A Constraint Transformation Technique in Petri Nets with Backward-Conflict-Free Uncontrollable Structures

Ziyue Ma, Zhiwu Li, and Alessandro Giua

July 1, 2015

Abstract

In this paper we study the problem of constraint transformation for Petri nets. We consider a special class of systems in which the uncontrollable subnet is backward-conflict-free, and a new special class of GMECs called singular GMECs. We propose an algorithm to transform a given uncontrollable singular GMEC into an equivalent controllable OR-GMEC. The algorithm is based on the composition technique of GMECs.

Published as:

[Z. Y. Ma, Z. W. Li, A. Giua, "A Constraint Transformation Technique in Petri Nets with Backward-Conflict-Free Uncontrollable Structures," ACC15: 2015 American Control Conference (Chicago, IL, USA), July 1-3, 2015.]

DOI: 10.1109/ACC.2015.7170763.

Z. Ma is with the School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China (e-mail: maziyue@gmail.com), and also with Dipartimento di Ingegneria Elettrica ed Elettronica, Università degli Studi di Cagliari, Cagliari 09123, Italy.

Z. Li is with the School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China, and also with Institute of Systems Engineering, Macau University of Science and Technology, Taipa, Macau (e-mail: zhwli@xidian.edu.cn, systemscontrol@gmail.com).

A. Giua is with Aix Marseille Université, CNRS, ENSAM, Université de Toulon, LSIS UMR 7296, Marseille 13397, France (e-mail: alessandro.giua@lsis.org), and also with Dipartimento di Ingegneria Elettrica ed Elettronica, Università degli Studi di Cagliari, Cagliari 09123, Italy (e-mail: giua@diee.unica.it).

1 Introduction

Generalized Mutual Exclusion Constraints (GMECs) [1] represent an efficient control approach in Petri nets which has drawn lots of attentions in recent years. A single GMEC defines a set of legal markings, and any marking which is not legal should be prohibited by a controller. The GMEC approach has many advantages since a single GMEC can be easily implemented by a single monitor place whose structure can be designed with negligible computational effort. Many type of forbidden marking problems in Petri nets, such as deadlock prevention [2], can be can be solved in the framework of GMECs.

When uncontrollable transitions are present in a Petri net model, as it is common in the supervisory control framework, such an implementation becomes difficult. A GMEC is said to be non admissible if the firing of some uncontrollable transitions will increase its token count thus possibly leading to a violation of the constraint. For a given inadmissible GMEC one need to consider a more restrictive control policy which not only prohibits the forbidden markings but also some other weakly forbidden markings, from which the system may uncontrollably violate the control law.

Up to now, both *on-line* and *off-line* approaches have been proposed to solve this problem. In the on-line approaches, at each step an integer or linear programming problem has to be solved [3]. The off-line approaches seek a solution substituting a given inadmissible GMEC by one or more admissible GMECs: this technique is also called *GMEC transformation*. The off-line approaches have some advantages since they do not require on-line computation. However, in [1] it has been proved that in some cases there does not exist a single GMEC which is equivalent to a given inadmissible GMEC. Moody proposed a method to transform a given inadmissible GMEC into an admissible new GMEC [4]. Their approach is very efficient from a computational point of view but the solution is not guaranteed to be optimal, i.e., some legal markings may no longer be reachable under their control policy. Holloway studied a very similar problem and proposed an algorithm to estimate the maximal number of tokens a place may uncontrollably get from a given marking [5]. Furthermore, Luo *et. al* and Wang *et. al* extensively studied the GMEC transformation problem in different subclass of Petri net systems [6–8], e.g., *forward-synchronization-forward-conflict-free* nets. Besides, in [9] it is proven that the class of monitor based supervisors may not have a supremal element for uncontrollable specifications, and some different approaches have also been proposed to solve the GMEC transformation problem [10].

Recently, Luo *et al.* extended their approach to solve GMEC transformation problems with fairly arbitrary uncontrollable subnet structures [11]. However, we found that some key results in [11] are not correct, as discussed in [12]. Therefore, although we believe that GMEC transformation is an interesting and fruitful technique to explore, the GMEC transformation problem still lacks a general solution.

In this paper, we focus on the Petri net models in which the uncontrollable subnet is a *backward-conflict-*

free net (BCF net). A BCF net is an ordinary Petri net subclass in which each place has at most one input transition. BCF nets are more general than *assembly flow systems* considered by Ma *et. al* [13]. BCF net can model both conflict and synchronization in the working flow. In this paper to simplify the analysis we focus on acyclic BCF nets. We also assume that the initial inadmissible GMEC to be transformed imposes an upper bound on the marking of a subset of places. This type of GMECs, which are fairly general and of practical usage, is commonly used in the supervisory control framework to trim an uncontrollable supervisor [14]. The approach presented in this paper could be used as a basic step to solve the *monolithic supervisor trimming problem* [14], since the legal marking set for a monolithic supervisor can be easily constructed by uncontrollable GMECs which are analogous to the GMECs in Problem 1 in Section III.

To our knowledge, the GMEC transformation problem in BCF nets has not been studied yet. Therefore the contribution of this paper is twofold. First, a special class of GMECs, that we call *singular GMECs*, is defined, and some of its properties are characterized. Secondly, we propose an iterative GMEC transformation approach based on *GMEC composition* to obtain an maximally permissive solution which is a disjunction of GMECs called *OR-GMEC*. Such type of GMEC can be implemented by a Petri net controller [15, 16]. This GMEC composition technique can well handle the conflict-synchronization structure in GMEC transformation.

