On structural reduction of liveness-enforcing Petri net supervisors for flexible manufacturing systems: an algebraic approach

MUHAMMAD BASHIR

School of Electro-Mechanical Engineering, Key Laboratory of Electronic Equipment Structure Design, Xidian University, No. 2 South Taibai Road, Xi’an 710071, China

ZHIWU LI*

Institute of Systems Engineering, Macau University of Science and Technology, Avenida Wai Long, Taipa, Macau SAR
*Corresponding author: zhwli@xidian.edu.cn

MURAT UZAM

Meliksah Universitesi, Muhendislik-Mimarlik Fakultesi Elektrik-Elektronik Muhendisligi Bolumu 38280 Talas, Kayseri, Turkey

NAIQI WU

Institute of Systems Engineering, Macau University of Science and Technology, Avenida Wai Long, Taipa, Macau SAR

AND

ABDULRAHMAN AL-AHMARI

Industrial Engineering Department, College of Engineering, King Saud University, Riyadh 11421, Saudi Arabia and FARCAMT Chair, Advanced Manufacturing Institute, King Saud University, Riyadh 11421, Saudi Arabia

[Received on 27 May 2016; revised on 23 October 2016; accepted on 1 March 2017]

A deluge of studies has been carried out on deadlock prevention and liveness enforcement for flexible manufacturing systems (FMSs). This paper reports an algebraic procedure to find a set of merged place invariants, aiming to reduce the structural complexity of a liveness-enforcing Petri net supervisor for an FMS. Firstly, given an uncontrolled Petri net model, a set of control places as well as their corresponding place invariants are computed by using the existing methods. A systematic approach is developed to find a set of merged place invariants that are much more compact than the original supervisor structure, which remarkably reduces the number of control places in a liveness-enforcing Petri net supervisor for an FMS. The structurally reduced supervisor provides the same or more permissive behavior than that of an original one. As an algebraic approach, the proposed method is computationally trivial and is applicable to all manufacturing-oriented Petri net classes currently available in the literature. Several typical manufacturing examples are used to demonstrate the approach reported in this research.

Keywords: deadlock; flexible manufacturing system (FMS); liveness-enforcing supervisor; Petri net.

1. Introduction

Nowadays, the economic situation of an enterprise depends on the diversified products contributed to the world market, majorly due to revolution of modern industry. For a particular manufacturer to involve a
transaction with the modern world market, the ability of producing different kinds of diversified products is thought of as an essential factor to compete with other competitors. This can be achieved by replacing an old style, fixed hardware sequential system with a flexible manufacturing system (FMS) that can easily make change to the product design by configuring a supervisory controller (Han & Lee, 2005). An FMS usually consists of two main parts: a physical system that is composed of manufacturing resources (such as machine tools, robots and a material-handling system) shared by a number of jobs; and a management system or decision making system responsible for the control of the physical system to achieve the goal of productivity and work-in-process (Han & Lee, 2005). Many advantages can be derived from an FMS. However, due to the high utilization of resources in such a system, deadlocks can occur, which may degrade the performance of an FMS (Han & Lee, 2005; Chen & Li, 2012; Ji & Wang, 2014; Chen et al., 2015, 2017a,b; Hou et al., 2016), and even lead to serious economy loss in highly automated production processes.

Deadlocks are an important issue to be considered in the design and control stage of an FMS, since their occurrences cause a lot of damage to the system or might halt the whole system from the operation as reported by Chen et al. (2011); Li et al. (2007); Liu et al. (2014). In general, Coffman formulates four necessary conditions for deadlock occurrence in a resource allocation system. They are popularly known as Coffman conditions (Chen & Li, 2011; Hu et al., 2011; Li et al., 2012): (1) Mutual exclusion: a resource can only be used by one process at a time; (2) Hold and wait: processes that use some resources may need another new resource; (3) Non-preemption: it is infeasible to remove a resource that is held by a particular process, but a process can only release a resource by an explicit action of that process; and (4) Circular-wait: two or more processes form a circular chain where each process waits for a resource that is held by the next process in the chain. To prevent the occurrence of deadlocks, at least one of the four conditions should be broken.

Several tools have been developed (Chen & Li, 2012, 2013) to deal with deadlocks in FMSs. Petri nets, automata and graph theory are the three main methodologies. This paper copes with structural reduction of a liveness-enforcing supervisor that is expressed by Petri nets. Petri nets have been increasingly used for modeling, scheduling, and supervisory control of discrete event systems (Wang et al., 2009, 2015, 2016; Wu & Zhou, 2012; Wu et al., 2013; Bai et al., 2016; Ma et al., 2016; Tong et al., 2016, 2017; Zhao & Hou, 2013; Mi & Uzam, 2016), particularly on deadlock resolution in an FMS, since they can appropriately describe FMSs’ structural and behavioral (Ezepeleta et al., 1998) properties such as conflicts, concurrency, casual dependency, liveness and boundedness. Furthermore, Petri nets are widely used in contemporary technological systems such as computer and communication networks, manufacturing processes, and automation systems (Abdallah & ElMaraghy, 1998; Li & Zhou, 2006a; Chen et al., 2014a,b, 2015; Liu et al., 2015; Ma et al., 2015; Patel et al., 2015; Wu et al., 2015; Ye et al., 2015; Zhang et al., 2015; Muhammad et al., 2016; Wu et al., 2016; Liu et al., 2015; Yang et al., 2017). Generally, the study in Chen et al. (2011); Guan et al. (2015); Li & Zhou (2008b,c); Li et al. (2012); Uzam et al. (2015) reported that there are four strategies to handle deadlocks in automated flexible manufacturing systems: (1) Deadlock ignoring; (2) Deadlock detection and recovery; (3) Deadlock avoidance, and (4) Deadlock prevention.

In deadlock ignoring, the occurrence of deadlocks is ignored due to the negligible probability. Deadlock detection and recovery allow the occurrence of deadlocks, and when a system detects deadlocks, it can be recovered and bring it back to a normal state by simply reallocating the resources (Jeng et al., 2004; Li & Zhou, 2008c; Pirrodi et al., 2008). Deadlock avoidance determines the possible system evolution at each system state (Wu et al., 2008; Wysk et al., 1994; Li & Zhou, 2008a,b; Uzam et al., 2016; Hao & Xing, 2014) using an online control policy and chooses the correct system evolution paths. Deadlock prevention is usually achieved by using an off-line computational mechanism to control the
request for resources to ensure that deadlocks never occur. In Petri net formalism, a supervisor derived from a deadlock prevention policy is usually composed of monitors (control places) and the transitions of a plant, as well as directed arcs connecting them (Li & Zhou, 2004; Chen et al., 2011, 2015; Li et al., 2012; Chen & Li, 2013; Uzam et al., 2016).

Deadlock prevention policies are widely used due to their advantages that their computation is done off-line and deadlocks can be totally eliminated, i.e., the controlled system can never enter a deadlock state. The performance for a deadlock prevention policy can be evaluated based on the following three criteria (Chen et al., 2011, 2012): (1) behavioral permissiveness, (2) structural complexity, and (3) computational complexity. A maximally permissive supervisor usually leads to sufficient utilization of resources and high system throughput (Chen et al., 2011). A supervisor with the minimal number of control places can decrease both hardware and software cost in the stage of control verification, validation and implementation. A deadlock control policy with low computational complexity (Wang et al., 2012) means that it can be applied to large-sized real-world systems. There are mainly two Petri net-based analysis techniques used for the study on deadlocks (Uzam, 2002; Li et al., 2007; Li & Zhou, 2008c; Chen et al., 2011; Zareiee et al., 2015a; Muhammad et al., 2016): (1) Structural analysis, and (2) reachability graph analysis.

In structural analysis, Petri net components, such as siphons, place and transition invariants, and resource transition circuits, are extensively used, as there are particular corresponding relationships between behavioral properties of a Petri net and its structural components. A deadlock prevention policy derived from siphon control is usually suboptimal, since it is difficult to ensure that every permissive state is included in the controlled system. To prevent a siphon from being insufficiently marked, a monitor called control place is usually used to disable the related transitions in a plant at some particular markings (Huang et al., 2006; Wang et al., 2009; Hu et al., 2011). In general, the number of siphons grows exponentially with the structural size of a net. The works of Li et al. (2007) and Li & Zhou (2008c) developed an elementary-siphon-based approach to reduce the number of siphons to be explicitly controlled. However, this concept does not in general provide a maximally permissive supervisor, unless for some special classes of Petri nets at particular initial markings. The work in Pirrodi et al. (2008) reports a siphon-based deadlock prevention policy that avoids a complete enumeration of siphons. As an improved method, it can reduce computational overheads. In the study of Pirrodi et al. (2009), a selective siphon control technique is proposed and deadlocks are prevented by an iterative way through siphon control. The relations between uncontrolled siphons and critical markings are identified and a set of siphons is selected by solving a set covering problem at each iteration step. The supervisor derived from this method can in most cases provide maximally permissive behavior for a Petri net model. However, it suffers from the computational complexity problem.

The reachability graph (RG) analysis enables one to check certain properties of FMSs, such as liveness, boundedness, synchronization, concurrency and safeness (Huang & Pan, 2011; Chen & Li, 2012). On the other hand, the RG analysis usually requires a complete or partial enumeration of reachable states. Therefore, it suffers from the state explosion problem. The theory of regions is developed by Ghaffari et al. (2003) as one of the most powerful methods of deadlock prevention for deriving a maximally permissive supervisor for FMSs. However, it is computationally expensive since too many inequality constraints have to be considered in the linear programming problems that are used to compute monitors (Uzam, 2004; Huang et al., 2006; Uzam & Zhou, 2007; Li & Zhou, 2008b; Pirrodi et al., 2008; Chen et al., 2011, 2012; Huang & Pan, 2011; Uzam et al., 2015), in addition to the generation of a complete state space. The work in Uzam & Zhou (2006); Uzam et al. (2007) divides the reachability graph of a Petri net model into two components: a live zone (LZ) and a deadlock zone (DZ). The former contains legal markings and the latter contains illegal markings. The partition of an RG is used to find the
first-met bad markings (FBMs) in the DZ such that, once all FBMs are prohibited, all illegal markings in DZ are not reachable. The deadlock prevention methods in Uzam & Zhou (2006); Uzam et al. (2007) are an iterative procedure in which at each iteration step, an FBM is controlled by designing a control place. The iterations are repeated until all FBMs of a Petri net model are forbidden. However, this method does not guarantee maximally permissive behavior of the controlled system.

The method developed by Chen et al. (2011) uses RG-based analysis and presents a maximally permissive liveness-enforcing supervisor, where a control place is designed to forbid an FBM and, at the same time, it keeps all legal markings reachable by solving integer linear programming problems (ILPPs). In addition, a vector covering approach is developed to reduce the sets of legal markings and FBMs by partitioning them into a minimal covering set of legal markings and a minimal covered set of FBMs, respectively. However, the method suffers from structural complexity problems. This work is then improved by finding the minimal number of control places in (Chen & Li, 2012).

Siphon control may generate redundant monitors, as the monitors used to control a siphon may implement the function of previously designed monitors, making them redundant. It is then important to find an effective and efficient method to remove the redundant monitors such that a structurally simple supervisor can be obtained (Zareiee et al., 2015b).

Currently, one of the available methods to reduce the structural complexity of a liveness-enforcing Petri net supervisor in the literature is developed by Chen et al. (2012); Chen & Li (2013) using reachability graph analysis. This method ensures that the derived liveness-enforcing supervisor is maximally permissive if such a supervisor exists. A vector covering approach is employed to compute the minimal covering set of legal markings and the minimal covered set of FBMs. Then, at each iteration step, a control place is designed to prohibit as many FBMs as possible by forming a place invariant associated with the control place and activity places in the plant Petri net model. The coefficients of the place invariant are computed using an integer linear programming problem (ILPP) that ensures the two conditions stated as: (i) no marking in the minimal covering set of legal markings is prohibited; and (ii) the objective function maximizes the number of FBMs that are forbidden by the place invariant. After finite steps of the prohibition of FBMs, a maximally permissive supervisor can be computed if it exists.

Liu et al. (2014) developed a method for merging two or more siphons to reduce the structural complexity of a liveness-enforcing supervisor that provides maximally permissive behavior without using the reachability graph analysis, but depending on siphon control. The proposed method guarantees that two or more siphons can be merged if their forbidden marking sets can be enforced by the same linear invariant constraint. It is an iterative procedure, ensuring the following conditions to be satisfied. First, an unmarked siphon at some marking may correspond to a number of FBMs, among which the non-reachability of a selected marking makes other markings forbidden. For this selected marking, the linear constraint is set to a constant $k$ to become a linear first-order equation. Second, siphon control may correspond to a number of live markings. Only one is selected to make others not forbidden. For this selected marking, the linear constraint is set to a constant $k - 1$ to become a linear first-order equation. Another method that guaranteed the minimality of the supervisor is proposed by Muhammad et al. (2016). The proposed method uses structural analysis to avoid the computation of reachability graph. The main idea of the proposed method is that, each process in the Petri net model can be controlled by a single control place such that the number of control places in the supervisor is the same as the number of processes in the Petri net model. The great advantage of the method is that, it reduces the computational complexity of the supervisor as it uses an algebraic simplification to avoid solving integer linear programming problem. However, the method can only be applied to the safe LS$^3$PR Petri net.

In this paper, a method is proposed to reduce the structural complexity of a liveness-enforcing supervisor of FMSs by using the knowledge of FBMs as well as the monitors. We prohibit multiple
FBMs through computing a generalized mutual exclusion constraint (GMEC) or a place invariant. The proposed method can be applied to large-sized FMSs modelled with different classes of Petri nets, since only algebraic operations are necessary, avoiding a partial or complete siphon or marking enumeration. Typical examples in the literature are used to demonstrate the proposed method.

The remainder of this paper is organized as follows. Basics of Petri nets are provided in Section II. The concept of first-met bad markings and their control are presented in Section III. Section IV reports a method for identifying a set of FBMs that are potentially controlled by a place invariant. In Section V, a method is developed for reducing the structural complexity of a liveness-enforcing supervisor. Applications of the proposed method are presented in Section VI. The complexity of the proposed algorithms is analyzed in Section VII. Finally, Section VIII concludes this paper.

2. Preliminaries

2.1. Petri Nets

A Petri net is a four-tuple \( N = (P, T, F, W) \), where \( P \) and \( T \) are finite and non-empty sets. \( P \) is a set of places and \( T \) is a set of transitions with \( P \cap T = \emptyset \). \( F \subseteq (P \times T) \cup (T \times P) \) is called a flow relation of the net, represented by arcs with places from transitions to transitions or from transitions to places. Places are graphically represented by circles, while transitions by bars or square boxes. \( W : (P \times T) \cup (T \times P) \rightarrow \mathbb{N} \) is a mapping that assigns a weight to an arc: \( W(x, y) > 0 \) if \( (x, y) \in F \), and \( W(x, y) = 0 \), otherwise, where \( (x, y) \in (P \times T) \cup (T \times P) \) is a pair of nodes and \( \mathbb{N} = \{0, 1, 2, \ldots \} \) is the set of non-negative integers. \( N = (P, T, F, W) \) is said to be a Petri net, denoted as \( N = (P, T, F, W) \), if \( \forall f \in F, W(f) = 1 \). Let \( \mathbb{N}^+ = \mathbb{N} \setminus \{0\} \), i.e., \( \mathbb{N}^+ = \{1, 2, \ldots \} \) is a set of positive integers.

