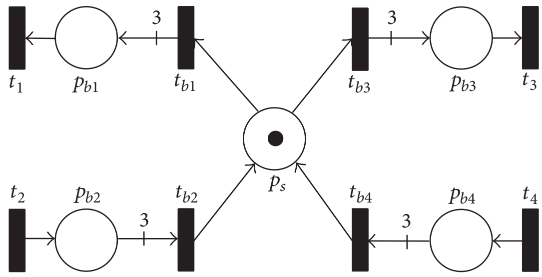


Figure 5: The periodic mutual exclusion supervisor of the two MESRs depicted in Figures 2(b) and 2(c).

come in pairs to derive and converge the common resource, respectively, in the periodic mutual exclusion supervisor of any MESR.

Figure 5 shows the periodic mutual exclusion supervisor $N_S = (\{p\} \cup P_B^D \cup P_B^C, T_M^I \cup T_M^O \cup T_B^D \cup T_B^C, F_S, W_S)$ for the two MESRs depicted in Figures 2(b) and 2(c), where $p = p_s$, $P_B^D = \{p_{b1}, p_{b3}\}$, $P_B^C = \{p_{b2}, p_{b4}\}$, $T_M^I = \{t_1, t_3\}$, $T_M^O = \{t_2, t_4\}$, $T_B^D = \{t_{b1}, t_{b3}\}$, and $T_B^C = \{t_{b2}, t_{b4}\}$. $N_1 = (\{p_s\} \cup \{p_{b1}\}, \{t_1\} \cup \{t_{b1}\}, F_1, W_1)$ is a three-derivation process from $p_s$ to $t_1$ and $N_2 = (\{p_s\} \cup \{p_{b2}\}, \{t_2\} \cup \{t_{b2}\}, F_2, W_2)$ is a three-convergence process from $t_2$ to $p_s$. Similarly, $N_3 = (\{p_s\} \cup \{p_{b3}\}, \{t_3\} \cup \{t_{b3}\}, F_3, W_3)$ is a three-derivation process from $p_s$ to $t_3$ and $N_4 = (\{p_s\} \cup \{p_{b4}\}, \{t_4\} \cup \{t_{b4}\}, F_4, W_4)$ is a three-convergence process from $t_4$ to $p_s$.

For a given MESR $N$ with $N = (P_K \cup P_R \cup P_L \cup \{p_s\}, T, F)$, it may contain the behavior of periodicity mutual exclusion. This behavior contains two important features, that is, mutual exclusion feature and periodic feature. The proposed approach to design the periodic mutual exclusion supervisor for MESR $N$ can be decomposed into two main steps as follows:

(1) Extracting the mutual exclusion supervisor $N_M = (\{p\}, T_M^I \cup T_M^O, F_M)$ from the given MESR $N$ according to Definition 9

(2) Converting the extracted mutual exclusion supervisor $N_M$ into the periodic mutual exclusion supervisor $N_S = (\{p\} \cup P_B^D \cup P_B^C, T_M^I \cup T_M^O \cup T_B^D \cup T_B^C, F_S, W_S)$ by using the paired $k$-derivation processes and $k$-convergence processes according to Definition 10.

After obtaining the periodic mutual exclusion supervisor $N_S$, the periodic mutual exclusion behavior of MESR $N$ can be controlled by combining the periodic mutual exclusion supervisor $N_S$ with the MESR $N$.

*Definition 11.* Let $N$ be an MESR with $N = (P_K \cup P_R \cup P_L \cup \{p_s\}, T, F)$ and $N_S = (\{p\} \cup P_B^D \cup P_B^C, T_M^I \cup T_M^O \cup T_B^D \cup T_B^C, F_S, W_S)$ be the periodic mutual exclusion supervisor of $N$, where $T_M^I \subseteq P_R^\bullet$ and $T_M^O \subseteq {}^\bullet P_R$. Petri net $N_C = (P_C, T_C, F_C, W_C)$ is called the enhanced MESR if $P_C = P_K \cup P_R \cup P_L \cup \{p_s\} \cup \{p\} \cup P_B^D \cup P_B^C$, $T_C = T \cup T_B^D \cup T_B^C$, $F_C = F \cup F_S$, $\forall f \in F_S$ s.t. $W_C(f) = W_S(f)$, and $\forall f' \in F$ s.t. $W_C(f') = 1$.

In an enhanced MESR, all of its linearity processes will compete for a common resource with periodic mutual exclusion behavior. However, the periodic mutual exclusion behavior is controlled by its periodic mutual exclusion supervisor. One of its all linearity processes can consecutively obtain the common resource with $k$ times from the following $k$ competitions, where $k \geq 1$ is an integer. All of its linearity processes will compete for the common resource again after this linearity process releases the common resource.

## 5. Discussion

Let $N^1, N^2, \ldots$ and $N^n$ with $N^i = (P_K^i \cup \{p_r^i\} \cup \{p_l^i\} \cup \{p_s^i\}, T^i, F^i)$ ($\forall i \in \{1, 2, \ldots, n\}$ and $n \geq 2$) be $n$ linearity processes with $n$ resources, $N = (P_K \cup P_R \cup P_L \cup \{p_s\}, T, F)$ be an MESR that is composed of the $n$ linearity processes, and $M_0$ be the initial marking of $N$. Then, the permutation [45] of executing the $n$ T-Processes of the $n$ linearity processes in $N$, denoted as $\psi(N)$, is discussed as follows:

(i) If $N$ is represented with dispersive pattern, we assume $P_R = \{p_r\}$. Then, $p_r = p_r^1 = p_r^2 = \cdots = p^n$, and $|p_r^\bullet| = n$. In the MESR $N$, $\forall t \in p_r^\bullet$, the linearity process that contains transition $t$ will be executed if $t$ fires. The firing of transition $t$ will take away a token from $p_s$ and a token from $p_r$. Therefore, each token contained in place $p_r$ may be taken away by one of the $n$ transitions in $p_r^\bullet$. Let $M_0(p_r) > 0$. According to the methods of combinatorics [45], we assume that all tokens contained in $p_r$ are taken away by the $n$ transitions. The permutation of executing the $n$ T-Processes of the $n$ linearity processes in $N$ is a full permutation that is

$$\psi(N) = \overbrace{n \cdot n \cdot \ldots \cdot n}^{M_0(p_r)} = n^{M_0(p_r)}. \quad (2)$$

(ii) If $N$ is represented with parallel or polymeric pattern, we have $|p_r^{i\bullet}| = 1$, $p_r^i \neq p_r^j$, $i \neq j$, $\forall i, j \in \{1, 2, \ldots, n\}$, and $\bigcap_{i=1}^n p_r^{i\bullet} = \emptyset$. Then, $\forall i \in \{1, 2, \ldots, n\}$, all tokens contained in place $p_r^i$ only can be taken away by the transitions contained in $p_r^{i\bullet}$. Let $M_0(p_r^i) > 0$, where $i \in \{1, 2, \ldots, n\}$. According to the methods of combinatorics [45], a multiset $B = \{M_0(p_r^1) \cdot N^1, M_0(p_r^2) \cdot N^2, \ldots, M_0(p_r^n) \cdot N^n\}$ is constructed. Therefore, the permutation of executing the $n$ T-Processes of the $n$ linearity processes in $N$ is also a full permutation that is

$$\psi(N) = \frac{\left(\sum_{i=1}^n M_0\left(p_r^i\right)\right)!}{M_0\left(p_r^1\right)! \cdot M_0\left(p_r^2\right)! \cdot \ldots \cdot M_0\left(p_r^n\right)!}. \quad (3)$$

If the MESR $N$ is supervised by a periodic mutual exclusion supervisor $N_S$ with $N_S = (\{p\} \cup P_B^D \cup P_B^C, T_M^I \cup T_M^O \cup T_B^D \cup T_B^C, F_S, W_S)$, the periodic mutual exclusion behavior will be controlled. Let $k$ be the derivation coefficient and emergence coefficient of the $k$-derivation processes and the $k$-convergence processes in the periodic mutual exclusion supervisor $N_S$, respectively, where $k \geq 1$ is an integer. In the enhanced MESR $N_C$ that is constructed by combining the MESR $N$ with the periodic mutual exclusion supervisor $N_S$, the permutation of executing the $n$ T-Processes of the $n$ linearity processes in $N_C$, denoted as $\psi(N_C)$, is discussed as follows:

(i) If $N_C$ is represented with dispersive pattern, we have $|P_R| = 1$ and $|p_R^\bullet| = n$. We assume $P_R = \{p_r\}$. In the enhanced MESR $N_C$, $\forall t \in p_r^\bullet$, each firing of $t$ will continuously take away $k$ tokens from $p_r$ once $t$ obtains the common resource since it will continuously fire $k$ times. Let $M_0(p_r) > 0$. The tokens contained in place $p_r$ can supply the firing of these transitions contained in $p_r^\bullet$ with $\lceil M_0(p_r)/k \rceil$ times, where $\lceil x \rceil$ is the ceiling function of $x$. According to the methods of combinatorics [45], the permutation of executing the $n$ T-Processes of the $n$ linearity processes in $N_C$ is

$$\psi(N_C) = \overbrace{n \cdot n \cdot \ldots \cdot n}^{\lceil M_0(p_r)/k \rceil} = n^{\lceil M_0(p_r)/k \rceil}. \quad (4)$$

(ii) If $N$ is represented with parallel or polymeric pattern, similarly, we have $|p_r^{i\bullet}| = 1$, $p_r^i \neq p_r^j$, $i \neq j$, $\forall i, j \in \{1, 2, \ldots, n\}$, and $\bigcap_{i=1}^n p_r^{i\bullet} = \emptyset$. Then, $\forall i \in \{1, 2, \ldots, n\}$, all tokens contained in place $p_r^i$ only can be taken away by the transitions contained in $p_r^{i\bullet}$. Let $M_0(p_r^i) > 0$, where $i \in \{1, 2, \ldots, n\}$. Then, $\forall i \in \{1, 2, \ldots, n\}$, all tokens contained in place $p_r^i$ can supply the firing of these transitions contained in $p_r^{i\bullet}$ with $\lceil M_0(p_r^i)/k \rceil$ times, where $\lceil x \rceil$ is the ceiling function of $x$. According to the methods of combinatorics [45], a multiset $B = \{\lceil M_0(p_r^1)/k \rceil \cdot N^1, \lceil M_0(p_r^2)/k \rceil \cdot N^2, \ldots, \lceil M_0(p_r^n)/k \rceil \cdot N^n\}$ is constructed. Therefore, the permutation of executing the $n$ T-Processes of the $n$ linearity processes in $N_C$ is

$$\psi(N_C) = \frac{\left(\sum_{i=1}^n \lceil M_0(p_r^i)/k \rceil\right)!}{\lceil M_0(p_r^1)/k \rceil! \cdot \lceil M_0(p_r^2)/k \rceil! \cdot \ldots \cdot \lceil M_0(p_r^n)/k \rceil!}. \quad (5)$$