The paper is organized in six sections. Section II recalls the basic notions on Petri net and GMECs. Section III introduces the notion of singularity of GMECs and its properties are studied. In Section IV an algorithm based on GMEC composition operation is proposed to transform a given GMEC into an equivalent admissible OR-GMEC if the uncontrollable subnet is a BCF net. An illustrative example is presented in Section V. Section VI draws the conclusions. The proof of Proposition 7 is given in the appendix.

2 Preliminaries

2.1 Petri Net

A Petri net is a four-tuple $N = (P, T, Pre, Post)$, where P is a set of m *places* represented by circles; n *transitions* represented by bars; $Pre : P \times T \to \mathbb{N}$ and $Post : P \times T \to \mathbb{N}$ are the *pre-* and *post-incidence functions* that specify the arcs in the net and are represented as matrices in $\mathbb{N}^{m\times n}$ (here $\mathbb{N} = \{0, 1, 2, \ldots\}$). The *incidence matrix* of a net is defined by $C = Post - Pre \in \mathbb{Z}^{m\times n}$ (here $\mathbb{Z} = \{0, \pm 1, \pm 2, \ldots\}$).

For a transition $t \in T$ we define its *set of input places* as ${}^\bullet t = \{p \in P \mid Pre(p,t) > 0\}$ and its *set of output places* as $t^\bullet = \{p \in P \mid Post(p,t) > 0\}$. The notion for ${}^\bullet p$ and $p^\bullet$ are analogously defined.

A *marking* is a vector $M : P \to \mathbb{N}$ that assigns to each place of a Petri net a non-negative integer number of tokens, represented by black dots and can also be represented as a m component vector. We denote by $M(p)$ the marking of place p. A *marked net* $\langle N, M_0 \rangle$ is a net N with an initial marking M_0. We denote by $R(N, M_0)$ the set of all markings reachable from the initial one.

A transition t is *enabled* at M if $M \geq Pre(\cdot, t)$ and may fire reaching a new marking $M' = M_0 + C(\cdot, t)$. We write $M[\sigma\rangle$ to denote that the sequence of transitions σ is enabled at M, and we write $M[\sigma\rangle M'$ to denote that the firing of σ yields M'.

The transition set T can be partitioned into T_c and T_u which represent the controllable and uncontrollable transition set, respectively. A transition $t_u \in T_u$ is not controllable, i.e., it cannot be disabled by control places.

Given a net $N = (P, T, Pre, Post)$ we say that $\hat{N} = (\hat{P}, \hat{T}, \hat{Pre}, \hat{Post})$ is a subnet of N if $\hat{P} \subset P$, $\hat{T} \subset T$ and $\hat{Pre}$ (resp., $\hat{Post}$) is the restriction of Pre (resp., $Post$) to $\hat{P} \times \hat{T}$. $\hat{N}$ is said to be the *uncontrollable subset* of N if $\hat{T} = T_u$ and $\hat{P} = \{p \in P | ({}^\bullet p \cup p^\bullet) \cap T_u \neq \emptyset\}$.

In a net $N = (P, T, Pre, Post)$, a *path* is a sequence of nodes $\pi = x_1 x_2 \cdots x_k$ such that $x_i \in P \cup T$ for all $i = 1, \ldots, k$, and $x_i \in {}^\bullet x_{i+1}$ for all $i = 1, \ldots, k-1$. In an acyclic Petri net, a *maximal path* is a path π such that $|{}^\bullet x_1| = |x_k{}^\bullet| = 0$.

2.2 GMECs

A Generalized Mutual Exclusion Constraint (GMEC) is a pair $(\mathbf{w}, k)$ where $\mathbf{w} \in \mathbb{Z}^m$ and $k \in \mathbb{N}$. A GMEC defines a set of legal markings:

$$\mathscr{L}_{(\mathbf{w},k)} = \{M \in \mathbb{N}^m \mid \mathbf{w}^T \cdot M \leq k\}$$

and a set of admissible markings:

$$\mathscr{A}_{(\mathbf{w},k)} = \{M | \forall \sigma_u \in T_u^*, M[\sigma_u\rangle M' \in \mathscr{L}_{(\mathbf{w},k)}\}$$

OR-GMEC [12]: An *OR-GMEC* is a set of GMECs: $W_{OR} = \{(\mathbf{w}_1, k_1), \ldots, (\mathbf{w}_r, k_r)\}$. An OR-GMEC defines a set of legal markings:

$$\mathscr{L}_{OR}(W_{OR}) = \{M \in \mathbb{N}^m \mid \exists (\mathbf{w}_i, k_i) \in W_{OR}, \mathbf{w}_i^T \cdot M \leq k_i\}$$

and a set of admissible markings:

$$\mathscr{A}_{OR}(W_{OR}) = \{M | \forall \sigma_u \in T_u^*, M[\sigma_u\rangle M' \in \mathscr{L}_{OR}(W_{OR})\}$$

For the sake of simplicity in the following we denote $\mathscr{L}_{OR}(W_{OR})$ by $\mathscr{L}(W_{OR})$ and $\mathscr{A}_{OR}(W_{OR})$ by $\mathscr{A}(W_{OR})$.

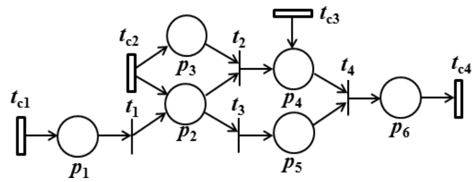

Figure 1: An example of BCF uncontrollable subnet.