Let \( x \in P \cup T \) be a node in \( N = (P, T, F, W) \). The preset of \( x \), denoted by \( ^*x \), is defined as \( ^*x = \{y \in P \cup T | (y, x) \in F\} \) and the postset of \( x \), denoted by \( x^* \), is defined as \( x^* = \{y \in P \cup T | (x, y) \in F\} \). A marking \( M \) of a Petri net \( N = (P, T, F, W) \) is a mapping \( M : P \rightarrow \mathbb{N} \), where \( M \) is an \( n \)-dimensional vector and \( n = |P| \) is the number of places in the net.

Markings and vectors concerned with a Petri net are usually represented via a multiset (bag) or formal sum notation for the sake of an expedient description. As a result, vector \( M \) is denoted by \( \sum_{p \in P} M(p)p \).

For instance, a marking that puts two tokens in place \( p_1 \) and three tokens in place \( p_4 \) only in a net with \( P = \{p_1, p_2, p_3, p_4, p_5, p_6\} \) is denoted as \( 2p_1 + 3p_4 \) instead of \( (2\ 0\ 0\ 3\ 0\ 0)^T \).

Let \( t \in T \) be a transition in \( N = (P, T, F, W) \) at a marking \( M \). Transition \( t \) is said to be enabled if \( \forall p \in ^*t, M(p) \geq W(p, t) \). An enabled transition \( t \) can fire, leading to a new marking \( M' \), i.e., \( \forall p \in P, M'(p) = M(p) - W(p, t) + W(t, p) \). A place \( p \in P \) is said to be \( k \)-bounded if \( \forall M \in R(N, M_0), \exists k \in \mathbb{N} \) \( (k \neq 0), M(p) \leq k \). A net system is said to be \( k \)-bounded if any place is \( k \)-bounded. A place \( p \in P \) is said to be safe if it is 1-bounded. A net is said to be safe if all of its places are safe. A transition \( t \in T \) is said to be live if \( \forall M \in R(N, M_0), \exists M' \in R(N, M_0), M'(t) \) holds. \( (N, M_0) \) is live if \( \forall t \in T, t \) is live at \( M_0 \). \( N \) is dead at \( M_0 \) if \( \exists t \in T, M_0[t] \) holds. Let \( \sigma \) be a transition sequence. The parikh vector of \( \sigma \), denoted by \( \sigma \), is a column vector, represented by \( \sigma = [\#\sigma(t_1), \#\sigma(t_2), \ldots, \#\sigma(t_n)]^T \), where \( n = |T| \) and \( \#\sigma(t_i) \) denotes the number of occurrences of \( t_i \) in \( \sigma \). A P-vector is a column vector \( L : P \rightarrow \mathbb{Z} \), indexed by \( P \), where \( \mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\} \). A P-vector \( I \) is a place invariant if \( I \neq 0 \) and \( I^T[N] = 0^T \). A T-vector is a column vector \( J : T \rightarrow \mathbb{Z} \) indexed by \( T \). A T-vector \( J \) is a transition invariant if \( J \neq 0 \) and \( [N]J = 0 \). The support of a place (transition) invariant \( I(J) \) is denoted by \( ||I|| = |p|I(p) \neq 0 \) \( (||J|| = t|J(t) \neq 0 \). A P-invariant \( I \) is said to be a P-semiflow if \( \forall p \in P, I(p) \geq 0 \). It is called a minimal P-invariant if \( ||I|| \) is not a superset of the support of any other one and its components are mutually.
prime. In what follows, by \( \mathbb{N}_n = \{1, 2, \ldots, n\} \), we denote a set of positive integers. The following results about generalized mutual exclusion constraints (GMECs) are from Giua et al. (1992).

**Definition 1** Let \((N, M_0)\) be a Petri net with \( N = (P, T, F, W) \). A generalized mutual exclusion constraint (GMEC) defines a set of markings 
\[
\mathcal{M}(l, k) = \{ M \in \mathbb{N}^{\lvert P \rvert} | l^T \cdot M \leq k \}
\]
where \( l \) is a non-negative P-vector, \( l(p) \) is called the control coefficient of place \( p \), and \( k \in \mathbb{N}^+ \) is the weighted token constant. The support of the GMEC is the set \( Q_l = \{ p | l(p) > 0 \} \).

A set of GMECs \((l, k)\) with \( l = (l_1, l_2, \ldots, l_m)^T \) and \( k = (k_1, k_2, \ldots, k_m)^T \) defines a set of markings 
\[
\mathcal{M}(l, k) = \cap_{i=1}^m \mathcal{M}(l_i, k_i) = \{ M \in \mathbb{N}^{\lvert P \rvert} | l^T \cdot M \leq k \}
\]

**Definition 2** Let \((N, M_0)\) be a Petri net. A GMEC \((l, k)\) is said to be redundant with respect to a set of markings \( A \subseteq \mathbb{N}^{\lvert P \rvert} \) if \( A \subseteq \mathcal{M}(l, k) \). A GMEC \((l, k)\) is redundant with respect to \((N, M_0)\) if \( R(N, M_0) \subseteq \mathcal{M}(l, k) \). A set of GMECs is redundant with respect to \((N, M_0)\) if \( \forall i \in \{1, 2, \ldots, m\}, (l_i, k_i) \) is redundant.

**Proposition 1** If the following linear programming problem has an optimal solution \( x^* \leq k + 1 \), then the GMEC \((l, k)\) is redundant with respect to \((N, M_0)\):
\[
x = \max l^T \cdot M 
\]
\[
\text{s.t.} \quad M = M_0 + [N] \sigma, M, \sigma \geq 0
\]

The solution to Eq. (1) can be simplified if there exists a basis \( B \) of P-semiflows in a net such as an M-net to be defined in this paper. In this case, Eq. (1) can be transformed into the following linear programming problem:
\[
x = \max l^T \cdot M 
\]
\[
\text{s.t.} \quad B^T \cdot M = B^T \cdot M_0, M \geq 0
\]

**Definition 3** Two sets of GMECs \((l_1, k_1)\) and \((l_2, k_2)\) are equivalent with respect to \((N, M_0)\) if \( R(N, M_0) \cap \mathcal{M}(l_1, k_1) = R(N, M_0) \cap \mathcal{M}(l_2, k_2) \).

**Corollary 1** Two GMECs \((l_1, k_1)\) and \((l_2, k_2)\) are equivalent with respect to \((N, M_0)\) if and only if \((l_1, k_1)\) is redundant with respect to \( R(N, M_0) \cap \mathcal{M}(l_2, k_2) \) and \((l_2, k_2)\) is redundant with respect to \( R(N, M_0) \cap \mathcal{M}(l_1, k_1) \).

By the results aforementioned, we conclude that the equivalence of two sets of GMECs can be checked in polynomial time by using linear programming.
2.2. M-Nets

This subsection describes a class of Petri net called M-nets, more detail of its definition can be found in Li et al. (2012).

An M-net is a Petri net \((N, M_0)\) with \(N = (P, T, F, W)\) that has a place partition with \(P = P_0 \cup P_A \cup P_R\) and a transition partition with \(T = T_1 \cup T_2 \cup \ldots \cup T_n\), where \(\forall i,j \in \mathbb{N}_n, i \neq j, T_i \cap T_j = \emptyset, P_0\) is a set of process idle places, \(P_A\) is a set of activity places, and \(P_R\) is a set of resource places with \(P_0, P_A\), and \(P_R\) being mutually disjoint. Activity place set \(P_A\) has a partition with \(P_A = P_{A1} \cup P_{A2} \cup \ldots \cup P_{An}\), where \(\forall i,j \in \mathbb{N}_n, i \neq j, P_{Ai} \cap P_{Aj} = \emptyset\), implying that a system is composed of \(n\) concurrent processes.

For an M-net \((N, M_0)\), its liveness requirements are usually represented by a set of GMECs. Accordingly, a liveness-enforcing supervisor \((N^c, M^c_0)\), either maximally permissive or non-maximally permissive (i.e., suboptimal), consists of a set of control places that implement the set of GMECs, denoted by \(P_C = \{p_{c1}, p_{c2}, \ldots, p_{cn}\}\), each of which decides a place invariant, in fact, a minimal P-semiflow with respect to the activity places in \((N, M_0)\). In other words, any control place \(p_{ci} \in P_C\) and some activity places form the support of a place invariant, denoted by \(P_I\), as seen in the monitor-based liveness-enforcing Petri net supervisors for FMSs developed by Chen et al. (2011); Chen & Li (2011, 2012); Chen et al. (2012); Chen & Li (2013); Ghaffari et al. (2003); Guan et al. (2015); Ji & Wang (2014); Huang & Pan (2011); Li et al. (2007); Li & Zhou (2008b,c); Li et al. (2012); Liu et al. (2014); Pirrodi et al. (2008, 2009); Uzam (2002, 2004); Uzam et al. (2007); Uzam & Zhou (2006, 2007); Zouari & Barkaoui (2003). Let \(G(N, M_0)\) denote the reachability graph of an M-net \((N, M_0)\). \(G(N, M_0)\) can be divided into two parts, LZ and DZ, representing permissive (legal) and forbidden (illegal) markings of \((N, M_0)\), respectively.

As stated in Li et al. (2012), M-nets are more general than almost all manufacturing-oriented Petri net subclasses in the literature such as PPN, augmented marked graphs (Chu & Xie, 1997), S^3PR (Ezepeleta et al., 1995), L-S^3PR (Ezepeleta et al., 1998), S^1R (Abdallah & ElMaraghy, 1998), S^3PR (Tricas & Martinez, 1995), ES^3PR (Huang et al., 2001), WS^3PSR (Tricas et al., 2000), S^3PMR (Huang et al., 2006), RCN-merged nets (Wysk et al., 1994), ERCN-merged nets (Chu & Xie, 1997), ERCN*-merged nets (Jeng et al., 2004), S^3PGR^2 (Park & Reveliotis, 2001), G-tasks (Barkaoui et al., 1997), and well-formed G-systems (Zouari & Barkaoui, 2003). Note that, we assume that each monitor in a liveness-enforcing supervisor \((N^c, M^c_0)\) is necessary, implying that its removal would deprive the liveness of the supervisor \((N^c, M^c_0)\). This paper aims to combine the necessary monitors as well as their corresponding place invariants such that the structure of \((N^c, M^c_0)\) can be reduced.

3. First-met Bad Markings and Their Control

A monitor in an optimal (maximally permissive) supervisor necessarily prohibits the reachability of at least one first-met bad marking (FBM) that represents the first entry from the live zone to the dead zone in the reachability graph of a Petri net model, otherwise liveness will be lost.

In a supervisor, monitors make all FBM’s forbidden, while keeping all (if the supervisor is optimal) or partial (if the supervisor is suboptimal) legal markings reachable in the controlled system. As stated previously, the reachability graph of a considered net model contains two disjoint parts: the LZ and DZ. Accordingly, we use \(LZ\) and \(DZ\) to represent the sets of markings in the LZ and DZ, respectively. Unless otherwise stated, a net system considered in this work is hereinafter referred to as an M-net \((N, M_0)\) with \(LZ = \{M|M \in R(N, M_0), \exists \sigma \in T^*, M[\sigma]M_0\}\) and \(DZ = \{M|M \in R(N, M_0), \nexists \sigma \in T^*, M[\sigma]M_0\}\).
DEFINITION 4 Let \((N, M_0)\) be a net system with \(N = (P, T, F, W)\). A marking \(M' \in DZ\) is said to be an FBM (first-met bad marking) if \(\exists M' \in LZ, \exists t \in T, M(t)M'\). The set of FBMs is denoted by \(\mathcal{M}_f\).

DEFINITION 5 A marking \(M \in LZ\) is dangerous if \(\exists t \in T, \exists M' \in \mathcal{M}_f, M(t)M'\).

DEFINITION 6 Let \(M_f\) be an FBM and \(PI_{M_f}\) be a place invariant (semiflow) that forbids \(M_f\) in a supervisor. The activity support of \(M_f\) \((PI_{M_f})\) is defined as \(|M_f| = \{p \in P_A|M_f(p) > 0\}\) \(|PI_{M_f}| = \{p \in P_A|PI_{M_f}(p) > 0\}\).

COROLLARY 2 Let \(M_f\) be an FBM and \(PI_{M_f}\) be a place invariant that forbids \(M_f\) in a supervisor. Then \(|M_f| = |PI_{M_f}|\).

Proof. By contradiction, we assume that \(\exists p \in |M_f|, p \notin |PI_{M_f}|\). Then, the number of tokens in place \(p\) is not controlled by the place invariant \(PI_{M_f}\), implying that \(M_f\) cannot be prevented from being reached by \(PI_{M_f}\).

For the sake of brevity, only the markings in the activity places of an FBM are considered, which carries sufficient information to design a supervisor thanks to the marking invariant laws between resources and their activity places. Fig. 1(a) depicts an M-net and Fig. 1(b) shows its reachability graph, where \(M_1, M_8, M_9, M_{11},\) and \(M_{14}\) are FBMs, and \(M_1, M_2, M_3, M_5, M_6,\) and \(M_{11}\) are dangerous markings. Each dangerous marking has in the DZ at least one successor that is an FBM. Specifically, we have \(M_4 = p_2 + p_5, M_8 = p_3 + p_5, M_9 = p_2 + p_6, M_{13} = p_2 + p_3 + p_5,\) and \(M_{14} = p_2 + p_5 + p_6\). In this paper, we compute one place invariant (or a monitor) in a supervisor to forbid the reachability of multiple FBMs.

DEFINITION 7 Let \((N, M_0)\) be an M-net. Suppose that a liveness-enforcing Petri net supervisor contains \(k\) monitors. Let \(PL_1, PL_2, \ldots, PL_k\) be the place invariants associated with the \(k\) monitors to prevent the reachability of \(k\) FBMs, \(FBM_1, FBM_2, \ldots, FBM_k\), respectively. \(PL_1, PL_2, \ldots, PL_r\) \((r \leq k)\) are said to be equivalent with a place invariant \(PL_{[1,r]}\) if it can forbid \(FBM_1, FBM_2, \ldots, FBM_r\) and cannot forbid any legal marking in \(LZ\).

PROPOSITION 2 Let \(M_{f_1}\) and \(M_{f_2}\) be two FBMs. If \(|M_{f_1}| \cap |M_{f_2}| \neq \emptyset\), then \(\exists M_1, M_2 \in LZ, t_i \in T\), such that \(M_1(t_i)M_{f_1}\) and \(M_2(t_i)M_{f_2}\).