Let $M_0(p_r) = k \cdot x$, where $k \geq 1$ and $x \geq 1$ are two integers. For the MESR $N$ with dispersive pattern and its enhanced MESR $N_C$, according to (2) and (4), we have

$$\begin{aligned} \psi(N) &= \overbrace{n \cdot n \cdot \ldots \cdot n}^{M_0(p_r)} = n^{M_0(p_r)} = n^{k \cdot x}, \\ \psi(N_C) &= \overbrace{n \cdot n \cdot \ldots \cdot n}^{\lceil M_0(p_r)/k \rceil} = n^{\lceil M_0(p_r)/k \rceil} = n^x. \end{aligned} \quad (6)$$

Let, $\forall i \in \{1, 2, \ldots, n\}$, $M_0(p_r^i) = k \cdot x$, where $k \geq 1$ and $x \geq 1$ are two integers. For the MESR $N$ with parallel or polymeric pattern and its enhanced MESR $N_C$, according to (3) and (5), we have

$$\begin{aligned} \psi(N) &= \frac{\left(\sum_{i=1}^n M_0(p_r^i)\right)!}{M_0(p_r^1)! \cdot M_0(p_r^2)! \cdot \ldots \cdot M_0(p_r^n)!} = \frac{(n \cdot k \cdot x)!}{n \cdot (k \cdot x)!}, \\ \psi(N_C) &= \frac{\left(\sum_{i=1}^n \lceil M_0(p_r^i)/k \rceil\right)!}{\lceil M_0(p_r^1)/k \rceil! \cdot \lceil M_0(p_r^2)/k \rceil! \cdot \ldots \cdot \lceil M_0(p_r^n)/k \rceil!} = \frac{(n \cdot x)!}{n \cdot x!}. \end{aligned} \quad (7)$$

It is clear that $\psi(N) \geq \psi(N_C)$, where $\psi(N) = \psi(N_C)$ if $k = 1$. This means that many undesirable execution sequences of the $n$ T-Processes in $N_C$ are forbidden compared with that in $N$. The reduced parts are due to the control of the periodic mutual exclusion supervisor $N_S$ for $N_C$.

## 6. Example

In real world applications, periodic mutual exclusion behavior widely exists in many discrete event systems, such as game server systems, flexible manufacturing systems, intelligent security inspection systems, and other resource distribution systems. The server systems of many online team games suffer from periodic mutual exclusion problems. Figure 6(a) shows a server system of QQ Chinese Poker that is a popular online team game in China. It is constructed with two local servers, called Server 1 and Server 2, that are geographically distributed in different regions. All players need to visit the server system through the Internet and log in the server system. The load balancer of the server system should reasonably distribute all legal players into the two local servers unless they choose Server 1 or Server 2 themselves. The load balancer contains two main distribution rules that are concluded as follows:

(1) Players should join a team if they want to play a game and each team consists of four players. A team can be created only when all four players participate in the same server system. Then, the four players can play a game together.

(2) A player can join only one team at a time.

Figure 6(b) shows the Petri net model of the server system. Figure 6(c) shows an MESR with dispersive pattern, denoted as $N = (\{p_3, p_6\} \cup \{p_2\} \cup \{p_4, p_7\} \cup \{p_s\}, T, F)$, which is a subsystem of the server system. It contains two linearity processes $N^1 = (\{p_3\} \cup \{p_2\} \cup \{p_4\}, \{t_2, t_3\}, F^1)$ and $N^2 = (\{p_6\} \cup \{p_2\} \cup \{p_7\}, \{t_6, t_7\}, F^2)$ that represent the two distribution processes of the two server systems. The two linearity processes share a common resource $p_s$ that represents the load balancer. The two T-Processes of the two linearity processes $N^1$ and $N^2$ are $t_2 t_3$ and $t_6 t_7$, respectively. The nodes of the MESR depicted in Figure 6(c) are illustrated as follows:

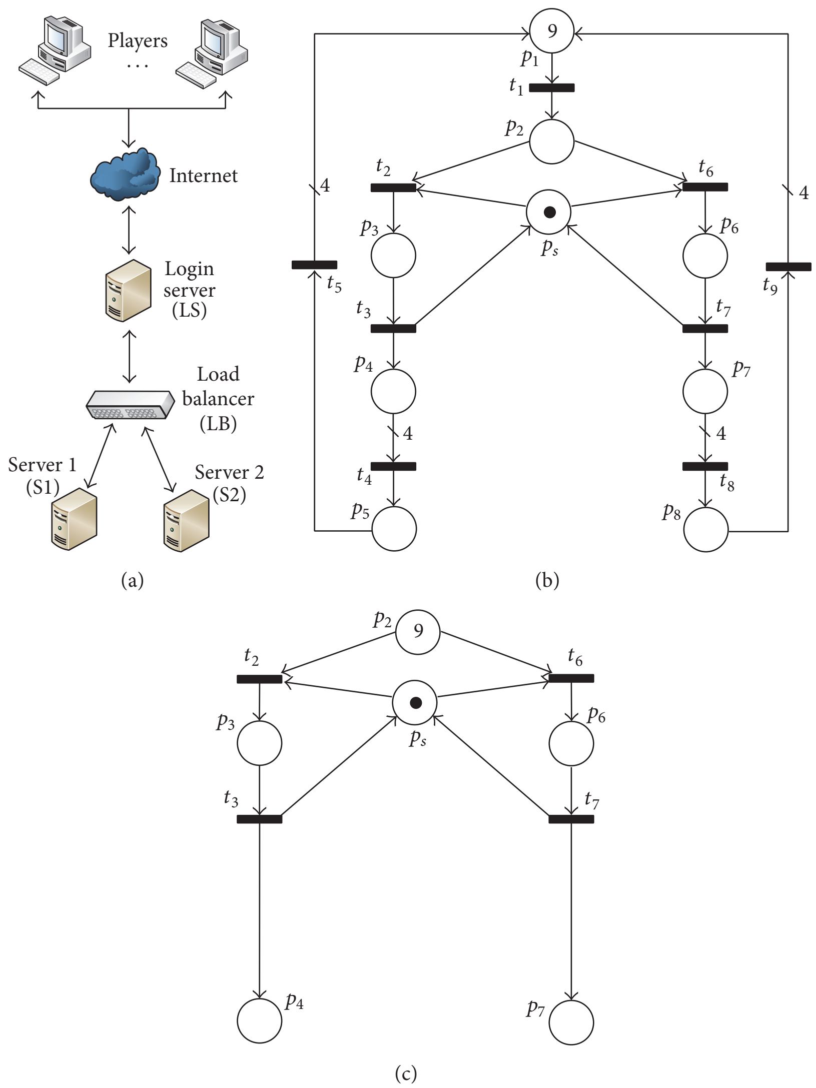


Figure 6: (a) The topology group of the server system of QQ Chinese Poker, (b) the Petri net model of the server system, and (c) the MESR of the server system.

(i) Place $p_2$ is the root of the MESR $N$. It represents the notion that all players are waiting for the distribution of the load balancer.

(ii) Place $p_3$ (resp., $p_6$) represents the notion that a player is being distributed to Server 1 (resp., Server 2).

(iii) Place $p_4$ (resp., $p_7$) represents the notion that a player has been distributed to Server 1 (resp., Server 2).

(iv) Transition $t_2$ (resp., $t_6$) represents the notion that the load balancer begins to distribute a player to Server 1 (resp., Server 2).

(v) Transition $t_3$ (resp., $t_7$) represents the notion that the load balancer finished distributing a player to Server 1 (resp., Server 2).

We can assume that there are nine players that are waiting for the distribution of the load balancer in MESR $N$. This means $M_0(p_2) = 9$ and $M_0(p_s) = 1$, where $M_0$ is the initial marking of $N$. If the periodic mutual exclusion behavior is ignored in the MESR $N$, according to (2), the permutation of executing the two T-Processes in $N$ is

$$\psi(N) = \overbrace{n \cdot n \cdot \ldots \cdot n}^{M_0(p_r)} = n^{M_0(p_r)} = 2^9 = 512. \tag{8}$$

This means that there are 512 distribution results for the nine players. For example, a distribution result is constituted by the six T-Process $t_2t_3$ and three T-Process $t_6t_7$. This distribution result represents the notion that six players are distributed into Server 1 and only three players are distributed into Server 2. In this distribution result, only one team can be created and only four players can play game in Server 1 at this time according to the distribution rules. Figure 7(a) shows the firing sequence of the six T-Process $t_2t_3$ and three T-Process $t_6t_7$. Similarly, Figure 7(b) shows another distribution result that is constituted by three T-Process $t_2t_3$ and six T-Process $t_6t_7$. In this distribution result, only one team can be created and only four players can play a game in Server 2 at this time.

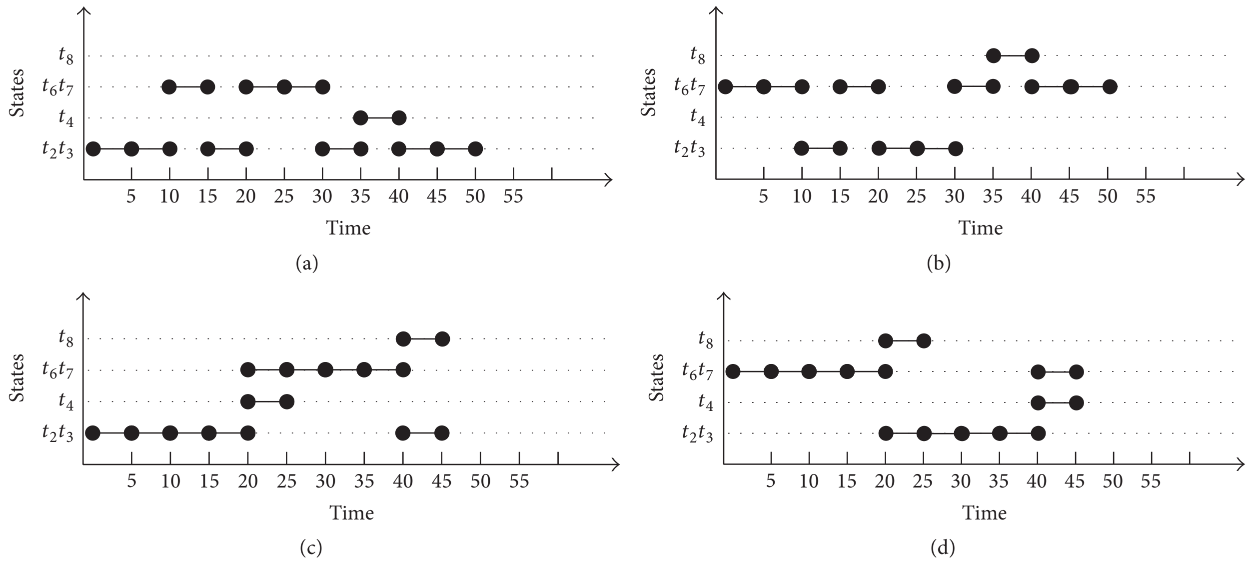


Figure 7: The firing sequences of (a) six T-Process $t_2t_3$ and three T-Process $t_6t_7$, (b) three T-Process $t_2t_3$ and six T-Process $t_6t_7$, (c) five T-Process $t_2t_3$ and four T-Process $t_6t_7$, and (d) four T-Process $t_2t_3$ and five T-Process $t_6t_7$.