3 Singularity of a GMEC and Problem Statement

In this section we introduce some useful notions including *singularity* and *composition* of GMECs. We point out that these notions can also be defined in non-BCF nets. However, in BCF nets they will have some special properties, which will be used later to design our algorithm.

First we give the definition of backward-conflict-free net.

Definition 1 *A* backward-conflict-free net *(BCF net) is an ordinary Petri net in which each place has at most one input transition.* $\triangle$

Example 1 *In the Petri net N_0 in Figure 1, $T_u = \{t_1, t_2, t_3, t_4\}$. The uncontrollable subnet N contains all places and transitions except $t_{c1}, t_{c2}, t_{c3}, t_{c4}$ and it is backward-conflict-free.* $\triangle$

The class of BCF nets strictly includes acyclic *marked graphs*, *synchronization-free nets*, and *assembly-flow systems* [13].

In an uncontrollable subnet which is BCF, the tokens in a place p can only arrive from its unique upstream transition.

Definition 2 *In an ordinary Petri net, we say that a GMEC $(\boldsymbol{w}, k)$ is* singular *if the following conditions hold: (1) there exists at most one t_x such that $\sum_{p \in ||\boldsymbol{w}||} Pre(p, t_x) = 0$ and $\sum_{p \in ||\boldsymbol{w}||} Post(p, t_x) = 1$, and if $Post(p, t_x) = 1$ then $w(p) = 1$ ($||\boldsymbol{w}||$ denotes the support places of $\boldsymbol{w}$); (2) for all $t \neq t_x$, $\boldsymbol{w}^T \cdot C(\cdot, t) \leq 0$; if t_x does not exist, for all t, $\boldsymbol{w}^T \cdot C(\cdot, t) \leq 0$. If $(\boldsymbol{w}, k)$ is singular and t_x exists, $t_x^\bullet \cap ||\boldsymbol{w}|| = \{p_x\}$, we call p_x the* head place *and t_x the* injection transition *of $(\boldsymbol{w}, k)$.* $\triangle$

Example 2 *In the uncontrollable subnet net N in Figure 1, the GMEC $M(p_4) + M(p_6) \leq 1$ is singular with the head place p_4 and the injection transition t_2. The GMEC $M(p_3) + M(p_4) + M(p_6) \leq 3$ is also singular with no head place nor injection transition. On the contrary, the GMECs $2M(p_4) + M(p_6) \leq 1$ and $M(p_3) + M(p_4) + 2M(p_6) \leq 3$ are not singular. The concept is that there exists at most one transition t_x the firing of which will put one token into a certain support place p_x of $\boldsymbol{w}$ with weight 1 but will not remove tokens from* *any support place of $\boldsymbol{w}$, and the firing of all other transitions will not increase the token count of $\boldsymbol{w}$. We also note that in a singular GMEC its support places (i.e., $||\boldsymbol{w}||$) are not necessary in one path. For instance, $M(p_2)+M(p_4)+M(p_5)+2M(p_6) \leq 3$ is also a singular GMEC.* △

The singularity of GMECs can be defined in arbitrary type of nets, not necessarily in BCF nets. However, as will be shown in this paper, singular GMECs in BCFs net have some special properties which are quite useful in constructing the GMEC transformation algorithm. Moreover, to verify if a GMEC is singular is simple: to check Condition 1 one could simply check the incident matrix of the net, and to check Condition 2 we only have to do $|T_u|$ vector multiplications.

Since the net is assumed ordinary, it is not difficult to prove that $|t_x^\bullet \cap ||\mathbf{w}||| = 1$, i.e., p_x is unique if t_x exists. Then we have the following property for singular GMECs.

Proposition 1 *In a BCF subnet, for any singular GMEC the firing of any transition t will increase its token count by one at most.*

Proof: Trivial, since for a singular GMEC $(\mathbf{w},k)$ with its inject transition t_x, the firing of t_x will increase its token count by one, while all other transition will never increase its token count. ■

Then we define two operations on singular GMECs.

Definition 3 *Given a singular GMEC $(\boldsymbol{w},k)$ with its head place p_x and injection transition t_x such that $\{^\bullet t_x\} = \{p_1,\ldots,p_{\bar{r}}\}$, the elementary GMEC set of $(\boldsymbol{w},k)$ is defined as $C_{(\boldsymbol{w},k)} = \{(\boldsymbol{w}_i,k_i), 1 \leq i \leq \bar{r}\}$, in which each $(\boldsymbol{w}_i,k_i)$ is computed by the following equations:*

$$\begin{cases} w_i(p) = w(p_x), p = p_i \\ w_i(p) = w(p), else \\ k_i = k \end{cases} \tag{1}$$

△

In brief, for each place p_i in $^\bullet t_x$ a new GMEC $(\mathbf{w}_i,k_i)$ is put in $C_{(\mathbf{w},k)}$ in which the weight of p_i is increased to the weight of p_x. The following proposition shows that in a BCF net, for a singular $(\mathbf{w},k)$, all GMECs in $C_{(\mathbf{w},k)}$ are singular.