Proof. Let \(PL_1\) and \(PL_2\) be two place invariants used to forbid FBMs \(M_{f_1}\) and \(M_{f_2}\), respectively. \(|M_{f_1}| \cap |M_{f_2}| \neq \emptyset\) means \(|PL_1| \cap |PL_2| \neq \emptyset\). Let \(p \in |PL_1| \cap |PL_2|\). Then, \(p \in P_A\) holds, since \(PL_1\) and \(PL_2\) do not share monitors. Let \(t_i \in^* p\) be a transition in the preset of activity place \(p\). We conclude that the enforcement of the marking invariant constraints decided by both place invariants disables transition \(t_i\) at some markings in \(LZ\), since the firing of \(t_i\) increases the number of tokens in \(p\), from which a deadlock state may stem. As a result, \(\exists M_1, M_2 \in LZ, t_i \in T\), such that \(M_1(t_i)M_{f_1}\) and \(M_2(t_i)M_{f_2}\) hold.

For economy of space, a place invariant \(I\) can be written as \(I = \sum_{p \in P} I(p)p\). For example, \(I = (1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0)^T\) is a place invariant of the net in Fig. 1(a). It can be written as \(I = p_1 + p_2 + p_3\). Figs. 1(c) and 1(d) show two supervisors, respectively. In Fig. 1(c), \(p_{c_1}, p_{c_2},\) and \(p_{c_3}\) are three monitors and their corresponding place invariants are \(PI_{p_{c_1}} = p_{c_1} + p_2 + p_3, PI_{p_{c_2}} = p_{c_2} + p_2 + p_6,\) and
$PI_{P_3} = p_3 + p_3 + p_5$. By considering the initial markings of the three monitors, these place invariants can be equivalently and unambiguously derived from the GMECs $PI_{P_1} \equiv p_2 + p_5 \leq 1$, $PI_{P_2} \equiv p_2 + p_6 \leq 1$, and $PI_{P_3} \equiv p_3 + p_5 \leq 1$, respectively. For brevity, the activity supports of the place invariants are $|PI_{P_1}| = \{p_2, p_5\}$, $|PI_{P_2}| = \{p_2, p_6\}$, and $|PI_{P_3}| = \{p_3, p_5\}$, respectively.

Proposition 2 indicates that if an activity place $p \in P_A$ is marked at two FBMs $M_{f_1}$ and $M_{f_2}$, then $p$ has only one input transition $t$ (by the definition of an $S^3$PR). The firing of $t$ will mark $p$. Specifically, if two FBMs or their corresponding place invariants share an activity place, then both place invariants are used to control the firing of the input transitions of the share activity place, since, in an FMS modeled with M-nets, deadlocks are caused by the fact that activity places overly hold the tokens initially marked in the

Fig. 1. (a) A Petri net model of an FMS, (b) its reachability graph, (c) a controlled system with three monitors, and (d) a controlled system with two monitors.
resource places. That is to say, deadlock prevention is achieved by prohibiting too many tokens to be held by activity places. As a result, we need to control the firing of the transitions in the preset of the shared activity place. In other words, it is the transitions in the preset of the shared activity places whose firing leads a system from the LZ to DZ. In Fig. 1(c), \( P_{T_1} \equiv p_2 + p_5 \leq 1 \) is used to prohibit the reachability of marking \( M_4 \) and \( P_{T_2} \equiv p_2 + p_6 \leq 1 \) is used to prohibit marking \( M_9 \). Due to \( |P_{T_1}| \cap |P_{T_2}| = \{p_2\} \) and \( t_1 \in p_2 \), we conclude that there exist markings \( M_2, M_5 \in L(z) \), such that \( M_2[t_1]M_4 \) and \( M_5[t_1]M_9 \), as seen in Fig. 1(b). Furthermore, in Fig. 1(d), four FBMs \( M_4, M_9, M_{13}, \) and \( M_{14} \) are controlled by one GMEC (place invariant) \( P_{T_2} \equiv 2\mu_2 + \mu_5 + \mu_6 \leq 2 \).

The above result motivates us to explore the possibility to control multiple FBMs by one place invariant if their supports share activity places in a Petri net model. It is natural to infer that the activity support of the computed place invariant is the union of those of the FBMs. As stated previously, token distribution only in activity places is considered for deadlock control purposes (Chen et al., 2011; Chen & Li, 2013). In what follows, let \( M_f = \sum_{p_i \in P_A} M(p_i) p_i \) denote a particular FBM and \( \mu_i \) be any reachable marking of activity place \( p_i \). For example, a marking \( M_f = p_3 + p_{11} \) is an FBM, implying that both activity places \( p_3 \) and \( p_{11} \) contain one token at marking \( M_f \), respectively. To prevent the reachability of such an FBM, a GMEC \( \mu_3 + \mu_{11} \leq 1 \) has to be enforced, where \( \mu_3 (\mu_{11}) \) is the number of tokens in activity place \( p_3 (p_{11}) \) at any reachable marking of a net model. Such a GMEC or its related place invariant ensures that at any reachable marking, the number of tokens in \( p_3 \) and \( p_{11} \) is not greater than one.

In summary, for an FBM \( M_f = \sum_{p_i \in P_A} M(p_i) p_i \), there are \( \sum_{p_i \in P_A} M(p_i) \) tokens in the activity places. Thus, the enforcement of a GMEC \( \sum_{p_i \in P_A} \mu_i = \sum_{p_i \in P_A} M(p_i) - 1 \) can prevent the reachability of the FBM \( M_f = \sum_{p_i \in P_A} M(p_i) p_i \). For example, \( p_2 + 2p_3 \) is an FBM and its corresponding GMEC is \( \mu_2 + \mu_3 \leq 1 - 2 - 1 = 2 \).

Such a GMEC can be implemented through a monitor place (or simply called a monitor) by the marking-invariant law of Petri nets (Giu et al., 1992; Barkaoui et al., 1997) to form a place invariant (in general, a place semiflow) associated with activity places, as seen in a liveness-enforcing supervisor for FMSs. In this sense, a constraint, i.e., GMEC, and a place invariant can be thought of as being equivalent. Generally, a place invariant in a supervisor implementing a GMEC takes a form of \( \alpha_1\mu_1 + \alpha_2\mu_2 + \ldots + \alpha_n\mu_n \leq k \), where \( \alpha_i \), a non-negative integer, is called the control coefficient of place \( p_i \), \( k \in \{1, 2, \ldots\} \) is called the weighted constant token, \( \mu_i \) is a marking of an activity place \( p_i \), and \( n \) is the number of activity places, i.e., \( n = |P_A| \). In what follows, we do not differentiate a place invariant from its corresponding GMEC in case of no confusion.

Suppose that \( p_2 + 2p_9, p_3 + 2p_9, \) and \( p_4 + 2p_9 \) are three FBMs in a Petri net model. Then, their activity supports are \( \{p_2, p_9\}, \{p_3, p_9\}, \) and \( \{p_4, p_9\} \), respectively. Since they share activity place \( p_9 \), it is possible that one place invariant with its activity support \( \{p_2, p_3, p_4, p_9\} \) can forbid the reachability of the three FBMs according to Proposition 2. Denoted by \( |mP| \), it is called a support of monolithic place invariant (GMEC). The monolithic place invariant (GMEC) takes the form \( \alpha_2\mu_2 + \alpha_3\mu_3 + \alpha_4\mu_4 + \alpha_9\mu_9 \leq k \). Thus, we need to decide the values of \( \alpha_2, \alpha_3, \alpha_4, \alpha_9, \) and \( k \).

For each FBM \( M_f = \sum_{p_i \in P_A} M(p_i) p_i \), let \( a_i = M(p_i) \). In order to compute the control coefficients \( \alpha_1, \alpha_2, \ldots, \alpha_n \) of the place invariant to prevent FBMs, we generate an equation for each FBM \( M_f = \sum_{p_i \in P_A} a_i p_i \):

\[
a_1\alpha_1 + a_2\alpha_2 + \ldots + a_n\alpha_n = k + 1
\]

For the three FBMs \( p_2 + 2p_9, p_3 + 2p_9, \) and \( p_4 + 2p_9 \) in Example 1, the equations that we generate are \( \alpha_2 + 2\alpha_9 = k + 1, \alpha_3 + 2\alpha_9 = k + 1, \) and \( \alpha_4 + 2\alpha_9 = k + 1 \), from which we conclude \( \alpha_2 = \alpha_3 = \alpha_4 \).
ON STRUCTURAL REDUCTION OF LIVENESS-ENFORCING PETRI NET SUPERVISORS

DEFINITION 8 Let $M_1, M_2, \ldots, M_d$ be FBMs to be forbidden. The set of shared activity places is $eta = |M_1| \cap |M_2| \cap \ldots \cap |M_d|$. Let $\beta' = (|M_1| \cup |M_2| \cup \ldots \cup |M_d|) \setminus \beta$ denote the activity places unshared by the place invariants that forbid $M_1, M_2, \ldots, M_d$.

COROLLARY 3 The activity support of a monolithic place invariant to forbid $M_1, M_2, \ldots, M_d$ is $\beta \cup \beta'$, i.e., $|M_1| \cup |M_2| \cup \ldots \cup |M_d|$.

DEFINITION 9 Let $\beta$ be the set of shared places by $M_1, M_2, \ldots, M_d$. $\forall p_i \in \beta$, the control coefficient of place $p_i$, denoted by $\alpha_i$, in the place invariant that forbids $d$ FBMs satisfies

$$\alpha_i \geq d$$

(3)

For the sake of easy computation, we have

$$\alpha_i = d$$

(4)

The control coefficients of the places in $\beta'$ are positive integer, i.e.,

$$\forall p_j \in \beta', \alpha_j \in \mathbb{N}^+$$

(5)

Let $\{M_i|i = 1, 2, \ldots, d\}$ be a set of FBMs in a Petri net with activity place set $P_A$ and $PI$ be a place invariant to forbid the $d$ FBMs. The control coefficients of activity places $\alpha_i$’s ($i \in \{1, 2, \ldots, |P_A|\}$) and the weighted token constant $k$ of $PI \equiv \sum_{i=1}^{|P_A|} \alpha_i \mu_i \leq k$ are the solution of the following linear programming problem:

$$\min \sum_{i=1}^{|P_A|} \alpha_i$$

subject to Eqs. (2), (4), and (5).

Assume that two FBMs in some Petri net model are $p_2 + p_3 + p_4$ and $p_2 + p_4 + p_{12}$. Then, we have $\beta = \{p_2, p_4\}$. According to Definition 9, the control coefficients of $p_2$ and $p_4$ in the computed place invariant are $\alpha_2 = \alpha_4 = 2$ due to $d = 2$, i.e., there are two FBMs to be considered. We have

$$\alpha_2 + \alpha_3 + \alpha_4 = k + 1$$

$$\alpha_2 + \alpha_4 + \alpha_{12} = k + 1$$

$$\alpha_2 = \alpha_4 = 2$$

A feasible solution to the above equations is $\alpha_3 = \alpha_{12} = 1$, $\alpha_2 = \alpha_4 = 2$, and $k = 4$. The computed monolithic GMEC (place invariant) that can possibly prevent the two FBMs is $2\mu_2 + \mu_3 + 2\mu_4 + \mu_{12} \leq 4$, implying that, at any reachable marking, if the weighted token sum in $p_2, p_3, p_4,$ and $p_{12}$ is not greater than 4, the two FBMs can be forbidden. Next we propose the conditions under which multiple FBMs can be forbidden by computing a monolithic GMEC (place invariant) in an optimal liveness-enforcing supervisor.
4. Identification of FBMs

This section presents a method to identify a set of FBMs in a Petri net model that can be prohibited by a monolithic GMEC (place invariant). This paper develops three algorithms to identify FBMs to be forbidden. Let \( X \) and \( Y \) denote two concurrent processes in a system modeled with an M-net. We use \( P_{AX} \) and \( P_{AY} \) to denote the sets of activity places in processes \( X \) and \( Y \), respectively.

**Definition 10** Given \( d \) FBMs \( M_{f1}, M_{f2}, \ldots, M_{fd} \) of an M-net \((N,M_0)\), they are said to be \( x \)-controlled by a monolithic place invariant if the following statements are satisfied:

(i) \( \beta = |M_{f1}| \cap |M_{f2}| \cap \ldots \cap |M_{fd}| \neq \emptyset \);

(ii) \( \exists i \in \mathbb{N}_d, \exists P_{AX}, P_{AY} \subseteq P_A, ||M_{fi}|| \subseteq P_{AX}, \text{i.e., } \forall p \in ||M_{fi}||, p \notin P_{AY} \);

(iii) \( \exists r \in P_R, \{r\} \cup \beta' \) is the support of a place invariant of \((N,M_0)\).

Let us consider an FMS as shown in Fig. 2(a). It consists of two robots R1-R2, each of which can hold one part at a time, four machines M1-M4, each of which can process one part at a time, two loading buffers I1-I2, and two unloading buffers O1-O2. Two part types are considered in the system: P1 and P2. The production sequences are shown in Fig. 2(b). The system has been studied in several papers (Ezepeleta et al., 1995; Chen et al., 2011, 2012; Uzam et al., 2007; Uzam & Zhou, 2007; Pirrodi et al., 2009).

The Petri net model of this FMS is shown in Fig. 3. It has 282 reachable states, 205 and 77 of which are legal and illegal, respectively. The net has 19 places and 14 transitions. Places can be thought of as the collection of \( P_A = \{p_2, \ldots, p_7, p_9, \ldots, p_{13}\} \), \( P^0 = \{p_1, p_8\} \), and \( P_R = \{p_{14}, \ldots, p_{19}\} \). Monitors (as well as their corresponding place invariants and FBMs) computed for this Petri net model shown in Fig. 3 are provided in Table 1 and the corresponding supervisor is maximally permissive with 205 reachable states.

![Fig. 2. (a) An FMS layout and (b) its production sequences.](https://academic.oup.com/imamci/article-abstract/35/4/1217/3792087)
Let us consider two FBMs $M_{f_1} = p_3 + p_{11}$ and $M_{f_2} = p_{11} + p_{12}$ in Table 1, where $P_{A1} = \{p_2, p_3, p_4, p_5, p_6, p_7\}$ and $P_{A2} = \{p_9, p_{10}, p_{11}, p_{12}, p_{13}\}$. Their supports are $\{p_3, p_{12}\}$ and $\{p_{11}, p_{12}\}$, respectively. Then, $\beta = \{p_{11}\} \neq \emptyset$ and Condition i) in Definition 10 is satisfied. Note that $\{p_{11}, p_{12}\}$ is included in $P_{A2}$. Condition ii) holds. Since $\beta' = \{p_3, p_{12}\}$, $p_{15}$ is a resource place, and $p_3 + p_{12} + p_{15}$ is a place

![Diagram](https://example.com/diagram.png)

Fig. 3. An S³PR Petri net model for the FMS in Fig. 2.

