Design of Supervisors for Linear Marking Specifications in Labeled Petri Nets

Ziyue Ma, Zhou He, Zhiwu Li, Alessandro Giua

November 30, 2021

Abstract

In this paper we study the problem of enforcing a generalized mutual exclusion constraint (GMEC) in a labeled Petri net that contains indistinguishable transitions by a monitor-based supervisor. We show that a monitor-based supervisor can be designed based on the *influence* of events if the GMEC is deterministic. On the other hand, for a nondeterministic GMEC, we introduce the notion of *dependency* of transitions, which provides a quantitative relation between the transition firings and the occurrence of dependent events. Based on the dependency of transitions, we propose a structural approach to compute a monitor function using *Hilbert basis*, which helps us develop an algorithm to design an online supervisor using the monitor function. Our approach has low online computational load since marking estimation is avoided.

Published as:

[Z. Ma, Z. He, Z. Li, A. Giua, "Design of Supervisors for Linear Marking Specifications in Labeled Petri Nets", *Automatica*, **2022**, 136, Article 110031.]

DOI: 10.1016/j.automatica.2021.110031

Ziyue Ma is with the School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China (e-mail: maziyue@xidian.edu.cn).

Zhou He is with the School of Electrical and Control Engineering, Shaanxi University of Science and Technology, Xi'an 710021, China (e-mail: hezhou@sust.edu.cn).

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

Alessandro Giua is with DIEE, University of Cagliari, Cagliari 09123, Italy (e-mail: giua@unica.it).

This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 61803246, 61873342, 61703321, the China Postdoctoral Science Foundation under Grant No. 2019M663608, Shaanxi Provincial Natural Science Foundation under Grant Nos. 2020JQ-733 and 2019JQ-022, the Fundamental Research Funds for the Central Universities under Grant JB210413, and the Science Technology Development Fund, MSAR, under Grant 0012/2019/A1.

1 Introduction

Petri nets have been proposed as a fundamental model for *discrete event systems* in a wide variety of applications and have been an asset to reduce the computational complexity involving many supervisory control problems [25, 26, 22, 3, 9]. In this paper we focus on the supervisory control problem in a special Petri net model called *labeled Petri nets* (LPNs) [13]. An LPN has been proved to be a useful tool for modeling discrete event dynamic systems with partial observations in many contexts such as supervisory control [13], fault diagnosis [12] and prognosis [30], detectability [29], etc. In an LPN some transitions are *unobservable*, i.e., their firing cannot be detected by a control agent, and some transitions may not be *distinguishable*, i.e., the agent may not determine which one has fired among all those sharing the same label. Moreover, an LPN may have *uncontrollable* transitions whose firings, once enabled, cannot be prevented by a supervisor.

A *marking specification* of a Petri net consists in a set of legal markings, and the control objective of a supervisor is to prevent the plant from reaching any marking that is not legal [13].[1] A widely used class of linear marking specifications is the *generalized mutual exclusion constraints* [11] (GMECs).[2] In a free-labeled net whose transitions are all controllable, a GMEC can be easily enforced by a supervisor represented by a *monitor place* added to the plant net [11]. On the other hand, if a net contains uncontrollable transitions, the problem of enforcing an uncontrollable GMEC can be reduced to the problem of enforcing a set of conjunctive/disjunctive controllable ones using *GMEC transformation* [25, 19, 2, 3, 23, 20], which will then be enforced as a monitor-based structure [16, 17, 22]. These methods are purely structural (i.e., they use the structure information of the plant net) and thus do not require an exhaustive reachability analysis. In addition, their control structures are simple to implement because they only require counting the firing of observable transitions.

Surprisingly enough, the problem of enforcing linear marking specifications in LPNs using structural approaches has not been addressed in the literature as far as we know. Although all the aforementioned methods can be used to design a monitor-based supervisor for GMECs in free-labeled LPNs or LPNs with unobservable transitions, they all implicitly assume that any two observable transitions are *distinguishable*, i.e., each observable transition is assigned a unique label. Since many real systems may not equip a distinct sensor for each observable event, in general this assumption does not hold [13]. In this case, since the marking consistent with an observation may not be unique, solving a control problem requires preliminary to address the issue of *marking estimation* [4, 1, 6]. In such a control scheme, a supervisor computes all consistent markings and takes a control decision accordingly. However, this usually requires enumerating all consistent markings of the plant LPN online, which is again unpractical. State-abstraction techniques were proposed [28, 24] to move such burdensome computations offline, but the resulting control structure is in general very large. Moreover, such control structures computed offline are not parametric and need to be re-computed for any minor change of the initial marking or of the net/observation structure.

In this paper we propose a method to design a *monitor-based* supervisor for GMECs in LPNs using their structural information. The main contributions of this paper are summarized as follows:

- First, we introduce the notion of *deterministic GMECs* and propose a method to design a monitor

[1] The specifications considered in this work are marking specifications (see [22]) that defines a set of "bad" markings. Note that a marking specification does not impose any *final markings* that a plant should eventually reach. On the other hand, *nonblocking specifications* [13] require that some final marking must be reachable from any reachable marking.

[2] All concepts mentioned in Section 1 will be formally introduced in Sections 2 and 3.

function for *structurally deterministic GMECs*. A GMEC is said to be *structurally deterministic* if all transitions with the same label have the same *influence*[3] on it.

- To design a monitor-based supervisor for GMECs that are not structurally deterministic, we define the *dependency of transitions* to efficiently estimate the current token count of a nondeterministic GMEC. A quantitative expression is then established based on the *Hilbert basis* [10] that can be computed offline using existing software packages [5].