Proposition 2 *Given a Petri net N which is BCF, for any singular GMEC $(\boldsymbol{w},k)$ with its head place p_x and its injection transition $t_x : |^\bullet t_x| = \bar{r}$, then any single GMEC $(\boldsymbol{w}_i,k_i)$ in $C_{(\boldsymbol{w},k)}$ is singular.*

Proof: Consider the arbitrary i-th GMEC $(\mathbf{w}_i, k_i)$ in $C_{(\mathbf{w},k)}$. Since $(\mathbf{w}, k)$ is singular with its injection transition t_x, it must hold $w(p_i) = 0$, and the only weight change from $\mathbf{w}$ to $\mathbf{w}_i$ is that of p_i. Therefore the firing of any $t \notin {}^\bullet p_i$ will not change the token count of $(\mathbf{w}_i, k_i)$. Since the net is BCF, there may exist at most one $t'_x \in {}^\bullet p_i$, and then (p_i, t'_x) is the only pair which satisfies the condition in Definition 2. Therefore $(\mathbf{w}_i, k_i)$ is singular. ■

Then we introduce the composition of two singular GMECs.

Definition 4 *Given two singular GMECs $(\boldsymbol{w}_a, k_a)$ and $(\boldsymbol{w}_b, k_b)$ which share the same head place $p_{x_a} = p_{x_b} = p_x$, their composition GMEC $(\boldsymbol{w}_{ab}, k_{ab})$ is computed by the following equations:*

$$\begin{cases} w_{ab}(p_x) = w_a(p_x) \\ w_{ab}(p) = w_a(p) + w_b(p), p \neq p_x \\ k_{ab} = [(k_a+1)+(k_b+1)] - 1 \end{cases} \tag{2}$$

△

Also we have a proposition to show that in a BCF net, the composition of two singular GMECs is also singular.

Proposition 3 *In a BCF net, (($\boldsymbol{w}_a, k_a$) and ($\boldsymbol{w}_b, k_b$) are singular) $\Rightarrow$ (($\boldsymbol{w}_{ab}, k_{ab}$) is singular).*

Proof: Since the net is BCF, the only input transition of p_x is t_x. For all transitions $t \neq t_x$, for all $M_1[t\rangle M_2$, it holds $\mathbf{w}_a^T \cdot M_1 \geq \mathbf{w}_a^T \cdot M_2$ and $\mathbf{w}_b^T \cdot M_1 \geq \mathbf{w}_b^T \cdot M_2$. Therefore $(\mathbf{w}_a + \mathbf{w}_b)^T \cdot M_1 \geq (\mathbf{w}_a + \mathbf{w}_b)^T \cdot M_2$. Now consider the GMEC $(\mathbf{w}_{ab}, k_{ab})$. Notice that $t \notin {}^\bullet p_x$, i.e., the firing of t will not change the token number in p_x, we have $(\mathbf{w}_a + \mathbf{w}_b)^T \cdot M_1 = \mathbf{w}_{ab}^T \cdot M_1, (\mathbf{w}_a + \mathbf{w}_b)^T \cdot M_2 = \mathbf{w}_{ab}^T \cdot M_2$. Therefore we have $\mathbf{w}_{ab}^T \cdot M_1 \geq \mathbf{w}_{ab}^T \cdot M_2$, indicating that the firing of any $t \neq t_x$ will not increase the token count of $(\mathbf{w}_{ab}, k_{ab})$.

Now consider the transition t_x, since t_x is also the injection transition for $(\mathbf{w}_a, k_a)$ and $(\mathbf{w}_b, k_b)$, and $||\mathbf{w}_{ab}|| = ||\mathbf{w}_a|| \cup ||\mathbf{w}_b||$, then it must hold $\sum_{p \in ||\mathbf{w}_{ab}||} Pre(p, t_x) = 0, \sum_{p \in ||\mathbf{w}_{ab}|| \setminus p_x} Post(p, t_x) = 0$. Since $\mathbf{w}_{ab}(p_x) = w_a(p_x) = 1$, it holds $Post(p_x, t_x) = 1$ and consequently $\sum_{p \in ||\mathbf{w}_{ab}||} Post(p, t_x) = 1$. Therefore $(\mathbf{w}_{ab}, k_{ab})$ is singular. ■

Corollary 1 *(1) Given a singular GMEC $(\boldsymbol{w}, k)$ with its injection transition t_x, the firing of t_x will not change the token count of $(\boldsymbol{w}', k') \in C_{(\boldsymbol{w},k)}$. (2) Given two singular GMECs $(\boldsymbol{w}_a, k_a)$ and $(\boldsymbol{w}_b, k_b)$ where $p_{x_a} = p_{x_b} = p_x$ and $w_a(p_x) = w_b(p_x)$, then the firing of any transition $t \neq t_x$ will not increase the token count of $(\boldsymbol{w}_a, k_a), (\boldsymbol{w}_b, k_b), (\boldsymbol{w}_{ab}, k_{ab})$.*

Proof: Straightforward from Definition 3 and 4. ■

Before the end of this section we make a comment. For the uncontrollable subnet N and an arbitrary GMEC control law, there may exists some places whose tokens will never uncontrollably flow to the support of $\mathbf{w}$. So practically we only need to consider the uncontrollable subnets which may potentially increase the token count of $(\mathbf{w}, k)$, since the uncontrollable evolution of the former will never lead to a violation of the control law.

Definition 5 *The uncontrollable subnet of a place p contains all nodes x in $P \cup T$ such that there is a directed path from x to p in N.*

Finally we state the problem which we will study in the remaining part of this paper.