### Table 1 FBMs and Control Places Computed for the Petri Net Shown in Fig. 3 from Chen et al. (2011); Chen & Li (2013)

<table>
<thead>
<tr>
<th>FBM</th>
<th>$P_{i}$</th>
<th>$C$</th>
<th>$\cdot C$</th>
<th>$C^*$</th>
<th>$M_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_3 + p_{11}$</td>
<td>$\mu_3 + \mu_{41} \leq 1$</td>
<td>$C_1$</td>
<td>$t_5, t_{12}$</td>
<td>$t_2, t_{11}$</td>
<td>1</td>
</tr>
<tr>
<td>$p_{11} + p_{12}$</td>
<td>$\mu_{11} + \mu_{12} \leq 1$</td>
<td>$C_2$</td>
<td>$t_{13}$</td>
<td>$t_{11}$</td>
<td>1</td>
</tr>
<tr>
<td>$p_2 + p_3 + p_4$</td>
<td>$\mu_2 + \mu_3 + \mu_4 \leq 2$</td>
<td>$C_3$</td>
<td>$t_4, t_5$</td>
<td>$t_{1}$</td>
<td>2</td>
</tr>
<tr>
<td>$p_2 + p_4 + p_{12}$</td>
<td>$\mu_2 + \mu_4 + \mu_{12} \leq 2$</td>
<td>$C_4$</td>
<td>$t_2, t_4, t_{13}$</td>
<td>$t_1, t_{12}$</td>
<td>2</td>
</tr>
<tr>
<td>$p_5 + p_6 + p_9 + p_{10}$</td>
<td>$\mu_5 + \mu_6 + \mu_9 + \mu_{10} \leq 3$</td>
<td>$C_5$</td>
<td>$t_7, t_{11}$</td>
<td>$t_4, t_5, t_9$</td>
<td>3</td>
</tr>
<tr>
<td>$p_3 + p_6 + p_9 + p_{10}$</td>
<td>$\mu_3 + \mu_6 + \mu_9 + \mu_{10} \leq 3$</td>
<td>$C_6$</td>
<td>$t_5, t_7, t_{11}$</td>
<td>$t_2, t_6, t_9$</td>
<td>3</td>
</tr>
<tr>
<td>$p_3 + p_5 + p_9 + p_{10}$</td>
<td>$\mu_3 + \mu_5 + \mu_9 + \mu_{10} \leq 3$</td>
<td>$C_7$</td>
<td>$t_6, t_{11}$</td>
<td>$t_2, t_4, t_9$</td>
<td>3</td>
</tr>
<tr>
<td>$p_2 + p_4 + p_6 + p_9 + p_{10}$</td>
<td>$\mu_2 + \mu_4 + \mu_6 + \mu_9 + \mu_{10} \leq 4$</td>
<td>$C_8$</td>
<td>$t_2, t_4, t_7, t_{11}$</td>
<td>$t_1, t_6, t_9$</td>
<td>4</td>
</tr>
</tbody>
</table>
invariant of \((N, M_0)\), we conclude that Condition iii) is true. The two FBMs are \(x\)-controlled by a monolithic GMEC (place invariant) that is the solution of the following system:

\[
\begin{align*}
\alpha_3 + \alpha_{11} &= k + 1, \\
\alpha_{11} + \alpha_{12} &= k + 1, \\
\alpha_{11} &= 2.
\end{align*}
\]

We have \(\alpha_3 = \alpha_{12} = 1\), \(\alpha_{11} = 2\), and \(k = 2\), i.e., the computed place invariant or GMEC is \(\mu_3 + 2\mu_{11} + \mu_{12} \leq 2\). It can be verified that \(\mu_3 + 2\mu_{11} + \mu_{12} \leq 2\) is equivalent to \(\mu_3 + \mu_{11} \leq 1\) and \(\mu_{11} + \mu_{12} \leq 1\) with respect to the Petri net model in Fig. 3. We can use \(\mu_3 + 2\mu_{11} + \mu_{12} \leq 2\) to replace \(\mu_3 + \mu_{11} \leq 1\) and \(\mu_{11} + \mu_{12} \leq 1\).

For FBMs \(M_{f_3} = p_2 + p_3 + p_4\) and \(M_{f_4} = p_2 + p_4 + p_{12}\) in Table 1, their activity supports are \(|M_{f_3}| = \{p_2, p_3, p_4\}\) and \(|M_{f_4}| = \{p_2, p_4, p_{12}\}\). By Definition 10, \(\beta = \{p_2, p_4\}\) is obtained. Since \(\beta\) is not empty, Condition i) is satisfied. With \(|M_{f_3}| = \{p_2, p_3, p_4\}\) and \(P_{A_1} = \{p_2, p_3, p_4, p_5, p_6, p_7\}\), Condition ii) is satisfied. By noting that \(\beta' = \{p_3, p_{12}\}\), \(p_{15}\) is a resource place, and \(p_1 + p_{12} + p_{15}\) is a place invariant of \((N, M_0)\), we conclude that Condition iii) is satisfied. Then, \(M_{f_3}\) and \(M_{f_4}\) are FBMs that are \(x\)-controlled by a monolithic GMEC that is a solution of the following equations:

\[
\begin{align*}
\alpha_2 + \alpha_3 + \alpha_4 &= k + 1, \\
\alpha_2 + \alpha_4 + \alpha_{12} &= k + 1, \\
\alpha_2 &= \alpha_4 = 2.
\end{align*}
\]

We have \(\alpha_3 = \alpha_{12} = 1\), \(\alpha_2 = \alpha_4 = 2\), and \(k = 4\), i.e., the computed monolithic GMEC is \(2\mu_2 + \mu_3 + 2\mu_4 + \mu_{12} \leq 4\). It can be verified that \(2\mu_2 + \mu_3 + 2\mu_4 + \mu_{12} \leq 4\) is equivalent to \(\mu_2 + \mu_3 + \mu_4 \leq 2\) and \(\mu_2 + \mu_3 + \mu_{12} \leq 2\) with respect to the Petri net model in Fig. 3.

**Theorem 1** Let \(mP_I\) be a monolithic GMEC (place invariant) obtained by Eq. (6) due to \(d\) FBMs \(M_{f_1}, M_{f_2}, \ldots, M_{f_d}\) satisfying Definition 10 and \(P_{I_{f_1}}, P_{I_{f_2}}, \ldots, P_{I_{f_d}}\) be the GMECs (place invariants) to control the \(d\) FBMs, respectively, in an optimal supervisor of an M-net \((N, M_0)\). \(mP_I\) prohibits the \(d\) FBMs if it is equivalent to \(P_{I_{f_1}}, P_{I_{f_2}}, \ldots, P_{I_{f_d}}\) with respect to \((N, M_0)\).

**Proof.** The \(d\) FBMs are controlled by \(P_{I_{f_1}}, P_{I_{f_2}}, \ldots, P_{I_{f_d}}\) in an optimal supervisor, implying that no legal marking is excluded due to their enforcement. If the monolithic GMECs defined by Eq. (6) is equivalent to them, then its enforcement forbids the \(d\) FBMs and no legal marking is excluded from the controlled system. \(\square\)

Theorem 1 indicates that if a monolithic GMEC is equivalent to a number of place invariants, then the GMEC can replace these place invariants in the original supervisor, i.e., many constraints or monitors are removed from the supervisor and the resulting supervisor is also maximally permissive, which leads to a structurally simple supervisor. Next we propose an algorithm to identify a set of FBMs that can potentially be \(x\)-controlled by one GMEC. We denote the set of FBMs in an optimal supervisor by \(\Theta\).

**Algorithm 1** Identification of FBMs from \(\Theta\)

**Input:** \(\Theta = \{M_{f_1}, M_{f_2}, \ldots, M_{f_m}\}\)

**Output:** \(\Theta_1, \Theta_2, \ldots, \Theta_l\), where \(\forall i \in N_I, \Theta_i\) satisfies Definition 10

\(i := 1; l := 0; \Theta^* := \Theta;\)
while \((i \leq m)\) do
\[
\Theta_i := \{M_i\};
\]
\[j := 2;\]
while \((j \leq m)\) do
\[
\Theta_i := \Theta_i \cup \{M_j\};
\]
while \((\Theta_i \text{ satisfies Definition 10})\) do
\[
j := j + 1;
\]
\[
\Theta_i := \Theta_i \cup \{M_j\};
\]
end while
\[
\Theta_i := \Theta_i \backslash \{M_j\};
\]
\[j := j + 1;\]
end while
output \(\Theta_i;\)
\[\Theta := \Theta \backslash \Theta_i;\]
\[l := l + 1;\]
\[i := i + 1;\]

Re-order the elements in \(\Theta\) by an increasing order from 1 to \(|\Theta|\);
\[m := |\Theta|;\]
end while
\[\Theta_x := \Theta \cup \Theta_2 \cup \ldots \cup \Theta_i;\]
\[\Theta_y := \Theta \cup \Theta_i;\]
\[\Theta_y := \Theta \cup \Theta_i;\]

End of the algorithm.

**Definition 11** Given \(d\) FBMs \(M_1, M_2, \ldots, M_d\) of an M-net \((N, M_0)\), they are said to be \(y\)-controlled by a monolithic GMEC (place invariant) if the following statements are satisfied:

i) \(\beta = |M_1| \cap |M_2| \cap \ldots \cap |M_d| \neq \emptyset;\)

ii) \(\exists P_{AX}, P_{AY} \subseteq P_A\) s.t. \(\beta \subseteq P_{AY}\), i.e., \(\beta \cap P_{AX} = \emptyset\); and

iii) \(\beta' \subseteq P_{AX}.\)

The Petri net model of an FMS depicted in Fig. 4 is an M-net. It has 250 reachable states, 194 and 56 of which are legal and illegal, respectively. The net has 16 places and 12 transitions. The places can be considered as the collection of \(P_A = \{p_2, p_3, \ldots, p_7, p_9, p_{10}, p_{11}\} \neq \emptyset\) and \(P_h = \{p_{12}, p_{13}, \ldots, p_{16}\} \neq \emptyset\). A maximally permissive liveness-enforcing supervisor consists of four control places as shown in Table 2 (Ghaffari et al., 2003). The controlled net system has 194 reachable states by the supervisor.

Let us consider three FBMs \(M_1 = 2p_4 + p_5, M_2 = p_5 + p_9, \text{ and } M_3 = p_6 + p_9\) in Table 2. Their activity supports are \(|M_1| = \{p_4, p_9\}\), \(|M_2| = \{p_5, p_9\}\), and \(|M_3| = \{p_6, p_9\}\). We have \(\beta = \{p_9\}\). Since \(\beta\) is not empty, Condition i) in Definition 11 is satisfied. From Fig. 4, \(P_{A1} = \{p_2, p_3, p_4, p_5, p_6, p_7\}\) and \(P_{A2} = \{p_9, p_{10}, p_{11}\}\) are the sets of activity places in two concurrent processes. We have \(\beta \subseteq P_{A2}\) and \(\beta \cap P_{A1} = \emptyset\). Hence Condition ii) is satisfied. By \(\beta' = \{p_4, p_5, p_6\}\), \(\beta' \subseteq P_{A1}\) is true. Thus, Condition iii) in Definition 11 holds.

We conclude that the three FBMs are \(y\)-controlled by a monolithic GMEC. The control coefficients of unshared places in \(\beta'\) and the weighted token constant \(k\) in the GMEC are the solution of \(2\alpha_4 + \alpha_9 = k + 1, \alpha_5 + \alpha_9 = k + 1, \alpha_6 + \alpha_9 = k + 1,\) and \(\alpha_9 = 3\). We have \(\alpha_4 = 1, \alpha_5 = \alpha_6 = 2, \alpha_9 = 3,\) and \(k = 4.\)
Accordingly, the resulting monolithic GMEC is $p_4 + 2p_5 + 2p_6 + 3p_9 \leq 4$. It can be verified that $p_4 + 2p_5 + 2p_6 + 3p_9 \leq 4$ is equivalent to $2\mu_4 + \mu_9 \leq 2$, $\mu_5 + \mu_9 \leq 1$, and $\mu_6 + \mu_9 \leq 1$ with respect to the Petri net model shown in Fig. 4.

Next, we formulate an algorithm to find the FBM$s satisfying Definition 11. Then, the supervisor with the newly generated GMECs is shown in Table 3. The performance of different supervisors is shown in Table 4.

**Theorem 2** Let $mPI$ be a monolithic GMEC (place invariant) obtained by Eq. (6) due to $d$ FBM$s M_{f_1}, M_{f_2}, \ldots, M_{f_d}$ satisfying Definition 11 and $PI_{f_1}, PI_{f_2}, \ldots, PI_{f_d}$ be the GMEC$s to control the $d$ FBM$s, respectively, in an optimal supervisor of an M-net $(N, M_0)$. $mPI$ prohibits the $d$ FBM$s if it is equivalent to $PI_{f_1}, PI_{f_2}, \ldots, PI_{f_d}$ with respect to $(N, M_0)$. 

---

**Table 2** Place invariants and monitors computed for the Petri net model shown in Fig. 4

<table>
<thead>
<tr>
<th>FBM</th>
<th>$PI_i$</th>
<th>$C_i$</th>
<th>$\mathbf{C_i}$</th>
<th>$\mathbf{C^*}$</th>
<th>$M_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2p_4 + p_9$</td>
<td>$\mu_4 + \mu_9 \leq 2$</td>
<td>$C_1$</td>
<td>$t_7, t_11$</td>
<td>$t_6, t_12$</td>
<td>2</td>
</tr>
<tr>
<td>$p_5 + p_9$</td>
<td>$\mu_5 + \mu_9 \leq 1$</td>
<td>$C_2$</td>
<td>$t_4, t_11$</td>
<td>$t_3, t_{12}$</td>
<td>1</td>
</tr>
<tr>
<td>$p_6 + p_9$</td>
<td>$\mu_6 + \mu_9 \leq 1$</td>
<td>$C_3$</td>
<td>$t_8, t_{11}$</td>
<td>$t_7, t_{12}$</td>
<td>1</td>
</tr>
<tr>
<td>$2p_4 + p_{10}$</td>
<td>$\mu_4 + \mu_{10} \leq 2$</td>
<td>$C_4$</td>
<td>$t_7, t_{10}$</td>
<td>$t_{6,11}$</td>
<td>2</td>
</tr>
</tbody>
</table>

Fig. 4. $S^3$PR model of an FMS from (Chen et al., 2011).
Table 3 Monolithic place invariants for the net model shown in Fig. 4

<table>
<thead>
<tr>
<th>$i$</th>
<th>$PI_i$</th>
<th>$C_i$</th>
<th>$C_i^*$</th>
<th>$M_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$mPI_1 = \mu_4 + 2\mu_5 + 2\mu_6 + 3\mu_9 \leq 4$</td>
<td>$C_1$</td>
<td>$2t_4, 2t_8, 3t_{11}$</td>
<td>$2t_3, t_6, t_7, 3t_{12}$</td>
</tr>
<tr>
<td>2</td>
<td>$PI_4 = \mu_4 + \mu_{10} \leq 2$</td>
<td>$C_2$</td>
<td>$t_7, t_{10}$</td>
<td>$t_6, t_{11}$</td>
</tr>
</tbody>
</table>

Table 4 Performance comparison of the Petri net model shown in Fig. 4

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of monitors</td>
<td>3</td>
<td>6</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Sum of arc weights</td>
<td>14</td>
<td>34</td>
<td>16</td>
<td>19</td>
<td>16</td>
<td>14</td>
</tr>
<tr>
<td>No. of tokens</td>
<td>7</td>
<td>18</td>
<td>6</td>
<td>8</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>No. of states</td>
<td>49</td>
<td>139</td>
<td>194</td>
<td>156</td>
<td>194</td>
<td>194</td>
</tr>
</tbody>
</table>

Proof. Similar to the proof of Theorem 1. □

Next we propose an algorithm to identify the FBMs that satisfy Definition 11. The algorithm is similar to Algorithm 1.