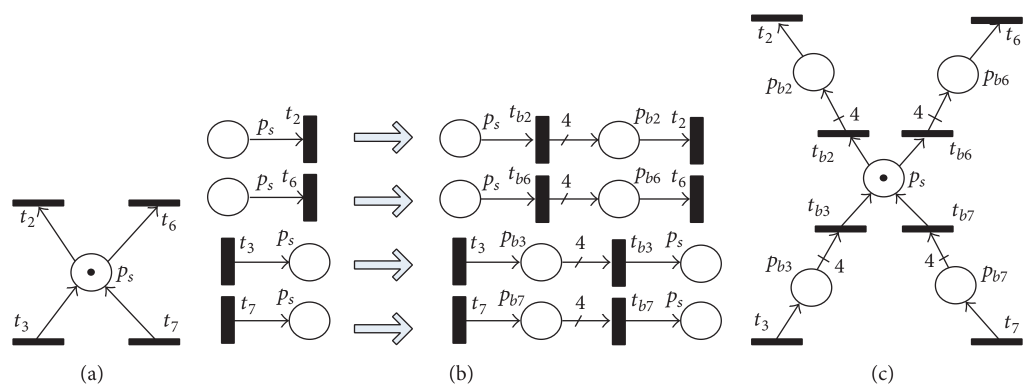


Figure 8: (a) The mutual exclusion supervisor $N_M$, (b) the conversion processes from $N_M$ to the periodic mutual exclusion supervisor $N_S$, and (c) the converted periodic mutual exclusion supervisor $N_S$.

The two distribution results are unreasonable since there are five players that cannot be distributed into a team to play a game. They are only two of the 512 distribution results. There are many similar unreasonable distribution results in the 512 distribution results, which should be controlled by a periodic mutual exclusion supervisor.

According to the distribution rules, the MESR has the periodic mutual exclusion behavior. In fact, there is an ideal result that two teams can be created and eight players can play a game at the same time for the nine players. The distribution result depicted in Figure 7(c) is an ideal distribution result that is constituted by five T-Process $t_2t_3$ and four T-Process $t_6t_7$. In this distribution result, eight players can play a game at this time, where a team is created in Server 1 and another team is created in Server 2. Similarly, Figure 7(d) shows also an ideal distribution result that is constituted by four T-Process $t_2t_3$ and five T-Process $t_6t_7$. In the two distribution results, only one player is waiting to join a team. We can see that the last two distribution results are better than the previous two distribution results. Therefore, this server system is not optimal.

In order to optimize this server system, we stipulate that if the load balancer distributes a player into Server 1 (resp., Server 2), the following three players should be also distributed into Server 1 (resp., Server 2). This is a classical periodic mutual exclusion problem. Therefore, a periodic mutual exclusion supervisor can be designed to control the load balancer to reasonably distribute players into Server 1 or Server 2.

For the MESR $N$ with $N = (\{p_3, p_6\} \cup \{p_2\} \cup \{p_4, p_7\} \cup \{p_s\}, T, F)$, we can extract the mutual exclusion supervisor $N_M = (\{p_s\}, \{t_2, t_6\} \cup \{t_3, t_7\}, F_M)$ from $N$ according to Definition 9. Figure 8(a) shows the extracted mutual exclusion supervisor $N_M$. Next, we can convert the extracted mutual exclusion supervisor $N_M$ into the periodic mutual exclusion supervisor $N_S = (\{p_s\} \cup \{p_{b2}, p_{b6}\} \cup \{p_{b3}, p_{b7}\}, \{t_2, t_6\} \cup \{t_3, t_7\} \cup$