Problem 1 *Consider a Petri net N_0 and a GMEC $(\boldsymbol{w}_0, k_0) : M(p_0) \leq k_0$. Assume that the uncontrollable subnet of p_0 is a backward-conflict-free system N. Determine an equivalent controllable OR-GMEC W_{OR} such that $\mathscr{L}(W_{OR}) = \mathscr{A}_{(\boldsymbol{w}_0, k_0)}$.* △

Example 3 *Consider again the net in Figure 1 with the initial GMEC $(\boldsymbol{w}_0, k_0) = ([0,0,0,0,0,1], 1)$, i.e., $M(p_6) \leq 1$. This initial GMEC meets the condition in Problem 1. The uncontrollable subnet of p_6 contains all places and transitions except $t_{c1}, t_{c2}, t_{c3}, t_{c4}$ and it is backward-conflict-free.* △

We believe that the assumption on the initial GMEC is not too restrictive since in the monolithic supervisor trimming problem and transition disabling problem, the initial GMEC usually is in the form:

$$(\mathbf{w}, k_2) : k_1 M(p) \leq k_2.$$

This type of GMEC can always be converted into its equivalent form:

$$(\mathbf{w}', k') : M(p) \leq \lfloor k_2/k_1 \rfloor$$

where $\lfloor x \rfloor$ denotes the maximal integer which does not exceed x. Furthermore, many real systems, such as the assembly workflows discussed in [13], contains both conflict places and synchronization transitions. These type of systems can be modeled by backward-conflict-free nets. We also notice the following result:

Proposition 4 *For Problem 1, the GMEC $(\boldsymbol{w}_0, k_0) : M(p_0) \leq k_0$ is singular.*

4 GMEC Transformation in BCF Uncontrollable Subnets

In this section we first present the following algorithm to solve Problem 1 and then will explain it.

Algorithm 1 *Computing a controllable OR-GMEC equivalent to a GMEC* $(\boldsymbol{w}_0, k_0) : M(p_0) \leq k_0$*.*

INPUT: A GMEC $(\boldsymbol{w}_0, k_0)$ *on* p_0*, and the BCF uncontrollable subnet* N *of* p_0*.*

OUTPUT: $\mathscr{L}_{OR}(W_{OR}) = \mathscr{A}_{(\boldsymbol{w}_0, k_0)}$

1. *Initialize two sets* $\mathscr{P}_u := P, \mathscr{P}_c := \emptyset$ *for places and two sets* $\mathscr{T}_u := T, \mathscr{T}_c := \emptyset$ *for transitions;*
2. *Initialize the counter* $s = 0$*. Initialize the OR-GMEC set* W_0 *containing the initial GMEC* $(\boldsymbol{w}_0, k_0)$*, and initialize* $W_C = W_D = \emptyset$*;*
3. *Move* p_0 *from* $\mathscr{P}_u$ *to* $\mathscr{P}_c$*;*
4. *Search* $\mathscr{T}_u$ *to find a transition* t_x *such that: (1)* t_x *is the injection transition of some GMECs in* W_s*; (2)* $t_x{}^\bullet \subseteq \mathscr{P}_c$*. If* $\mathscr{T}_u = \emptyset$*, goto Step 10;*
5. *For all* $(\boldsymbol{w}_i, k_i) \in W_s$ *such that* $\boldsymbol{w}_i^T \cdot C(\cdot, t_x) > 0$*, add them to* W_C *and then delete them from* W_s*;*
6. *In* W_C*, use* $C_{(\boldsymbol{w}_i, k_i)}$ *instead of each* $(\boldsymbol{w}_i, k_i)$*;*
7. *Check all GMECs in* W_C *and* W_s*. If there exist two GMECs* $(\boldsymbol{w}_b, k_b) \in W_C$ *and* $(\boldsymbol{w}_a, k_a) \in W_s$ *such that* $p_{x_1} = p_{x_2} = p_x$*, compute their composition GMEC* $(\boldsymbol{w}_{ab}, k_{ab})$ *and put it in* W_D*;*
8. $W_{s+1} := W_s \cup W_C \cup W_D, W_C := \emptyset, W_D := \emptyset$*. Add* t_x *to* $\mathscr{T}_c$ *and then delete* t_x *from* $\mathscr{T}_u$*;*
9. *For all* $p \in \mathscr{P}_u$*, if* $p^\bullet \subseteq \mathscr{T}_c$*, add* p *to* $\mathscr{P}_c$ *and delete* p *from* $\mathscr{P}_u$*.* $s := s + 1$*, Goto Step 4;*
10. *Output* $W_{OR} = W_s$*, END.*

$\triangle$

We explain how Algorithm 1 works. In Step 1 four sets $\mathscr{P}_u, \mathscr{P}_c, \mathscr{T}_u, \mathscr{T}_c$ are initialized to record the current state of places and transitions: *unchecked, checked*. Step 2 initializes two temporary sets of GMECs W_C, W_D. In Step 3 p_0 is moved from $\mathscr{P}_u$ to $\mathscr{P}_c$ indicating it is checked.

Steps 4 to 9 compose the iteration cycle. In each iteration the new generated GMECs are based on elementary GMEC computation and/or GMEC composition. Since the initial GMEC $(\mathbf{w}_0, k_0)$ is singular, in each iteration s, all GMECs in W_s are singular (this will be proved shortly). The iteration process plays like a step-by-step analysis of N backward from p_0. In each iteration, an unchecked transition t_x is picked from $\mathscr{T}_u$

in Step 4, indicating the firing of t_x may increase some GMECs in W_s. Then all these GMECs are moved to W_C to be further treated in Step 6. The selection rule of transition t_x is not random but follows the rule given in Step 4. Intuitively speaking, for any conflict place p, any t which is in the upstream of p cannot be picked until all transitions in the downstream of p are checked. This rule is important since it will detect the conflict and do the GMEC composition accordingly.