Algorithm 2 Identification of FBMs from $\Theta_{yzu}$

Input: $\Theta_{yzu} = \{M_1, M_2, \ldots, M_m\}$

Output: $\Theta_1, \Theta_2, \ldots, \Theta_q$, where $\forall i \in \mathbb{N}_q$, $\Theta_i$ satisfies Definition 11

1. $i := 1; l := 0; \Theta^* := \Theta_{yzu}$;
2. while ($i \leq m$) do
3.   $\Theta_i := \{M_i\}$;
4.   $j := 2$;
5.   while ($j \leq m$) do
6.     $\Theta_i := \Theta_i \cup \{M_j\}$;
7.     while ($\Theta_i$ satisfies Definition 11) do
8.       $j := j + 1$;
9.       $\Theta_i := \Theta_i \cup \{M_j\}$;
10.  end while
11.  $\Theta_i := \Theta_i \setminus \{M_j\}$;
12.  $j := j + 1$;
13. end while
14. output $\Theta_i$;
15. $\Theta_{yzu} := \Theta_{yzu} \setminus \Theta_i$;
16. $l := l + 1$;
17. $i := i + 1$;
18. Re-order the elements in $\Theta_{yzu}$ by an increasing order from 1 to $|\Theta_{yzu}|$
19. $m := |\Theta_{yzu}|$;
end while
\[ \Theta_y = \Theta_1 \cup \Theta_2 \cup \ldots \cup \Theta_l; \]
\[ \Theta_{zu} := \Theta^* \setminus \Theta_y; \]
/* the set of unidentified place invariants is denoted by \( \Theta_{zu} \) */
End of the algorithm.

**Definition 12** Given \( d \) FBMs \( M_1, M_2, \ldots, M_d \) of an M-net \((N, M_0)\), they are said to be \( z \)-controlled by a monolithic GMEC (place invariant) if the following statements are satisfied:

i) \( \beta = \{M_{11}\} \cap \{M_{22}\} \cap \ldots \cap \{M_{dd}\} \neq \emptyset; \)

ii) \( \#P_{Ax} \subseteq P_{\beta}, \beta \subseteq P_{Ax}; \)

iii) \( \beta' \) is a proper subnet of the support of a place invariant in \((N, M_0)\).

**Theorem 3** Let \( mPI \) be a monolithic GMEC (place invariant) obtained by Eq. (6) due to \( d \) FBMs \( M_1, M_2, \ldots, M_d \) satisfying Definition 12 and \( Pl_1, Pl_2, \ldots, Pl_d \) be the GMECs (place invariants) to control the \( d \) FBMs, respectively, in an optimal supervisor of an M-net \((N, M_0)\). \( mPI \) prohibits the \( d \) FBMs if it is equivalent to \( Pl_1, Pl_2, \ldots, Pl_d \) with respect to \((N, M_0)\).

**Proof.** Similar to the proof of Theorem 1. \( \square \)

The supervisor simplification recipe underlying Theorems 2 and 3 is similar to Theorem 1. Let us consider two FBMs \( M_{11} = p_2 + p_3 + p_9 + 2p_{11} + p_{15} + p_{16} \) and \( M_{14} = p_3 + p_9 + 2p_{11} + p_{15} + p_{16} \) in the Petri net model shown in Fig. 5. We have \( P_{A1} = \{p_2, p_3, p_4\}, P_{A2} = \{p_6, p_7, p_8, p_9, p_{10}, p_{11}, p_{12}, p_{13}\} \) and \( P_{A3} = \{p_{15}, p_{16}, p_{17}, p_{18}, p_{19}\} \) that are the sets of the activity places of three concurrent processes, respectively.

The activity supports of the two FBMs are \( \{M_{13}\} = \{p_2, p_3, p_9, p_{11}, p_{15}, p_{16}\} \) and \( \{M_{14}\} = \{p_3, p_9, p_{11}, p_{15}, p_{16}\} \). We have \( \beta = \{p_3, p_9, p_{11}, p_{15}, p_{16}\} \). We conclude that Condition i) in Definition 12 is satisfied. Since \( \forall i \in \{1, 2, 3\}, \beta \not\subseteq P_{Ai}; \) Condition ii) is satisfied. Note that \( \beta' = \{p_2, p_9\} \) and \( I = p_2 + p_4 + p_8 + p_{12} + p_{17} + p_{21} \) is a place invariant of the Petri net model in Fig. 5, where \( p_{21} \) is a resource place. Obviously, \( \beta' \) is a proper subnet of the support of the place invariant \( I \). FBMs \( M_{13} \) and \( M_{14} \) are \( z \)-controlled by a monolithic GMEC that is the solution of the following linear programming problem:

\[
\min \{x_2 + x_3 + x_8 + x_9 + x_{11} + x_{15} + x_{16}\} \\
x_2 + x_3 + x_9 + 2x_{11} + x_{15} + x_{16} = k + 1 \\
x_3 + x_8 + x_9 + 2x_{11} + x_{15} + x_{16} = k + 1 \\
x_3 = x_9 = x_{11} = x_{15} = x_{16} = 2
\]

We have \( x_2 = x_8 = 1, x_3 = x_9 = x_{11} = x_{15} = x_{16} = 2, \) and \( k = 12 \). Thus, the resulting monolithic GMEC is \( \mu_2 + 2\mu_3 + \mu_8 + 2\mu_9 + 2\mu_{11} + 2\mu_{15} + 2\mu_{16} \leq 12 \). Accordingly, we formulate an algorithm to find the set of FBMs satisfying Definition 12.

**Algorithm 3** Identification of FBMs from \( \Theta_{zu} \)

**Input:** \( \Theta_{zu} = \{Pl_1, Pl_2, \ldots, Pl_m\} \)

**Output:** \( \Theta_1, \Theta_2, \ldots, \Theta_e, \forall i \in \{1, 2, \ldots, e\}, \Theta_i \) satisfies Definition 12

\( i := 1; l := 0; \Theta^* := \Theta_{zu}; \)
while ($i \leq m$) do
  $\Theta_i := \{M_f\}$;
  $j := 2$;
  while ($j \leq m$) do
    $\Theta_i := \Theta_i \cup \{M_f\}$;
    while ($\Theta_i$ satisfies Definition 12) do
      $j := j + 1$;
      $\Theta_i := \Theta_i \cup \{M_f\}$;
    end while
    $\Theta_i := \Theta_i \setminus \{M_f\}$;
    $j := j + 1$;
  end while
  output $\Theta_i$;
  $l := l + 1$;
  $i := i + 1$;
  $\Theta_{zu} := \Theta_{zu} \setminus \Theta_i$;
end while

Re-order the elements in $\Theta_{zu}$ by an increasing order from 1 to $|\Theta_{zu}|$;

$m := |\Theta_{zu}|$;
end while

$\Theta_z = \Theta_1 \cup \Theta_2 \cup \ldots \cup \Theta_i$;
$\Theta_a := \Theta^* \setminus \{\Theta_1, \Theta_2, \ldots, \Theta_i\}$; ^ the set of unidentified place invariants is denoted by $\Theta_a$ ^

End of the algorithm.
Table 5  Monolithic GMECs (place invariants) and monitors for the Petri net model in Fig. 3

<table>
<thead>
<tr>
<th>i</th>
<th>PI_i</th>
<th>C_i</th>
<th>C_i*</th>
<th>M_0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>mPI_1 = μ_3 + 2μ_11 + μ_12 ≤ 2</td>
<td>C_1</td>
<td>t_5, t_12, t_13</td>
<td>t_2, 2t_11</td>
</tr>
<tr>
<td>2</td>
<td>mPI_2 = 2μ_2 + μ_3 + 2μ_4 + μ_12 ≤ 4</td>
<td>C_2</td>
<td>t_2, 2t_4, t_5, t_13</td>
<td>2t_1, t_12</td>
</tr>
<tr>
<td>3</td>
<td>mPI_3 = μ_2 + 2μ_3 + μ_4 + 2μ_6 + 2μ_9 + 4μ_10 ≤ 11</td>
<td>C_3</td>
<td>2t_7, 4t_11</td>
<td>t_1, t_2, t_4, 4t_9</td>
</tr>
</tbody>
</table>

5. Experimental Examples

This section presents some FMS examples available in the literature to show the applicability of the proposed method. Let us consider the Petri net model in Fig. 3. The FBMs and their corresponding place invariants are shown in Table 1. We have M_f_1 = p_3 + p_11, M_f_2 = p_11 + p_12, M_f_3 = p_2 + p_3 + p_4, M_f_4 = p_2 + p_4 + p_12, M_f_5 = p_5 + p_6 + p_9 + p_10, M_f_6 = p_3 + p_6 + p_9 + p_10, M_f_7 = p_5 + p_5 + p_9 + p_10, and M_f_8 = p_2 + p_4 + p_6 + p_9 + p_10. The sets of FBMs identified are \{M_f_1, M_f_2\} (by Algorithm 1), \{M_f_3, M_f_4\} (by Algorithm 1) and \{M_f_5, M_f_6, M_f_7, M_f_8\} (by Algorithm 2).

As stated previously, for M_f_1 = p_3 + p_11 and M_f_2 = p_11 + p_12, we can obtain a monolithic GMEC μ_3 + 2μ_11 + μ_12 ≤ 2 that is equivalent to μ_3 + μ_11 ≤ 1 and μ_11 + μ_12 ≤ 1 that prevent M_f_1 = p_3 + p_11 and M_f_2 = p_11 + p_12 from being reached, respectively. For the two FBMs M_f_3 and M_f_4, a GMEC 2μ_2 + μ_3 + 2μ_4 + μ_12 ≤ 4 can be computed. It is equivalent to μ_2 + μ_3 + μ_4 ≤ 2 and μ_2 + μ_4 + μ_12 ≤ 2.

Let us now consider the four FBMs M_f_5 = p_5 + p_6 + p_9 + p_10, M_f_6 = p_3 + p_6 + p_9 + p_10, M_f_7 = p_3 + p_5 + p_6 + p_9 + p_10, and M_f_8 = p_2 + p_4 + p_6 + p_9 + p_10. We have β = {p_9, p_10} as well as

\[
\begin{align*}
\alpha_5 + \alpha_6 + \alpha_9 + \alpha_{10} & = k + 1 \\
\alpha_3 + \alpha_6 + \alpha_9 + \alpha_{10} & = k + 1 \\
\alpha_3 + \alpha_5 + \alpha_9 + \alpha_{10} & = k + 1 \\
\alpha_2 + \alpha_4 + \alpha_9 + \alpha_{10} & = k + 1 \\
\alpha_9 - \alpha_{10} & = d = 4.
\end{align*}
\]

A solution to the above equations is \alpha_2 = \alpha_4 = 1, \alpha_3 = \alpha_5 = \alpha_6 = 2, \alpha_9 = \alpha_{10} = 4, and k = 11. Thus, the computed GMEC is μ_2 + 2μ_5 + μ_4 + 2μ_6 + 4μ_9 + 4μ_{10} ≤ 11. It can be verified that the computed GMEC is equivalent to the four GMECs μ_5 + μ_6 + μ_9 + μ_{10} ≤ 3, μ_3 + μ_6 + μ_9 + μ_{10} ≤ 3, μ_3 + μ_5 + μ_9 + μ_{10} ≤ 3, and μ_2 + μ_4 + μ_6 + μ_9 + μ_{10} ≤ 4 that prohibits the reachability of the four FBMs, respectively.

Finally we have three GMECs as shown in Table 5. When the three monitors are added to the uncontrolled Petri net model shown in Fig. 3, a live Petri net system can be obtained with maximally permissive behaviour, i.e., 205 reachable states. Table 6 shows the performance comparison for the deadlock control policies available in the literature based on the numbers of monitors, sum of arcs weight, and reachable states in the controlled system.

A flexible manufacturing cell shown in Fig. 6 has four machine tools M1-M4. Each machine tool can hold two parts at a time. The cell also contains three robots R1-R3, and each of them can hold one part at a time. Parts enter the cell through three loading buffers I1-I3, and leave the cell through three unloading buffers O1-O3. Three part types J1-J3 are produced. The machine tools perform operations on raw parts and the robots deal with the movements of parts.
ON STRUCTURAL REDUCTION OF LIVENESS-ENFORCING PETRI NET SUPERVISORS

Table 6 Performance comparison of deadlock control policies for the model in Fig. 3

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of monitors</td>
<td>8</td>
<td>8</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>Sum of arc weights</td>
<td>37</td>
<td>37</td>
<td>20</td>
<td>43</td>
<td>20</td>
<td>3</td>
</tr>
<tr>
<td>No. of tokens</td>
<td>19</td>
<td>19</td>
<td>12</td>
<td>23</td>
<td>12</td>
<td>19</td>
</tr>
<tr>
<td>No. of states</td>
<td>205</td>
<td>205</td>
<td>205</td>
<td>205</td>
<td>205</td>
<td>205</td>
</tr>
</tbody>
</table>

Fig. 6. (a) The layout of an FMS, and (b) the production routes.

Its Petri net model is depicted in Fig. 5. The place set has the following partition: \( P_R = \{p_{20}, p_{22}, \ldots, p_{26}\} \), \( P_o = \{p_1, p_5, p_{14}\} \), and \( P_A = \{p_2, \ldots, p_4, p_6, \ldots, p_{13}, p_{15}, \ldots, p_{19}\} \). The Petri net model has 26,750 reachable states, among which 21,581 are legal states. The monitors for this Petri net model are provided in Table 7 and the corresponding supervisor leads to a controlled system with 21,581 reachable states. The sets of FBMs identified by the proposed methods are \( \Theta_1 = \{M_1, M_2\} \) (by Algorithm 1), \( \Theta_2 = \{M_{f_5}, M_{f_9}\} \), \( \Theta_3 = \{M_{f_5}, M_{f_8}, M_{f_9}\} \) (by Algorithm 2), \( \Theta_4 = \{M_{f_{10}}, M_{f_{11}}\} \), \( \Theta_5 = \{M_{f_{13}}, M_{f_{14}}\} \), and \( \Theta_6 = \{M_{f_4}, M_{f_{17}}\} \) (by Algorithm 3). Note that \( M_{f_5}, M_{f_6}, M_{f_{12}} \), and \( M_{f_{15}} \) are unidentified by three algorithms.