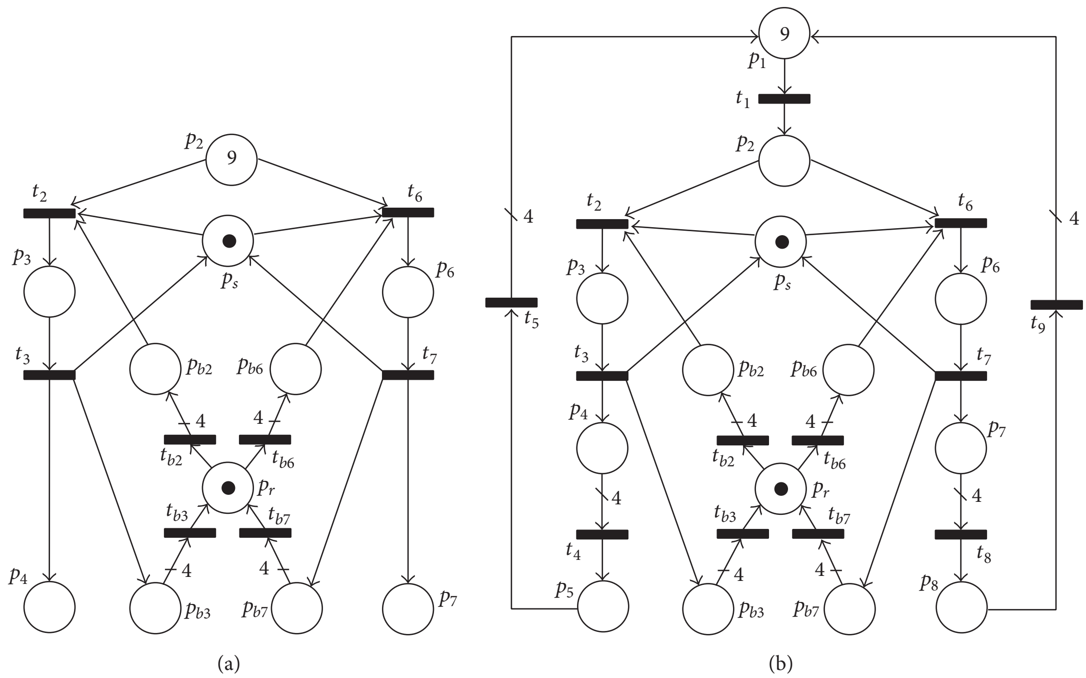


Figure 9: (a) The enhanced MESR $N_C$ with dispersive pattern and (b) the complete control system with periodic mutual exclusion supervisor $N_S$.

$\{t_{b2}, t_{b6}\} \cup \{t_{b3}, t_{b7}\}, F_S, W_S)$ by using two paired $k$-derivation processes and $k$-convergence processes according to Definition 10, where the derivation coefficient and convergence coefficient are four; that is, $k = 4$. Figure 8(b) shows the details of the conversion processes. The conversion processes from $p_s$ to $t_2$ and from $p_s$ to $t_6$ are two four-derivation processes and the conversion processes from $t_3$ to $p_s$ and from $t_7$ to $p_s$ are two four-convergence processes. Figure 8(c) shows the converted periodic mutual exclusion supervisor $N_S$.

After obtaining the periodic mutual exclusion supervisor $N_S$, an enhanced MESR $N_C$ with dispersive pattern can be constructed according to Definition 11. Figure 9(a) shows the enhanced MESR $N_C$. Transitions $t_2$ and $t_6$ of the two T-Processes are mutually exclusive. According to (4), the permutation for executing the two T-Processes in $N_C$ is

$$\psi(N_C) = \overbrace{n \cdot n \cdot \ldots \cdot n}^{\lceil M_0(p_r)/k \rceil} = n^{\lceil M_0(p_r)/k \rceil} = 2^{\lceil 9/4 \rceil} = 2^3 = 8. \quad (9)$$

This means that there are eight distribution results for the nine players in the enhanced MESR $N_C$. It contains the two distribution results depicted in Figures 7(c) and 7(d). Comparing with the general MESR $N$, about 504 (98.44%) undesirable distribution results are forbidden by the periodic mutual exclusion supervisor $N_S$. The complete control system with periodic mutual exclusion supervisor $N_S$ is shown in Figure 9(b). For the nine players in the complete control system, two teams can be created and eight players can play a game. Only one player needs to wait for other players to visit this server.

Periodic mutual exclusion problems also exist in flexible manufacturing systems. For example, let a flexible manufacturing system contain a robot, two conveyors, and two machines. The robot is responsible for moving a part from one of the two conveyors to a corresponding machine. The two machines can use four parts to produce a product, respectively. This flexible manufacturing system will suffer from periodic mutual exclusion problems. If there are only nine parts that will be used to produce products, an ideal result is that two products are produced by eight parts. However, it is possible in this flexible manufacturing system that only one product is produced by four of the nine parts. For the other five parts, two parts are distributed to a machine and three parts are distributed to another machine. Therefore, it is feasible to design a periodic mutual exclusion supervisor to control the periodic mutual exclusion behavior of this flexible manufacturing system. Moreover, intelligent security inspection systems also suffer from periodic mutual exclusion problems. An effective measure to solve the heavy congestion problems of public transportation stations is to quickly disperse passengers. Generally, several familiar passengers will adjacently stand in the same queue to go through a security inspection system. They will leave the station after all of them go through the security inspection system. When all passengers stand in several queues to go through a security inspection system, it is a typical periodic mutual exclusion problem. If a passenger goes through the security inspection system, the adjacent passengers who stand in the same queue with the previous one should preferentially go through the security inspection system since they may be familiar with each other. Therefore, the proposed periodic mutual exclusion supervisors can control the periodic mutual exclusion behavior in the security inspection system to improve the congestion problems as far as possible. Furthermore, many resource distribution systems also suffer from periodic mutual exclusion problems. The proposed methodology can be extended to design periodic mutual exclusion supervisors to control these periodic mutual exclusion problems.

## 7. Conclusions

In this paper, periodic mutual exclusion behavior is controlled by the proposed periodic mutual exclusion supervisors in discrete event systems. The proposed $k$-derivation processes and $k$-convergence processes can forbid the undesirable execution sequences. The purpose is to optimize the control ability of the designed periodic mutual exclusion supervisors. The example results illustrate that a common resource can be derived to several virtual resources by using a $k$-derivation process. Similarly, several virtual resources can be converged into a common resource by using a $k$-convergence process after the executing process releases all virtual resources. The performance of the enhanced MESRs is optimized by using the proposed periodic mutual exclusion supervisors to control the distribution of the common resources. In future work, the performance of the proposed periodic mutual exclusion supervisors should be simulated and verified by developing a simulating tool. The construction method of periodic mutual exclusion supervisors should be optimized by constructing linear constraints based on some extended generalized mutual exclusion constraints.

## Competing Interests

The authors declare that they have no competing interests.

## Acknowledgments

This work was supported in part by the Foundations of Sichuan Educational Committee under Grants nos. 16ZA0158 and 15ZB0134, the Key Scientific Research Fund of Xihua University under Grant no. Z1512625, the Open Research Subject of Key Laboratory (Research Base) of Network Intelligent Information Processing under Grant no. szjj2016-045, the "Chun hui" Research Funds for Educational Department of China under Grant no. Z2015100, and the Scientific Research Funds Project of Science and Technology Department of Sichuan Province under Grant no. 2016JY244.