Since the net is backward conflict free, for each $(\mathbf{w}, k)$ in W_C, the head place p can only uncontrollable get tokens from t_x. In Step 6, each $(\mathbf{w}, k)$ in W_C is substituted by its elementary GMEC set $C_{(\mathbf{w},k)}$. In Step 7, by checking elements in W_C and W_s, the conflict of two paths can be determined. For each pair of GMECs $(\mathbf{w}_a, k_a)$ and $(\mathbf{w}_b, k_b)$, a new constraint $(\mathbf{w}_{ab}, k_{ab})$ will be generated. All composition GMECs are put in W_D.

Step 8 computes W_{s+1} the union of W_s, W_C and W_D, and then moves t_x to $\mathscr{T}_c$ indicating t_x has been checked. In Step 9 the set $\mathscr{P}_u$ is updated, while all fully checked places are moved to $\mathscr{P}_c$. Then another iteration starts until there is no t_x which could increase the token counts of GMECs in W_s, indicating all GMECs in W_s are controllable.

We claim that the OR-GMEC W_{OR} the output of Algorithm 1 is the solution of Problem 1, i.e., $\mathscr{L}_{OR}(W_{OR}) = \mathscr{A}_{(\mathbf{w}_0,k_0)}$. To prove this we need to prove some intermediate results first.

First, we prove that all GMECs in W_s in Algorithm 1 are singular.

Theorem 1 *All GMECs in W_s in Algorithm 1 are singular.*

Proof: The initial GMEC $(\mathbf{w}_0, k_0)$ is obviously singular. From Proposition 2 and 3, in any iteration all new generated GMECs in W_s are singular. Therefore the statement holds. ■

From Theorem 1 and Proposition 1 we immediately have the following propositions.

Proposition 5 *For all iterations j, of Algorithm 1, and for all $t \in T_u$, the firing of t will increase the token count of any GMEC in W_s by one at most.*

Proposition 6 *For all iterations j in Algorithm 1, for all $(\boldsymbol{w}, k)$ in W_s and for all $t \in \mathscr{T}_c$, the firing of t will not increase the token count of $(\boldsymbol{w}, k)$.*

Proof: First, if $t = t_x$ which is picked from $\mathscr{T}_u$ at a given iteration, then after this iteration, t does not increases $(\mathbf{w}, k)$ from Corollary 1. Second, if $t \in \mathscr{T}_c$ does not increase $(\mathbf{w}, k) \in W_s$ in any s-th iteration, then it does not increase $(\mathbf{w}', k') \in W_j$ for any j-th iteration where $j > s$. If not, it indicates that the weight of a place $p \in t^\bullet$ is increased by the elementary GMEC computation in the j-th iteration. However, this cannot happen since t can only be put into $\mathscr{T}_c$ when all its downstream places are checked. ■

From Proposition 6 we have the following theorem.

Theorem 2 *The OR-GMEC the output of Algorithm 1 is controllable.*

Proof: Trivial, since in the end all $t \in T_u$ are in $\mathscr{T}_c$. ■

Then we give the key step to the final result.

Proposition 7 *In Algorithm 1, it holds $\mathscr{A}_{OR}(W_{s+1}) = \mathscr{A}_{OR}(W_s)$.*

Proof: Rather than presenting a detailed formal proof, here we just give some intuitions to show why the result holds. To prove $\mathscr{A}_{OR}(W_{s+1}) = \mathscr{A}_{OR}(W_s)$ we just need to prove that both $\mathscr{A}_{OR}(W_{s+1}) \subseteq \mathscr{A}_{OR}(W_s)$ and $\mathscr{A}_{OR}(W_{s+1}) \supseteq \mathscr{A}_{OR}(W_s)$ hold.

To prove the $\subseteq$ containment is not difficult. Notice that firing any uncontrollable transitions in $\mathscr{T}_c$ will not increase the token count of any GMEC in W, if $(\mathbf{w}, k) \in W_{s+1}$ is satisfied at M, then all reachable markings from M must satisfy some GMEC in W_s.

For the $\supseteq$ containment, if a marking M_0 is not in $\mathscr{A}_{OR}(W_{s+1})$, i.e., M_0 would evolve to a marking M violates all GMECs in $\mathscr{L}_{OR}(W_{s+1})$ by firing only uncontrollable transitions, then from M by repeatedly firing t_x for enough times a new marking M' which violates all GMECs in $\mathscr{L}_{OR}(W_s)$, indicating $\mathscr{A}_{OR}(W_{s+1}) \supseteq \mathscr{A}_{OR}(W_s)$. ■

Finally we can state the main result of this paper.

Theorem 3 *The output of Algorithm 1 satisfies $\mathscr{L}_{OR}(W_{OR}) = \mathscr{A}_{(\mathbf{w}_0, k_0)}$.*

Proof: First, the initial GMEC $(\mathbf{w}_0, k_0)$ is singular. Since the net is BCF, according to Proposition 2 and 3, all GMECs in W_s are singular in the iterations. Therefore Proposition 7 could be repeatedly applied, and finally it holds $\mathscr{A}_{OR}(W_{OR}) = \mathscr{A}_{(\mathbf{w}_0, k_0)}$. According to Theorem 2, $\mathscr{L}_{OR}(W_{OR}) = \mathscr{A}_{OR}(W_{OR})$. Therefore $\mathscr{L}_{OR}(W_{OR}) = \mathscr{A}_{(\mathbf{w}_0, k_0)}$. ■

We have some comments on Algorithm 1 and Theorem 3.