For \( \Theta_1, \Theta_2, \Theta_3, \Theta_4, \Theta_5, \) and \( \Theta_6 \), we obtain six monolithic GMECs (place invariants): \( \mu_2 + 2\mu_3 + \mu_8 \leq 4, 3\mu_{11} + \mu_{16} + 2\mu_{17} \leq 7, 2\mu_{11} + 4\mu_{12} + 4\mu_{13} + 5\mu_{15} + 5\mu_{16} \leq 7, \mu_2 + 2\mu_3 + \mu_8 + 2\mu_9 + 2\mu_{13} + 2\mu_{15} + 2\mu_{16} \leq 10, \mu_2 + 2\mu_3 + \mu_8 + 2\mu_9 + 2\mu_{11} + 2\mu_{15} + 2\mu_{16} \leq 12, \) and \( \mu_6 + \mu_7 + \mu_9 + \mu_{11} + \mu_{11} + 8\mu_{12} + 8\mu_{16} + \mu_{18} \leq 23 \), respectively. Finally we have 10 monitors as shown in Table 8. The 10 monitors lead to a live controlled system with maximally permissive behaviour, i.e., 21581 reachable states. Table 9 shows the performance comparison of the deadlock control policies in the literature based on the numbers of tokens, monitors, sum of arcs weight and reachable states.

A Petri net model of an FMS taken from Chen et al. (2011); Chen & Li (2013) is further considered. As shown in Fig. 7, there are 48 places and 38 transitions. The places can be considered as the collection of \( P_o^o = \{p_1, p_{10}, p_{16}, p_{22}, p_{31}\} \), \( P_R = \{p_{37}, p_{38}, \ldots, p_{48}\} \) and \( P_A = \{p_2, \ldots, p_9, p_{11}, \ldots, p_{15}, p_{17}, \ldots, p_{21}, p_{23}, \ldots, p_{32}, \ldots, p_{46}\} \). The Petri net model has 142,865,280 reachable states, among which 84,489,428 and 40319,764 states are legal and illegal, respectively. An optimal supervisor computed for this Petri
Table 7  Place invariants and monitors computed for the S³PR shown in Fig. 5 (Chen et al., 2011; Chen & Li, 2013)

<table>
<thead>
<tr>
<th>FBM</th>
<th>( P_i )</th>
<th>( C_i )</th>
<th>( C_i^* )</th>
<th>( M_0 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p_2 + p_3 )</td>
<td>( \mu_2 + \mu_3 \leq 2 )</td>
<td>( C_1 )</td>
<td>( t_{i1} )</td>
<td>( t_{i1} )</td>
</tr>
<tr>
<td>( 2p_3 + p_8 )</td>
<td>( \mu_3 + \mu_8 \leq 2 )</td>
<td>( C_2 )</td>
<td>( t_{i4}, t_{i3} )</td>
<td>( t_{i3}, t_{i2} )</td>
</tr>
<tr>
<td>( 2p_{11} + p_{17} )</td>
<td>( \mu_{11} + \mu_{17} \leq 2 )</td>
<td>( C_3 )</td>
<td>( t_{i8}, t_{i18} )</td>
<td>( t_{i7}, t_{i17} )</td>
</tr>
<tr>
<td>( p_{12} + 2p_{16} )</td>
<td>( \mu_{12} + \mu_{16} \leq 2 )</td>
<td>( C_4 )</td>
<td>( t_{i9}, t_{i17} )</td>
<td>( t_{i8}, t_{i16} )</td>
</tr>
<tr>
<td>( 2p_{13} + p_{15} )</td>
<td>( \mu_{13} + \mu_{15} \leq 2 )</td>
<td>( C_5 )</td>
<td>( t_{i10}, t_{i16} )</td>
<td>( t_{i9}, t_{i15} )</td>
</tr>
<tr>
<td>( 2p_{11} + 2p_{16} )</td>
<td>( \mu_{11} + \mu_{16} \leq 3 )</td>
<td>( C_6 )</td>
<td>( t_{i8}, t_{i17} )</td>
<td>( t_{i7}, t_{i16} )</td>
</tr>
<tr>
<td>( p_{12} + p_{13} + p_{15} + p_{16} )</td>
<td>( \mu_{12} + \mu_{13} + \mu_{15} + \mu_{16} \leq 3 )</td>
<td>( C_7 )</td>
<td>( t_{i10}, t_{i17} )</td>
<td>( t_{i8}, t_{i15} )</td>
</tr>
<tr>
<td>( 2p_{11} + p_{12} + p_{15} + p_{16} )</td>
<td>( \mu_{11} + \mu_{12} + \mu_{15} + \mu_{16} \leq 4 )</td>
<td>( C_8 )</td>
<td>( t_{i9}, t_{i17} )</td>
<td>( t_{i7}, t_{i15} )</td>
</tr>
<tr>
<td>( 2p_{11} + p_{13} + p_{15} + p_{16} )</td>
<td>( \mu_{11} + \mu_{13} + \mu_{15} + \mu_{16} \leq 4 )</td>
<td>( C_9 )</td>
<td>( t_{i8}, t_{i10}, t_{i17} )</td>
<td>( t_{i7}, t_{i9}, t_{i15} )</td>
</tr>
<tr>
<td>( p_{2} + p_{3} + p_{9} + p_{13} + p_{15} + p_{16} )</td>
<td>( \mu_{2} + \mu_{3} + \mu_{9} + \mu_{13} + \mu_{15} + \mu_{16} \leq 5 )</td>
<td>( C_{10} )</td>
<td>( t_{i5}, t_{i10}, t_{i13}, t_{i7} )</td>
<td>( t_{i4}, t_{i9}, t_{i11}, t_{i5} )</td>
</tr>
<tr>
<td>( p_{3} + p_{8} + p_{9} + p_{13} + p_{15} + p_{16} )</td>
<td>( \mu_{3} + \mu_{8} + \mu_{9} + \mu_{13} + \mu_{15} + \mu_{16} \leq 5 )</td>
<td>( C_{11} )</td>
<td>( t_{i5}, t_{i10}, t_{i13}, t_{i7} )</td>
<td>( t_{i3}, t_{i9}, t_{i12}, t_{i5} )</td>
</tr>
<tr>
<td>( p_{6} + 2p_{7} + p_{11} + p_{17} + p_{18} )</td>
<td>( \mu_{6} + \mu_{7} + \mu_{11} + \mu_{17} + \mu_{18} \leq 5 )</td>
<td>( C_{12} )</td>
<td>( t_{i3}, t_{i8}, t_{i19} )</td>
<td>( t_{i1}, t_{i7} )</td>
</tr>
<tr>
<td>( p_{2} + p_{3} + p_{9} + 2p_{11} + p_{15} + p_{16} )</td>
<td>( \mu_{2} + \mu_{3} + \mu_{9} + \mu_{11} + \mu_{15} + \mu_{16} \leq 6 )</td>
<td>( C_{13} )</td>
<td>( t_{i5}, t_{i13}, t_{i17} )</td>
<td>( t_{i4}, t_{i7}, t_{i11}, t_{i5} )</td>
</tr>
<tr>
<td>( p_{3} + p_{8} + p_{9} + 2p_{11} + p_{15} + p_{16} )</td>
<td>( \mu_{3} + \mu_{8} + \mu_{9} + \mu_{11} + \mu_{15} + \mu_{16} \leq 6 )</td>
<td>( C_{14} )</td>
<td>( t_{i5}, t_{i18}, t_{i13}, t_{i17} )</td>
<td>( t_{i3}, t_{i7}, t_{i12}, t_{i15} )</td>
</tr>
<tr>
<td>( p_{6} + 2p_{7} + p_{8} + p_{9} + p_{11} + p_{17} )</td>
<td>( \mu_{6} + 2\mu_{7} + \mu_{8} + \mu_{9} + \mu_{11} + \mu_{17} )</td>
<td>( C_{15} )</td>
<td>( t_{i5}, t_{i8}, t_{i10}, t_{i19} )</td>
<td>( t_{i1}, t_{i6}, t_{i15}, t_{i18} )</td>
</tr>
<tr>
<td>( p_{6} + 2p_{7} + 2p_{8} + p_{9} + p_{11} + p_{13} )</td>
<td>( \mu_{6} + \mu_{7} + \mu_{8} + \mu_{9} + \mu_{11} + \mu_{13} )</td>
<td>( C_{16} )</td>
<td>( t_{i3}, t_{i5}, t_{i8}, t_{i10}, t_{i18} )</td>
<td>( t_{i4}, t_{i9}, t_{i15} )</td>
</tr>
<tr>
<td>( p_{15} + p_{16} + p_{17} )</td>
<td>( \mu_{15} + \mu_{16} + \mu_{17} \leq 9 )</td>
<td>( C_{17} )</td>
<td>( t_{i3}, t_{i5}, t_{i9}, t_{i17}, t_{i9} )</td>
<td>( t_{i4}, t_{i5}, t_{i15}, t_{i18} )</td>
</tr>
</tbody>
</table>

A monolithic GMEC is used to identify FBM s that can be possibly controlled by a monolithic GMEC through solving an optimal supervisor is obtained by including 24 monitors in the uncontrolled Petri net model shown in Fig. 7. This shows that the proposed method can reduce the complexity of the supervisor compared with the available method in the literature.

6. Computational Complexity of the Proposed Approach

This section discusses the computational complexity of the proposed methods. First we consider the three algorithms to identify FBMs that can be possibly controlled by a monolithic GMEC through solving Eq. (6) if they satisfy Definitions 10, 11, or 12.

Obviously, the three FBM identification algorithms have the same complexity. We assume that there are \( n \) monitors in an optimal liveness-enforcing supervisor of an M-net \( (N, M_0) \) with \( N = (P_A \cup P^0 \cup P) \).
Table 8  Monolithic GMECs (place invariants) for the net model in Fig. 5

<table>
<thead>
<tr>
<th>i</th>
<th>$mPI_i$</th>
<th>$C_i$</th>
<th>$C_i^*$</th>
<th>$M_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$mPI_1 = \mu_2 + 2\mu_3 + \mu_8 \leq 4$</td>
<td>$C_1$</td>
<td>$t_4, 2t_{13}$</td>
<td>t_3, t_{11}, t_{12}</td>
</tr>
<tr>
<td>2</td>
<td>$mPI_2 = 3\mu_{11} + \mu_{16} + 2\mu_{17} \leq 7$</td>
<td>$C_3$</td>
<td>$3t_8, 2t_{18}$</td>
<td>$3t_7, t_{16}, t_{17}$</td>
</tr>
<tr>
<td>3</td>
<td>$mPI_3 = 2\mu_{11} + 4\mu_{12} + 4\mu_{13} + 5\mu_{15} + 5\mu_{16} \leq 17$</td>
<td>$C_4$</td>
<td>$4t_{10}, 5t_{17}$</td>
<td>$2t_7, 2t_8, 5t_{15}$</td>
</tr>
<tr>
<td>4</td>
<td>$mPI_4 = \mu_2 + 2\mu_3 + \mu_8 + 2\mu_9 + 2\mu_{13} + 2\mu_{15} + 2\mu_{16} \leq 10$</td>
<td>$C_5$</td>
<td>$2t_5, 2t_{10}, 2t_{13}, 2t_{17}$</td>
<td>$t_3, t_4, 2t_9, t_{11}, t_{12}, 2t_{15}$</td>
</tr>
<tr>
<td>5</td>
<td>$mPI_5 = \mu_2 + 2\mu_3 + \mu_8 + 2\mu_9 + 2\mu_{11} + 2\mu_{15} + 2\mu_{16} \leq 12$</td>
<td>$C_6$</td>
<td>$2t_5, 2t_6, 2t_{13}, 2t_{17}$</td>
<td>$t_3, t_4, 2t_7, t_{11}, t_{12}, 2t_{15}$</td>
</tr>
<tr>
<td>6</td>
<td>$mPI_6 = \mu_6 + \mu_7 + \mu_9 + \mu_{11} + \mu_{11} + 8\mu_{12} + \mu_{15} + 8\mu_{16} + \mu_{18} \leq 23$</td>
<td>$C_7$</td>
<td>$t_3, t_5, 8t_9, 8t_{17}, t_{19}$</td>
<td>$t_1, t_4, 7t_8, t_{15}, 7t_{16}, t_{18}$</td>
</tr>
<tr>
<td>7</td>
<td>$PI_5 = \mu_{13} + \mu_{15} \leq 2$</td>
<td>$C_8$</td>
<td>$t_3, t_{10}, t_{16}$</td>
<td>$t_9, t_{15}$</td>
</tr>
<tr>
<td>8</td>
<td>$PI_{12} = \mu_6 + \mu_7 + \mu_{11} + \mu_{17} + \mu_{18} \leq 5$</td>
<td>$C_9$</td>
<td>$t_5, t_{18}, 2t_{10}, 2t_{17}, t_{19}$</td>
<td>$t_1, t_9, 2t_{15}, t_{18}$</td>
</tr>
<tr>
<td>9</td>
<td>$PI_{15} = \mu_6 + 2\mu_7 + \mu_8 + \mu_9 + \mu_{11} + \mu_{13} + 2\mu_{15} + 2\mu_{16} + \mu_{18} \leq 12$</td>
<td>$C_{10}$</td>
<td>$t_3, t_5, t_8, t_{10}, t_{18}$</td>
<td>$t_1, t_4, t_9, t_{15}$</td>
</tr>
</tbody>
</table>

Table 9  Performance comparison of control policies for the model shown in Fig. 5

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of monitors</td>
<td>6</td>
<td>19</td>
<td>17</td>
<td>13</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>Sum of arc weights</td>
<td>32</td>
<td>112</td>
<td>101</td>
<td>119</td>
<td>763</td>
<td>763</td>
</tr>
<tr>
<td>No. of tokens</td>
<td>18</td>
<td>78</td>
<td>81</td>
<td>123</td>
<td>570</td>
<td>570</td>
</tr>
<tr>
<td>No. of states</td>
<td>6287</td>
<td>21562</td>
<td>21581</td>
<td>21581</td>
<td>21581</td>
<td>21581</td>
</tr>
</tbody>
</table>

$P_h, T, F, W$). Accordingly, we can, without loss of generality, assume that there are $n$ FBMs in the M-net, as in the worst case, a monitor controls at least one FBM. Since we need to consider all subsets (with the minimal cardinality being two) of the set of $n$ FBMs, the complexity of an FBM identification algorithm is $O(C_n^2 + C_n^3 + \ldots + C_n^n)$, i.e., $O(n^2)$.

For the equivalence check of a computed GMEC with a set of GMECs that prohibit the reachability of their corresponding FBMs, we have, at the worst case, to solve $n$ linear programming problems. As known, a linear programming problem can be solved within polynomial time. The work in Megiddo (1984) shows that a linear programming problem with $d$ variables and $n$ constraints can be solved in $O(n)$ time when $d$ is fixed. We conclude that the linear programming problem for the equivalence check is $O(n)$, since we have at most $n$ constraints while the number of variables is bounded by $|P_h|$. In the worst case, we have to perform $n$ times of the equivalence check. As a result, the complexity of the equivalence check is $O(n^2)$. 