## References

[1] Z. W. Li and M. C. Zhou, "Elementary siphons of Petri nets and their application to deadlock prevention in flexible manufacturing systems," *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, vol. 34, no. 1, pp. 38–51, 2004.

[2] F. Basile and P. Chiacchio, "On the implementation of supervised control of discrete event systems," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 4, pp. 725–739, 2007.

[3] S. Song II, Y. H. Kim, and J. S. Yoo, "An enhanced concurrency control scheme for multidimensional index structures,," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 1, pp. 97–111, 2004.

[4] P. S. Yu and D. M. Dias, "Performance analysis of concurrency control using locking with deferred blocking," *IEEE Transactions on Software Engineering*, vol. 19, no. 10, pp. 982–996, 1993.

[5] M. C. Zhou and F. DiCesare, "Parallel and sequential mutual exclusions for Petri net modeling of manufacturing systems with shared resources," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 4, pp. 515–527, 1991.

[6] L. Lamport, "A fast mutual exclusion algorithm," *ACM Transactions on Computer Systems*, vol. 5, no. 1, pp. 1–11, 1987.

[7] J. B. Jørgensen and L. M. Kristensen, "Computer aided veri*fi*cation of lamport's fast mutual exclusion algorithm using coloured petri nets and occurrence graphs with symmetries," *IEEE Transactions on Parallel and Distributed Systems*, vol. 10, no. 7, pp. 714–732, 1999.

[8] J. Czyzowicz, L. Gasieniec, D. R. Kowalski, and A. Pec, "Consensus and mutual exclusion in a multiple access channel," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 7, pp. 1092–1104, 2011.

[9] R. Proietti, C. J. Nitta, Y. Yin, R. Yu, S. J. B. Yoo, and V. Akella, "Scalable and distributed contention resolution in AWGR-based data center switches using RSOA-based optical mutual exclusion," *IEEE Journal on Selected Topics in Quantum Electronics*, vol. 19, no. 2, p. 3600111, 2013.

[10] W. G. Wu, J. B. Zhang, A. Luo, and J. N. Cao, "Distributed mutual exclusion algorithms for intersection traffic control," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 1, pp. 65–74, 2015.

[11] H. Attiya, A. Kogan, and J. L. Welch, "Efficient and robust local mutual exclusion in mobile Ad Hoc networks," *IEEE Trans. Mobile Computing*, vol. 9, no. 3, pp. 361–375, 2009.

[12] P. J. Ramadge and W. M. Wonham, "Supervisory control of a class of discrete event processes," *SIAM Journal on Control and Optimization*, vol. 25, no. 1, pp. 206–230, 1987.

[13] W. M. Wonham and P. J. Ramadge, "On the supremal controllable sublanguage of a given language," *SIAM Journal on Control and Optimization*, vol. 25, no. 3, pp. 637–659, 1987.

[14] P. J. Ramadge and W. M. Wonham, "Modular feedback logic for discrete event systems," *SIAM Journal on Control and Optimization*, vol. 25, no. 5, pp. 1202–1218, 1987.

[15] P. J. G. Ramadge and W. M. Wonham, "Control of discrete event systems," *Proceedings of the IEEE*, vol. 77, no. 1, pp. 81–98, 1989.

[16] H. Flordal, R. Malik, M. Fabian, and K. Akesson, "Compositional synthesis of maximally permissive supervisors using supervision equivalence," *Discrete Event Dynamic Systems*, vol. 17, no. 4, pp. 475–504, 2007.

[17] Y. Chen, Z. Li, M. Khalgui, and O. Mosbahi, "Design of a maximally permissive liveness-enforcing Petri net supervisor for flexible manufacturing systems," *IEEE Transactions on Automation Science and Engineering*, vol. 8, no. 2, pp. 374–393, 2011.

[18] A. Nazeem, S. Reveliotis, Y. Wang, and S. Lafortune, "Designing compact and maximally permissive deadlock avoidance policies for complex resource allocation systems through classification theory: the linear case," *IEEE Transactions on Automatic Control*, vol. 56, no. 8, pp. 1818–1833, 2011.

[19] K. Schmidt and C. Breindl, "Maximally permissive hierarchical control of decentralized discrete event systems," *IEEE Transactions on Automatic Control*, vol. 56, no. 4, pp. 723–737, 2011.

[20] Y. Chen, Z. Li, and M. Zhou, "Behaviorally optimal and structurally simple liveness-enforcing supervisors of flexible manufacturing systems," *IEEE Transactions on Systems, Man,*

*and Cybernetics Part A: Systems and Humans*, vol. 42, no. 3, pp. 615–629, 2012.

[21] B. Huang, M. C. Zhou, G. X. Zhang, A. C. Ammari, A. Alabdulwahab, and A. G. Fayoumi, "Lexicographic multiobjective integer programming for optimal and structurally minimal petri net supervisors of automated manufacturing systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 11, pp. 1459–1470, 2015.

[22] M. Bashir, B. B. Muhammad, and Z. Li, "Minimal supervisory structure for flexible manufacturing systems using Petri nets," in *Proceedings of the 2nd International Conference on Control, Automation and Robotics (ICCAR '16)*, April 2016.

[23] M. C. Zhou, F. Dicesare, and D. L. Rudolph, "Design and implementation of a petri net based supervisor for a flexible manufacturing system," *Automatica*, vol. 28, no. 6, pp. 1199–1208, 1992.