Remark 1 *In the approaches in literatures, in each iteration only one GMEC is modified. Here in each iteration (typically) a package of GMECs in the disjunctive GMEC set is substituted by a new package of GMECs. We notice that $\mathscr{A}_{OR}(W_{s+1}) = \mathscr{A}_{OR}(W_s)$ holds only for the GMEC transformation based on such package substitution. If we do the GMEC transformation for just a single GMEC in the package one by one, although the final result is the same, in the intermediate steps the admissible marking sets are not always identical.* △

Remark 2 *One may have noticed that the GMEC in the final W_{OR} is strongly dependent on the conflict-synchronization structure in the uncontrollable subnet N. However, from our observation two nets N_1 and N_2 containing the same number of conflict places and synchronization transitions may have quite different number of GMECs in W_{OR}s. Therefore the complexity depends on not only the number of conflict places and synchronization transitions, but also their relative positions in N. Up to now, it is difficult to propose a complexity indicator to evaluate the complexity of W_{OR}, or to predict the complexity in a certain case, i.e., there is no direct relation between the complexity of W_{OR} and the size of the uncontrollable subnet. Qualitatively speaking, a large uncontrollable subnet with no conflict places nor synchronization transitions would have a very simple W_{OR}. On the other hand, the alternation of conflict places and synchronization transitions in one path would greatly increase the number of GMECs in W_{OR}. To find a proper index to estimate the complexity of W_{OR} would be part of our future work.* $\triangle$

5 A Case Study

In this section we give an example to illustrate Algorithm 1. To simplify the expression we use $c_{(\cdot)}$ to denote $(\mathbf{w}_{(\cdot)}, k_{(\cdot)})$ in case there is no confusion. The iteration process is listed in Tabel 1.

i	Pick	$\mathscr{P}_u$	$\mathscr{P}_c$	$\mathscr{T}_u$	W_s
0	-	p_1, p_2, p_3, p_4, p_5	p_6	t_1, t_2, t_3, t_4	c_0
1	t_4	p_1, p_2, p_3	p_4, p_5, p_6	t_1, t_2, t_3	c_1, c_2
2	t_2	p_1, p_2	p_3, p_4, p_5, p_6	t_1, t_3	c_2, c_3, c_4
3	t_3	p_1	p_2, p_3, p_4, p_5, p_6	t_1	c_3, c_4, c_5, c_6
4	t_1	$\emptyset$	$p_1, p_2, p_3, p_4, p_5, p_6$	$\emptyset$	c_3, c_7, c_8, c_9

Table 1. Iteration steps of Examp

Example 4 *Consider the Petri net N_0 in Figure 1 and the initial GMEC is $c_0 = (\boldsymbol{w}_0, k_0) = ([0,0,0,0,0,1], 1)$, i.e., $M(p_6) \leq 1$. Initially all places except p_6 are in $\mathscr{P}_u$ and all uncontrollable transitions are in $\mathscr{T}_u$. In W_0 there is only one constraint c_0:*

$$c_0 : M(p_6) \leq 1 \tag{3}$$

In the first iteration, only transition t_4 could be picked since its downstream places (only p_6) are in $\mathscr{P}_c$. In Step 6, c_0 is substituted by C_{c_0}, i.e., two new constraints:

$$\begin{cases} c_1 : M(p_4) + M(p_6) \leq 1 \\ c_2 : M(p_5) + M(p_6) \leq 1 \end{cases} \tag{4}$$

Then $W_1 = \{c_1, c_2\}$ from Step 7. In Step 8 t_4 is moved from $\mathscr{T}_u$ to $\mathscr{T}_c$. In Step 9, since $p_3^\bullet \subseteq \mathscr{T}_c, p_4^\bullet \subseteq \mathscr{T}_c$, p_3 and p_4 are moved from $\mathscr{P}_u$ to $\mathscr{P}_c$.

In the second iteration, there are two transitions t_2 and t_3 which can be picked. Suppose t_2 is picked. Since t_2 is the injection transition of c_1, in this iteration c_1 is substituted by

$$\begin{cases} c_3 : M(p_3) + M(p_4) + M(p_6) \leq 1 \\ c_4 : M(p_2) + M(p_4) + M(p_6) \leq 1 \end{cases} \tag{5}$$

Then $W_2 = \{c_2, c_3, c_4\}$ and t_2 is now in $\mathscr{T}_c$. Since $p_2^\bullet \nsubseteq \mathscr{T}_c$, indicating p_2 is a conflict place and the GMEC composition will be taken in some further iteration(s). Therefore only p_3 is moved to $\mathscr{P}_c$.