Finally, we analyze the complexity of solving Eq. (6). We aim to find its integer solutions for this linear programming problem. Branch-and-bound search algorithms can be used to achieve this purpose. As before, we assume that there are \( n \) GMECs in an optimal liveness-enforcing supervisor. The weighted token constants are denoted by \( k_1, k_2, \ldots, k_n \). In Eq. (6), we have at most \( |P_A| + 1 \) variables, i.e., \( \alpha_1, \alpha_2, \ldots, \alpha_{|P_A|} \), and \( k \). The lower bound of variable \( \alpha_i \) (\( i \in \{1, 2, \ldots, P_A|\} \)) is 1 and the upper bound of variable \( \alpha_i \) is \( \sum_{i=1}^{n} k_i \). For variable \( k \), it falls into the interval \([1, n \sum_{i=1}^{n} k_i]\). As the bounds of each variable are known, the branch-and-bound algorithm to find an integer solution to Eq. (6) is polynomial (Zhang, 2015). In summary, the proposed method in this paper has the complexity of polynomial time.

7. Discussion and Conclusion

The control place (invariants) merging technique proposed in this paper differs from the concept of implicit places. An implicit place (García-Vallés & Colom, 1999) in a Petri net is a place whose removal from a net system does not change/modify the behavior of the system under consideration. It is shown that implicit monitors always exist in a liveness-enforcing Petri net supervisor that is derived from structural analysis techniques such as siphon control. However, even if two or more monitors are necessary, viz., their removal will lead to the loss of liveness of a net system, they can be combined or merged to be one monitor by the proposed method in this paper. That is to say, for a liveness-enforcing Petri net supervisor without implicit monitors, its structure can be further reduced by merging necessary place invariants (monitors). Siphon-based methods including elementary siphons (Li & Zhou, 2004, 2006b) are an important methodology for deadlock prevention, as they can derive a
liveness-enforcing supervisor with neither a complete state nor siphon enumeration. A liveness-enforcing Petri net supervisor derived from siphon control in general consists of redundant or implicit monitors whose removal does not change its liveness. This paper aims to further reduce the structure of a supervisor even if it has no redundant or implicit monitors by merging place invariants associated with control places (monitors). In other words, for a supervisor derived from a siphon control policy, the proposed method in this paper is usually effective to find a structurally simple supervisor. Note that siphon-based deadlock prevention policies simplify the structure of a supervisor by the implicit controllability of dependent siphons Li & Zhou (2008a,c). The methodology proposed in Chen et al. (2011); Chen & Li (2011, 2012); Chen et al. (2012); Chen & Li (2013) aims to provide a unified treatment of a maximally permissive liveness-enforcing supervisor with a minimal structure for flexible manufacturing systems using integer linear programming approaches by dichotomizing the state space of a resource allocation system into a safe subspace and an unsafe subspace on the premise that a complete state enumeration is obtained. By this methodology, an optimal supervisor with a minimal structure can be computed. However, its computation is rather expensive, since the involved integer linear programming problem contains too many integer variables and constraints, which is computationally intractable. To reduce the computational overheads, an iterative method is developed to calculate an optimal supervisor. In this case, redundant monitors in general exist and the proposed monitor merging method can be applied to simplify the structure of such a supervisor.

A structurally simple supervisor usually implies low costs in the stage of validation, verification, and control implementation. This paper reports a method that can combine multiple monitors into one such that the structure of a supervisor is reduced. Experimental studies show the effectiveness of the proposed method. From a computational viewpoint, the proposed method is efficient, as it involves solutions to linear inequalities. The weakness of the proposed method is that it fails to guarantee the minimal supervisory structure. Thus, the future work will focus on the conditions under which a minimal set of monitors can be found.

Acknowledgments

The authors extend their appreciation to the International Scientific Partnership Program ISPP at King Saud University for funding this research Work through ISPP#0079.

References


M. BASHIR ET AL.


**Appendix**

**Table A.1** Place invariants and monitors of the Petri net model shown in Fig. 7 from Chen et al. (2011)

<table>
<thead>
<tr>
<th>FBM</th>
<th>$P_i$</th>
<th>$C_i$</th>
<th>$C_i^*$</th>
<th>$M_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_2 + p_3 + p_4 + p_5 +$</td>
<td>$\mu_2 + \mu_3 + \mu_4 + \mu_5 + \mu_6 + \mu_7$</td>
<td>$C_1$</td>
<td>$i_5, i_7, i_{15}, i_{21}, i_{35}, i_{37}$</td>
<td>$11$</td>
</tr>
<tr>
<td>$p_6 + p_{14} + p_{18} +$</td>
<td>$\mu_{14} + \mu_{18}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$p_{19} + 2p_{33} + p_{35}$</td>
<td>$\mu_{19} + \mu_{32} + \mu_{33} + \mu_{35} \leq 11$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$p_2 + p_3 + p_4 + p_5 +$</td>
<td>$\mu_2 + \mu_3 + \mu_4 + \mu_5 + \mu_6 + \mu_7$</td>
<td>$C_2$</td>
<td>$i_5, i_7, i_{15}, i_{21}, i_{35}, i_{37}$</td>
<td>$11$</td>
</tr>
<tr>
<td>$p_6 + p_{14} + p_{18} +$</td>
<td>$\mu_{14} + \mu_{18}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$p_{28} + p_{32} + 2p_{33} + p_{35}$</td>
<td>$\mu_{28} + \mu_{32} + \mu_{33} + \mu_{35} \leq 11$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$p_2 + p_3 + p_4 + p_5 +$</td>
<td>$\mu_2 + \mu_3 + \mu_4 + \mu_5 + \mu_6 + \mu_7$</td>
<td>$C_3$</td>
<td>$i_5, i_7, i_{15}, i_{21}, i_{35}, i_{37}$</td>
<td>$11$</td>
</tr>
<tr>
<td>$p_{18} + p_{19} + p_{29} +$</td>
<td>$\mu_{18} + \mu_{19} + \mu_{29} + \mu_{35} \leq 8$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$p_{35}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$p_2 + p_3 + p_4 + p_6 +$</td>
<td>$\mu_2 + \mu_3 + \mu_4 + \mu_5 + \mu_6 + \mu_{14}$</td>
<td>$C_4$</td>
<td>$i_4, i_7, i_{15}, i_{21}, i_{35}, i_{37}$</td>
<td>$11$</td>
</tr>
<tr>
<td>$p_{14} + p_{18} + p_{19} +$</td>
<td>$\mu_{14} + \mu_{18}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$p_{29} + 2p_{33} + p_{35}$</td>
<td>$\mu_{29} + \mu_{32} + \mu_{33} + \mu_{35} \leq 11$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$p_2 + p_3 + p_4 + p_6 +$</td>
<td>$\mu_2 + \mu_3 + \mu_4 + \mu_5 + \mu_6 + \mu_{14}$</td>
<td>$C_5$</td>
<td>$i_4, i_7, i_{15}, i_{19}, i_{21}, i_{31}, i_{33}$</td>
<td>$12$</td>
</tr>
<tr>
<td>$p_{14} + p_{18} + p_{20} +$</td>
<td>$\mu_{14} + \mu_{20} + \mu_{21} + \mu_{29} +$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$p_{21} + p_{29} +$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$p_{32} + 2p_{33} + p_{35}$</td>
<td>$\mu_{32} + \mu_{33} + \mu_{35} \leq 12$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$p_2 + p_3 + p_4 + p_6 +$</td>
<td>$\mu_2 + \mu_3 + \mu_4 + \mu_5 + \mu_6 + \mu_{14}$</td>
<td>$C_6$</td>
<td>$i_4, i_7, i_{15}, i_{18}, i_{21}, i_{29}, i_{31}$</td>
<td>$12$</td>
</tr>
<tr>
<td>$p_{14} + p_{18} + p_{21} +$</td>
<td>$\mu_{14} + \mu_{21} + \mu_{27} + \mu_{29} +$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$p_{27} + p_{29} +$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(Continued)
<table>
<thead>
<tr>
<th>Equation</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p_{12} + 2p_{33} + p_{35} )</td>
<td>( \mu_2 + \mu_3 + \mu_4 + \mu_6 + \mu_{14} + \mu_{13} \leq 11 )</td>
<td>( C_1 )</td>
</tr>
<tr>
<td>( p_{12} + 2p_{33} + p_{35} )</td>
<td>( \mu_2 + \mu_3 + \mu_4 + \mu_6 + \mu_{14} + \mu_{13} \leq 12 )</td>
<td>( C_2 )</td>
</tr>
<tr>
<td>( p_{12} + 2p_{33} + p_{35} )</td>
<td>( \mu_2 + \mu_3 + \mu_4 + \mu_6 + \mu_{14} + \mu_{13} \leq 11 )</td>
<td>( C_3 )</td>
</tr>
<tr>
<td>( p_{12} + 2p_{33} + p_{35} )</td>
<td>( \mu_2 + \mu_3 + \mu_4 + \mu_6 + \mu_{14} + \mu_{13} \leq 12 )</td>
<td>( C_4 )</td>
</tr>
<tr>
<td>( p_{12} + 2p_{33} + p_{35} )</td>
<td>( \mu_2 + \mu_3 + \mu_4 + \mu_6 + \mu_{14} + \mu_{13} \leq 11 )</td>
<td>( C_5 )</td>
</tr>
<tr>
<td>( p_{12} + 2p_{33} + p_{35} )</td>
<td>( \mu_2 + \mu_3 + \mu_4 + \mu_6 + \mu_{14} + \mu_{13} \leq 12 )</td>
<td>( C_6 )</td>
</tr>
<tr>
<td>( p_{12} + 2p_{33} + p_{35} )</td>
<td>( \mu_2 + \mu_3 + \mu_4 + \mu_6 + \mu_{14} + \mu_{13} \leq 11 )</td>
<td>( C_7 )</td>
</tr>
<tr>
<td>( p_{12} + 2p_{33} + p_{35} )</td>
<td>( \mu_2 + \mu_3 + \mu_4 + \mu_6 + \mu_{14} + \mu_{13} \leq 12 )</td>
<td>( C_8 )</td>
</tr>
<tr>
<td>( p_{12} + 2p_{33} + p_{35} )</td>
<td>( \mu_2 + \mu_3 + \mu_4 + \mu_6 + \mu_{14} + \mu_{13} \leq 11 )</td>
<td>( C_9 )</td>
</tr>
<tr>
<td>( p_{12} + 2p_{33} + p_{35} )</td>
<td>( \mu_2 + \mu_3 + \mu_4 + \mu_6 + \mu_{14} + \mu_{13} \leq 12 )</td>
<td>( C_{10} )</td>
</tr>
<tr>
<td>( p_{12} + 2p_{33} + p_{35} )</td>
<td>( \mu_2 + \mu_3 + \mu_4 + \mu_6 + \mu_{14} + \mu_{13} \leq 11 )</td>
<td>( C_{11} )</td>
</tr>
<tr>
<td>( p_{12} + 2p_{33} + p_{35} )</td>
<td>( \mu_2 + \mu_3 + \mu_4 + \mu_6 + \mu_{14} + \mu_{13} \leq 12 )</td>
<td>( C_{12} )</td>
</tr>
<tr>
<td>( p_{12} + 2p_{33} + p_{35} )</td>
<td>( \mu_2 + \mu_3 + \mu_4 + \mu_6 + \mu_{14} + \mu_{13} \leq 11 )</td>
<td>( C_{13} )</td>
</tr>
<tr>
<td>( p_{12} + 2p_{33} + p_{35} )</td>
<td>( \mu_2 + \mu_3 + \mu_4 + \mu_6 + \mu_{14} + \mu_{13} \leq 12 )</td>
<td>( C_{14} )</td>
</tr>
<tr>
<td>( p_{12} + 2p_{33} + p_{35} )</td>
<td>( \mu_2 + \mu_3 + \mu_4 + \mu_6 + \mu_{14} + \mu_{13} \leq 11 )</td>
<td>( C_{15} )</td>
</tr>
<tr>
<td>( p_{12} + 2p_{33} + p_{35} )</td>
<td>( \mu_2 + \mu_3 + \mu_4 + \mu_6 + \mu_{14} + \mu_{13} \leq 12 )</td>
<td>( C_{16} )</td>
</tr>
<tr>
<td>( p_{12} + 2p_{33} + p_{35} )</td>
<td>( \mu_2 + \mu_3 + \mu_4 + \mu_6 + \mu_{14} + \mu_{13} \leq 11 )</td>
<td>( C_{17} )</td>
</tr>
<tr>
<td>( p_{12} + 2p_{33} + p_{35} )</td>
<td>( \mu_2 + \mu_3 + \mu_4 + \mu_6 + \mu_{14} + \mu_{13} \leq 12 )</td>
<td>( C_{18} )</td>
</tr>
<tr>
<td>( p_{12} + 2p_{33} + p_{35} )</td>
<td>( \mu_2 + \mu_3 + \mu_4 + \mu_6 + \mu_{14} + \mu_{13} \leq 11 )</td>
<td>( C_{19} )</td>
</tr>
<tr>
<td>( p_{12} + 2p_{33} + p_{35} )</td>
<td>( \mu_2 + \mu_3 + \mu_4 + \mu_6 + \mu_{14} + \mu_{13} \leq 12 )</td>
<td>( C_{20} )</td>
</tr>
</tbody>
</table>