[24] J.-S. Lee, "A Petri net design of command filters for semi-autonomous mobile sensor networks," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 4, pp. 1835–1841, 2008.

[25] M. Uzam and A. H. Jones, "Discrete event control system design using automation Petri nets and their ladder diagram implementation," *International Journal of Advanced Manufacturing Technology*, vol. 14, no. 10, pp. 716–728, 1998.

[26] L. E. Holloway and B. H. Krogh, "Synthesis of feedback control logic for a class of controlled Petri nets," *IEEE Transactions on Automatic Control*, vol. 35, no. 5, pp. 514–523, 1990.

[27] A. Ghaffari, N. Rezg, and X. Xie, "Feedback control logic for forbidden-date problems of marked graphs: application to a real manufacturing system," *IEEE Transactions on Automatic Control*, vol. 48, no. 1, pp. 18–29, 2003.

[28] A. Giua and F. DiCesare, "Supervisory design using Petri nets," in *Proceedings of the 30th IEEE Conference on Decision and Control*, vol. 1, pp. 92–97, Brighton, UK, December 1991.

[29] R. S. Sreenivas and B. H. Krogh, "On Petri net models of infinite state supervisors," *IEEE Transactions on Automatic Control*, vol. 37, no. 2, pp. 274–277, 1992.

[30] F.-S. Hsieh and S.-C. Chang, "Dispatching-driven deadlock avoidance controller synthesis for flexible manufacturing systems," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 2, pp. 196–209, 1994.

[31] R. S. Sreenivas, "An application of independent, increasing, Free-Choice Petri nets to the synthesis of policies that enforce liveness in arbitrary Petri nets," *Automatica*, vol. 34, no. 12, pp. 1613–1615, 1998.

[32] K. X. He and M. D. Lemmon, "Liveness-enforcing supervision of bounded ordinary Petri nets using partial order methods," *IEEE Transactions on Automatic Control*, vol. 47, no. 7, pp. 1042–1055, 2002.

[33] M. V. Iordache, J. O. Moody, and P. J. Antsaklis, "Method for the synthesis of deadlock prevention controllers in systems modeled by Petri nets," in *Proceedings of the American Control Conference*, pp. 3167–3171, Cagliari, Italy, June 2000.

[34] M. V. Iordache, J. O. Moody, and P. J. Antsaklis, "A method for the synthesis of liveness enforcing supervisors in Petri nets," in *Proceedings of the American Control Conference*, pp. 4943–4948, Arlington, Tex, USA, June 2001.

[35] A. Ghaffari, N. Rezg, and X. Xie, "Design of a live and maximally permissive Petri net controller using the theory of regions," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 1, pp. 137–141, 2003.

[36] G. Balbo, G. Chiola, S. C. Bruell, and P.-Z. Chen, "An example of modeling and evaluation of a concurrent program using colored stochastic Petri nets: lamport's fast mutual exclusion algorithm," *IEEE Transactions on Parallel and Distributed Systems*, vol. 3, no. 2, pp. 221–240, 1992.

[37] A. Hellgren, M. Fabian, and B. Lennartson, "On the execution of sequential function charts," *Control Engineering Practice*, vol. 13, no. 10, pp. 1283–1293, 2005.

[38] A. Dideban, M. Kiani, and H. Alla, "Implementing PN-based controller with mutually exclusive transitions by SFC," in *Proceedings of the 35th Annual Conference of the IEEE Industrial Electronics Society (IECON '09)*, pp. 4353–4358, November 2009.

[39] A. Giua, F. DiCesare, and M. Silva, "Generalized mutual exclusion contraints on nets with uncontrollable transitions," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, pp. 974–979, Chicago, Ill, USA, October 1992.

[40] J. L. Luo, W. M. Wu, H. Y. Su, and J. Chu, "Supervisor synthesis for enforcing a class of generalized mutual exclusion constraints on Petri nets," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 39, no. 6, pp. 1237–1246, 2009.

[41] Z. Y. Ma, Z. W. Li, and A. Giua, "Design of optimal Petri net controllers for disjunctive generalized mutual exclusion constraints," *IEEE Transactions on Automatic Control*, vol. 60, no. 7, pp. 1774–1785, 2015.

[42] F. Basile, P. Chiacchio, and A. Giua, "An optimization approach to Petri net monitor design," *IEEE Transactions on Automatic Control*, vol. 52, no. 2, pp. 306–311, 2007.

[43] S. Wang, C. Wang, and M. Zhou, "Design of optimal monitor-based supervisors for a class of Petri nets with uncontrollable transitions," *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans*, vol. 43, no. 5, pp. 1248–1255, 2013.

[44] T. Murata, "Petri nets: properties, analysis and applications," *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541–580, 1989.

[45] H. J. Ryser, *Combinatorial Mathematics*, The Carus Mathematical Monographs, No. 14, Wiley, New York, NY, USA, 1963.


Advances in Operations Research
Advances in Decision Sciences
Journal of Applied Mathematics
Algebra
Journal of Probability and Statistics
The Scientific World Journal
Hindawi
Submit your manuscripts at
https://www.hindawi.com
International Journal of Differential Equations
International Journal of Combinatorics
Advances in Mathematical Physics
Journal of Complex Analysis
Journal of Mathematics
Mathematical Problems in Engineering
Abstract and Applied Analysis
Discrete Dynamics in Nature and Society
International Journal of Mathematics and Mathematical Sciences
Journal of Discrete Mathematics
Journal of Function Spaces
International Journal of Stochastic Analysis
Journal of Optimization