In the third iteration, because $p_2 \notin \mathscr{P}_c$, t_1 cannot be picked. Therefore t_3 is picked and c_2 is substituted by c_5:

$$c_5 : M(p_2) + M(p_5) + M(p_6) \leq 1 \tag{6}$$

At this moment, c_5 and c_4 are conflict at p_2. Therefore Step 7 is triggered and a new composition GMEC c_6 is added:

$$c_6 : M(p_2) + M(p_4) + M(p_5) + 2M(p_6) \leq 3 \tag{7}$$

Now $W_3 = \{c_3, c_4, c_5, c_6\}$ and p_2 is moved to $\mathscr{P}_c$. In the final iteration, t_1 is picked and c_4, c_5 and c_6 are substituted by c_7, c_8 and c_9, respectively:

$$\begin{cases} c_7 : M(p_1) + M(p_2) + M(p_4) + M(p_6) \leq 1 \\ c_8 : M(p_1) + M(p_2) + M(p_5) + M(p_6) \leq 1 \\ c_9 : M(p_1) + M(p_2) + M(p_4) + M(p_5) + 2M(p_6) \leq 3 \end{cases} \tag{8}$$

The algorithm ends and outputs $W_{OR} = W_4 = \{c_3, c_7, c_8, c_9\}$. One can verify this OR-GMEC is controllable and optimal, i.e., $\mathscr{L}_{OR}(W_{OR}) = \mathscr{A}_{(w_0,k_0)}$. $\square$

In Moody's approach [4], any input single GMEC would always produce a controllable single GMEC as the solution. For the sake of space, we do not present the detailed procedures of Moody's method here. If one applies Moody's method on this example taken the initial GMEC c_0 as the input, we point out that the output solution could be either c_3, c_7 or c_8. One can easily verify that none of them is an optimal solution: the legal marking set defined by either c_3, c_7 or c_8 is a strict subset of $\mathscr{L}_{OR}(W_{OR})$ that is equivalent to $\mathscr{A}_{c_0}$.

6 Discussion and Conclusion

In this paper a GMEC transformation technique is proposed to transform an uncontrollable singular GMEC into a set of disjunctive controllable ones when the uncontrollable subnet is backward-conflict-free. This could be used as a basic step to solve the *monolithic supervisor trimming problem* [14], since the legal marking set for a monolithic supervisor can be easily defined by means of uncontrollable GMECs of the form considered in Problem 1. We hope to further extend our approach to the systems which contain more general uncontrollable structures. However we conjecture that there does not exist a generalized method to do such transformation for arbitrary GMEC in arbitrary Petri nets since, as we have shown in [17], in some cases the expected optimal Petri net controller does not exists.

References

[1] A. Giua, F. DiCesare, and M. Silva, "Generalized mutual exclusion constraints for Petri nets with uncontrollable transitions," in *Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics*, Chicago, USA, 1992, pp. 947–949.

[2] Z. W. Li, N. Q. Wu, and M. C. Zhou, "Deadlock control of automated manufacturing systems based on Petri nets - a literature review," *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, vol. 42, pp. 437–462, 2012.

[3] F. Basile, C. Carbone, and P. Chiacchio, "Feedback control logic for backward conflict free choice nets," *IEEE Trans. On Automatic Control*, vol. 52, pp. 387–400, 2007.

[4] J. Moody and P. Antsaklis, "Petri net supervisors for DES with uncontrollable and unobservable transitions," *IEEE Trans. on Automatic Control*, vol. 45, pp. 462–476, 2000.

[5] L. Holloway, A. Khare, and Y. Gong, "Computing bounds for forbidden state reachability functions for controlled Petri nets," *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, vol. 34, pp. 219–228, 2004.

[6] J. L. Luo, W. M. Wu, H. Y. Su, and J. Chu, "Supervisor synthesis for enforcing GMECs on a controlled Petri net," in *Proc. American Control Conference*, 2006, pp. 4165–4170.

[7] ——, "Supervisor synthesis for a class of forbidden state problems in Petri nets," in *Proc. American Control Conference*, 2006, pp. 4171–4176.

[8] S. G. Wang, C. Y. Wang, and M. C. Zhou, "A transformation algorithm for optimal admissible generalized mutual exclusion constraints on Petri nets with uncontrollable transitions," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2011, pp. 3745–3750.

[9] F. Basile, P. Chiacchio, and A. Giua, "Suboptimal supervisory control of Petri nets in presence of uncontrollable transitions via monitor places," *Automatica*, vol. 42, pp. 995–1004, 2006.

[10] F. Basile, R. Cordone, and L. Piroddi, "Integrated design of optimal supervisors for the enforcement of static and behavioral specifications in Petri net models," *Automatica*, vol. 49, pp. 3432–3439, 2013.

[11] J. L. Luo, H. Shao, K. Nonami, and F. J. Jin, "Maximally permissive supervisor synthesis based on a new constraint transformation method," *Automatica*, vol. 48, pp. 1097–1101, 2012.

[12] Z. Y. Ma, Z. W. Li, and A. Giua, "Comments on "Maximally permissive supervisor synthesis based on a new constraint transformation method" [Automatica 48 (2012), 1097-1101]," *Automatica*, vol. 51, pp. 131–134, 2015.

[13] ——, "A constraint transformation technique for Petri nets with certain uncontrollable structures," in *Proc. IEEE International Workshop on Discrete Event Systems*, Cachan, France, 2014, pp. 66–72.

[14] A. Giua, "Supervisory control of Petri nets with language specifications," in *Control of discrete-event systems*, C. Seatzu, M. Silva, and J. van Schuppen, Eds. London: Springer, 2013, vol. 433, pp. 235–255.

[15] M. Iordache and P. Antsaklis, "Petri net supervisors for disjunctive constraints," in *Proc. American Control Conference*, 2007, pp. 4951–4956.

[16] Z. Y. Ma, Z. W. Li, and A. Giua, "Petri net controllers for disjunctive generalized mutual exclusion constraints," in *Proc. IEEE Int. Conf. on Emerging Technologies and Factory Automation*, Cagliari Italy, 2013, pp. 1–8.

[17] ——, "An approach to determine controllability of monolithic supervisors," in *Proc. International Federation of Automatic Control*, Capetown, South Africa, 2014, pp. 2429–2434.