(Continued)
Table A.1 (Continued)

\[ p_2 + p_3 + p_4 + \mu_2 + \mu_3 + \mu_4 + \mu_{14} + C_{20} \]
\[ 2p_{18} + p_{29} + p_{32} + \mu_{18} + \mu_{29} + \mu_{32} + \mu_{33} + 2p_{33} + \mu_{35} \leq 10 \]
\[ p_2 + p_4 + p_6 + p_{14} + \mu_2 + \mu_4 + \mu_6 + \mu_{14} + C_{20} \]
\[ 2p_{28} + p_{32} + \mu_{28} + \mu_{32} + \mu_{33} + 2p_{33} + \mu_{35} \leq 10 \]
\[ p_2 + p_6 + p_{13} + p_{14} + \mu_2 + \mu_6 + \mu_{13} + \mu_{14} + C_{22} \]
\[ 2p_{27} + p_{32} + \mu_{27} + \mu_{32} + \mu_{33} + 2p_{33} + \mu_{35} \leq 11 \]
\[ p_2 + p_6 + p_{14} + p_{18} + \mu_2 + \mu_6 + \mu_{14} + \mu_{18} + C_{24} \]
\[ 2p_{27} + p_{32} + \mu_{27} + \mu_{32} + \mu_{33} + 2p_{33} + \mu_{35} \leq 11 \]
\[ p_2 + p_6 + p_{14} + p_{19} + \mu_2 + \mu_6 + \mu_{14} + \mu_{19} + C_{26} \]
\[ 2p_{33} + \mu_{35} \leq 11 \]
\[ p_2 + p_6 + p_{18} + p_{19} + \mu_2 + \mu_6 + \mu_{18} + \mu_{19} + C_{27} \]
\[ 2p_{33} + \mu_{35} \leq 5 \]
\[ p_2 + p_6 + p_{18} + p_{20} + \mu_2 + \mu_6 + \mu_{18} + \mu_{20} + C_{28} \]
\[ 2p_{33} + \mu_{35} \leq 6 \]
\[ p_2 + p_{18} + p_{20} + \mu_2 + \mu_6 + \mu_{18} + \mu_{20} + C_{28} \]
\[ 2p_{33} + \mu_{35} \leq 6 \]
\[ p_3 + p_4 + p_6 + \mu_3 + \mu_4 + \mu_5 + \mu_6 + \mu_{14} + C_{30} \]
\[ 2p_{18} + p_{19} + \mu_{18} + \mu_{19} + \mu_{32} + \mu_{33} + 3p_{32} + 2p_{33} + p_{34} + p_{35} \]
\[ 3p_4 + p_5 + p_{14} + \mu_3 + \mu_4 + \mu_5 + \mu_{14} + C_{32} \]
\[ 2p_{32} + 2p_{33} + \mu_{32} + \mu_{33} + 3p_{34} + p_{35} \]
\[ 3p_5 + p_6 + p_{13} + \mu_3 + \mu_5 + \mu_6 + \mu_{13} + C_{34} \]
\[ 3p_{14} + p_{18} + p_{19} + \mu_{14} + \mu_{18} + \mu_{19} + \mu_{32} + 3p_{32} + 2p_{33} + p_{34} + p_{35} \]

\[ (Continued) \]

\[ (Continued) \]
### Table A.1 (Continued)

<table>
<thead>
<tr>
<th>p_3 + p_5 + p_6 + p_{13} + \mu_3 + \mu_5 + \mu_6 + \mu_{13} + C_{35}</th>
<th>t_3, t_5, t_7, t_{14}, t_{21}, t_{31}, t_{37}</th>
<th>t_2, t_4, t_6, t_{13}, t_{19}, t_{30}, t_{35}</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>p_{18} + p_{19} + p_{29} + \mu_{18} + \mu_{19} + \mu_{29} + \mu_{34} + \mu_{35} ≤ 8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p_{34} + p_{35}</td>
<td>\mu_{35} ≤ 8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>p_{3} + p_{5} + p_{13} + p_{14} + \mu_{3} + \mu_{5} + \mu_{13} + \mu_{14} + C_{36}</td>
<td>t_3, t_5, t_{15}, t_{30}, t_{37}</td>
<td>t_2, t_{14}, t_{13}, t_{29}, t_{33}</td>
<td>9</td>
</tr>
<tr>
<td>p_{28} + p_{32} + 2p_{33} + \mu_{28} + \mu_{32} + \mu_{33} + \mu_{34} + \mu_{35} ≤ 8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p_{34} + p_{35}</td>
<td>\mu_{35} ≤ 9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>p_{5} + p_{13} + p_{14} + p_{29} + \mu_{5} + \mu_{13} + \mu_{14} + \mu_{29} + \mu_{34} + \mu_{35} ≤ 8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p_{32} + 2p_{33} + p_{34} + p_{35}</td>
<td>\mu_{32} + \mu_{33} + \mu_{34} + \mu_{35} ≤ 8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>p_{2} + p_{5} + p_{28} + p_{29}</td>
<td>\mu_{2} + \mu_{5} + \mu_{28} + \mu_{29} ≤ 3</td>
<td>C_{38}</td>
<td>t_5, t_{31}</td>
</tr>
<tr>
<td>p_{4} + 2p_{29}</td>
<td>\mu_{4} + \mu_{29} ≤ 2</td>
<td>C_{39}</td>
<td>t_4, t_{31}</td>
</tr>
<tr>
<td>p_{5} + p_{13} + p_{28} + p_{29}</td>
<td>\mu_{5} + \mu_{13} + \mu_{28} + \mu_{29} ≤ 3</td>
<td>C_{40}</td>
<td>t_5, t_{14}, t_{31}</td>
</tr>
<tr>
<td>2p_{5} + p_{28}</td>
<td>\mu_{5} + \mu_{28} ≤ 2</td>
<td>C_{41}</td>
<td>t_5, t_{30}</td>
</tr>
<tr>
<td>2p_{6} + p_{19}</td>
<td>\mu_{6} + \mu_{19} ≤ 2</td>
<td>C_{42}</td>
<td>t_7, t_{20}</td>
</tr>
<tr>
<td>2p_{6} + p_{20} + p_{21}</td>
<td>\mu_{6} + \mu_{20} + \mu_{21} ≤ 3</td>
<td>C_{43}</td>
<td>t_7, t_{19}</td>
</tr>
<tr>
<td>p_{7} + p_{8} + p_{20}</td>
<td>\mu_{7} + \mu_{8} + \mu_{20} ≤ 2</td>
<td>C_{44}</td>
<td>t_{9}, t_{19}</td>
</tr>
<tr>
<td>p_{7} + p_{8} + p_{27}</td>
<td>\mu_{7} + \mu_{8} + \mu_{27} ≤ 2</td>
<td>C_{45}</td>
<td>t_{9}, t_{29}</td>
</tr>
<tr>
<td>p_{7} + p_{20} + p_{27}</td>
<td>\mu_{7} + \mu_{20} + \mu_{27} ≤ 2</td>
<td>C_{46}</td>
<td>t_{8}, t_{19}, t_{29}</td>
</tr>
<tr>
<td>p_{8} + p_{20} + p_{21}</td>
<td>\mu_{8} + \mu_{20} + \mu_{21} ≤ 3</td>
<td>C_{47}</td>
<td>t_9, t_{19}</td>
</tr>
<tr>
<td>p_{8} + p_{21} + p_{27}</td>
<td>\mu_{8} + \mu_{21} + \mu_{27} ≤ 2</td>
<td>C_{48}</td>
<td>t_{9}, t_{18}, t_{29}</td>
</tr>
<tr>
<td>p_{11} + p_{12} + p_{24}</td>
<td>\mu_{11} + \mu_{12} + \mu_{24} ≤ 2</td>
<td>C_{49}</td>
<td>t_{13}, t_{25}</td>
</tr>
<tr>
<td>p_{13} + 2p_{29}</td>
<td>\mu_{13} + \mu_{29} ≤ 2</td>
<td>C_{50}</td>
<td>t_{14}, t_{31}</td>
</tr>
<tr>
<td>p_{20} + p_{21} + p_{27}</td>
<td>\mu_{20} + \mu_{21} + \mu_{27} ≤ 2</td>
<td>C_{51}</td>
<td>t_{19}, t_{29}</td>
</tr>
<tr>
<td>p_{29} + 2p_{26}</td>
<td>\mu_{25} + \mu_{26} ≤ 2</td>
<td>C_{52}</td>
<td>t_{27}</td>
</tr>
<tr>
<td>p_{34} + 2p_{35}</td>
<td>\mu_{34} + \mu_{35} ≤ 2</td>
<td>C_{53}</td>
<td>t_{37}</td>
</tr>
</tbody>
</table>
Table A.2. Monolithic GMECs (place invariants) and monitors of the Petri net model shown in Fig. 7

<table>
<thead>
<tr>
<th>i</th>
<th>PLi</th>
<th>PI</th>
<th>Cj</th>
<th>*Cj</th>
<th>Cj</th>
<th>M0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>mPL1 = 4μt1 + 4μt3 + 4μt4 + 2μt2 + 4μt6 + 4μt14 + 4μt19 + 2μt9 + 2μt2 + 2μt20 + 4μt13 + 4μt16 + 4μt13</td>
<td>C1</td>
<td>2t4, 2t5, 4t7, 4t15, 4t21, 2t31, 4t35, 4t37</td>
<td>4t1, 4t14, 2t19, 2t20, 2t26, 4t31, 4t36</td>
<td>43</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>mPL2 = 4μt2 + 4μt3 + 2μt5 + 4μt6 + 4μt14 + 4μt18 + 2μt19 + 2μt2 + 2μt20 + 4μt32 + 4μt33 + 4μt35</td>
<td>C2</td>
<td>4t5, 2t5, 4t7, 4t15, 4t21, 2t31, 4t35, 4t37</td>
<td>4t1, 4t14, 2t19, 2t20, 2t26, 4t31, 4t36</td>
<td>43</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>mPL3 = 3μt2 + 3μt6 + 3μt14 + 2μt19 + 2μt20 + 2μt21 + 3μt27 + 3μt32 + 3μt33 + 3μt35</td>
<td>C3</td>
<td>3t2, 3t7, 3t15, 2t21, 3t29, 3t33, 3t37</td>
<td>3t1, 3t4, 2t7, 3t28, 3t33, 3t36</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>mPL4 = 2μt3 + μt4 + 2μt5 + 2μt6 + μt13 + 2μt18 + 2μt19 + 2μt20 + 2μt31 + 2μt32 + 2μt34 + 2μt35 + 2μt36</td>
<td>C4</td>
<td>t3, 2t8, 2t9, 2t15, 2t20, 2t37</td>
<td>2t2, 2t4, 3t4, 3t14, 3t18, 3t33</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>mPL5 = 4μt2 + 4μt3 + 4μt4 + 4μt6 + 4μt14 + μt10 + 3μt20 + 3μt21 + 3μt27 + 3μt32 + 3μt33</td>
<td>C5</td>
<td>4t4, 4t7, 4t15, 2t21, 4t31, 4t35, 4t37</td>
<td>4t1, 4t4, 3t7, 4t30, 4t31, 4t36</td>
<td>46</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>mPL6 = 4μt2 + 4μt3 + 4μt6 + 4μt14 + 2μt18 + 2μt19 + 2μt20 + 3μt21 + 3μt27 + 3μt32 + 3μt33 + 3μt35</td>
<td>C6</td>
<td>3t4, 4t7, 4t15, 3t19, 2t21, 3t29, 3t33, 4t37, 4t7, 4t15, 3t7, 3t38, 4t30, 4t31, 4t36</td>
<td>47</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>mPL7 = 2μt2 + 2μt3 + μt4 + μt13 + 2μt14 + 2μt18 + 2μt20 + 2μt21 + 2μt31 + 2μt32</td>
<td>C7</td>
<td>t3, 4t4, 2t6, 2t15, 2t20, h1, 3t28, 2t37</td>
<td>2t1, 3t1, 4t4, 2t6, 3t19, 3t30, 4t31, 2t36</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>mPL8 = 2μt2 + 4μt3 + 4μt4 + 4μt6 + 2μt13 + 4μt18 + 4μt32</td>
<td>C8</td>
<td>2t2, 4t7, 6t7, 2t14, 4t21, 4t31, 4t37</td>
<td>2t1, 2t2, 4t7, 6t7, 2t19, 4t30, 4t31, 2t36</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>mPL9 = 2μt3 + μt4 + μt13 + 2μt14 + 2μt20 + 2μt21 + 2μt31 + 2μt32 + 2μt33</td>
<td>C9</td>
<td>t3, t4, 2t6, 2t15, 2t20, 2t37</td>
<td>2t2, 3t14, 4t13, 2t30, 2t36</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>mPL10 = 2μt2 + 4μt3 + 2μt6 + μt13 + 2μt14 + 2μt18 + 2μt20 + 2μt21 + 2μt31 + 2μt32 + 2μt33</td>
<td>C10</td>
<td>2t2, 2t4, 2t7, 2t15, 2t30, 2t31, 4t37, 2t15, 3t19, 2t20, 2t30, 2t36</td>
<td>32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>mPL11 = 2μt3 + μt4 + 2μt5 + μt13 + 2μt14 + 2μt20 + 2μt21 + 2μt31</td>
<td>C11</td>
<td>t3, 3t5, 2t5, 2t30, 2t37</td>
<td>2t2, 3t14, 4t13, 2t30, 2t36</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>mPL12 = 2μt3 + μt4 + μt13 + 2μt28 + 2μt29</td>
<td>C12</td>
<td>2t5, 4t7, 4t31</td>
<td>3t1, 3t4, 3t39, 3t59</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>mPL13 = 3μt7 + 2μt8 + 2μt21</td>
<td>C13</td>
<td>2t8, 2t9, 2t9</td>
<td>3t1, 3t7</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>mPL14 = 2μt8 + 2μt20 + 3μt21 + 2μt27 + 2μt32 + 2μt33 + 2μt35</td>
<td>C14</td>
<td>2t9, 4t18, 2t30, 2t39</td>
<td>2t8, 3t17, 2t58</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>mPL15 = μt4 + μt13 + 2μt28 ≤ 4</td>
<td>C15</td>
<td>3t4, 4t4, 2t31</td>
<td>1t1, 3t13, 2t30</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>PL2 = μt2 + μt6 + μt13 + μt18 + 2μt20 + 2μt21 + 2μt31 + 2μt32 + 2μt33 + 2μt34 + 2μt35</td>
<td>C16</td>
<td>2t2, 4t7, 4t13, 3t17, 3t37</td>
<td>1t1, 3t9, 3t36</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>PL25 = μt2 + μt6 + μt13 + μt18 + μt20 + μt21 + μt31 + μt32 + μt33 + μt35</td>
<td>C17</td>
<td>2t2, 4t7, 4t13, 3t17, 3t37</td>
<td>1t1, 3t9, 3t36</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>PL29 = μt2 + μt18 + μt35</td>
<td>C18</td>
<td>2t2, 4t6, 4t21, 4t37</td>
<td>1t1, 2t20, 3t36</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>PL41 = μt5 + μt28 ≤ 2</td>
<td>C19</td>
<td>t3, 3t30</td>
<td>4t7, 4t39</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>PL42 = μt6 + μt19</td>
<td>C20</td>
<td>t3, 3t30</td>
<td>4t7, 4t39</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>PL43 = μt6 + μt30 + μt31 ≤ 3</td>
<td>C21</td>
<td>t3, 3t30</td>
<td>4t7, 4t39</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>PL49 = μt11 + μt24</td>
<td>C22</td>
<td>t3, 3t30</td>
<td>4t7, 4t39</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>PL52 = μt35 + μt26</td>
<td>C23</td>
<td>t3, 3t30</td>
<td>4t7, 4t39</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>PL53 = μt34 + μt35</td>
<td>C24</td>
<td>t3, 3t30</td>
<td>4t7, 4t39</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>
学霸图书馆

www.xuebalib.com

本文献由“学霸图书馆-文献云下载”收集自网络，仅供学习交流使用。

学霸图书馆（www.xuebalib.com）是一个“整合众多图书馆数据库资源，提供一站式文献检索和下载服务”的24小时在线不限IP图书馆。

图书馆致力于便利、促进学习与科研，提供最强文献下载服务。

图书馆导航：

图书馆首页 文献云下载 图书馆入口 外文数据库大全 疑难文献辅助工具