- Finally, we propose an algorithm to compute a monitor function for a nondeterministic GMEC using transition dependency. Our approach is based on the structural information of a plant LPN and thus does not require offline reachability analysis or online marking estimation.

Preliminary results related to the approach that we develop in this paper were presented in [21]. In particular, it was assumed in [21] that among all transitions assigned by the same label at most one may have non-zero influence. In this paper such a restrictive assumption is removed. Moreover, notions of deterministic GMECs are formally formulated, and the correctness of the proposed approach is now formally proved.

The paper is organized in seven sections. Section 2 recalls the basic notions of LPNs and GMECs. Section 3 defines deterministic GMECs and proposes a method to design monitor-based supervisor for them. For GMECs that are not structurally deterministic, Section 4 defines and studies the dependency of sets of transitions, and Section 5 develops a monitor-based supervisor. Sections 6 and 7 present an example and draw the conclusions, respectively.

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; T is a set of 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 from places to transitions and transitions to places, respectively, 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 $\mathbb{Z} = \{0, \pm1, \pm2, \ldots\}$).

For a transition $t \in T$ we define the *set of input places* of it as ${}^\bullet t = \{p \in P \mid Pre(p, t) > 0\}$ and the *set of output places* of it as $t^\bullet = \{p \in P \mid Post(p, t) > 0\}$. The notion for ${}^\bullet p$ and $p^\bullet$ are analogously defined. A transition t is called a *source* transition (resp., *sink* transition) if ${}^\bullet t = \emptyset$ (resp., $t^\bullet = \emptyset$).

A *marking* is a function $M : P \to \mathbb{N}$ that assigns to each place of a Petri net a non-negative integer number of tokens, represented by black dots. A marking can also be described as a $|P|$-component vector $M \in \mathbb{N}^{|P|}$. We use $M(p)$ to denote the marking of place p. A *marked net* $\langle N, M_0\rangle$ is a net N with an initial marking M_0.

A transition t is *enabled* at M if $M \geq Pre(\cdot, t)$ and may fire reaching a new marking $M' = M + C(\cdot, t)$. We write $M[\sigma\rangle M'$ to denote that the sequence of transitions $\sigma \in T^*$ is enabled at M and yields M', where T^* denotes the *Kleene closure* of T, consisting of all finite sequences of transitions in T (including the empty

[3]This notion is formally defined in Section 2. In brief, the influence of a transition t on a GMEC is the corresponding token change by the firing of t.

sequence ε). We denote by $R(N, M_0)$ the set of all markings reachable from the initial one. The set of all transition sequences firable from M_0 is denoted as $L(N, M_0)$, i.e., $L(N, M_0) = \{\sigma \in T^* \mid M_0[\sigma\rangle\}$. We denote by $\sigma(T) = k$ the number of occurrence of transitions $t \in T$ in σ. The vector $\mathbf{y}_\sigma$ is the firing vector of $\sigma \in T^*$, i.e., $y_\sigma(t) = \sigma(t) = k$ if transition t occurs k times in σ.

A transition sequence σ is called *repetitive* if $C \cdot \mathbf{y}_\sigma \geq \mathbf{0}$. If a repetitive sequence σ can fire at a marking M, then it can fire infinite number of times from M. Precisely speaking, a repetitive sequence σ satisfies the following property:

$$(M[\sigma\rangle M') \wedge (M' \geq M) \quad \Rightarrow \quad M[\sigma^r\rangle, \forall r \in \mathbb{N}.$$

2.2 Labeled Petri Nets

A *labeled Petri net* (LPN) is a 4-tuple $G = (N, M_0, E, \ell)$, where $\langle N, M_0\rangle$ is a marked net, E is the *alphabet* (a set of labels), and $\ell : T \to E \cup \{\varepsilon\}$ is the *labeling function* that assigns to each transition $t \in T$ either a symbol from E or the *empty string* ε. We use T_e to denote the set of transitions labeled by $e \in E$, i.e., $T_e = \{t \in T \mid \ell(t) = e\}$.

Given a labeling function ℓ, the set of transitions T is naturally partitioned into two disjoint sets $T = T_o \cup T_{uo}$, where $T_o = \{t \in T \mid \ell(t) \in E\}$ is the set of *observable* transitions and $T_{uo} = T \setminus T_o = \{t \in T \mid \ell(t) = \varepsilon\}$ is the set of *unobservable* transitions. A transition t is said to be *uniquely labeled* if $t' \neq t$ implies $\ell(t') \neq \ell(t)$.

The labeling function can be extended to transition sequences. In this case, we write $\ell : T^* \to E^*$ and define: (a) $\ell(\lambda) = \varepsilon$, i.e., the empty firing λ produces no observation; (b) $\ell(\sigma t) = \ell(\sigma)\ell(t)$ for $\sigma \in T^*$ and $t \in T$. We denote $w \in E^*$ the *word* that is observed when the sequence $\sigma \in T^*$ fires, i.e., $w = \ell(\sigma)$. The *language* of G is defined as $L_o(G) = \{w \in E^* \mid (\exists \sigma \in L(N, M_0))\ \ell(\sigma) = w\}$. The *inverse observation* ℓ^{-1} is defined as follows: given $w \in L(G)$, $\ell^{-1}(w) = \{\sigma \in L(N, M_0) \mid \ell(\sigma) = w\}$. Note that the inverse observation depends on both the net structure and the labeling function.

As in [13], an LPN $G = (N, M_0, E, \ell)$ is said to be: (i) *free-labeled* if $(t_1 \neq t_2) \to (\ell(t_1) \neq \ell(t_2))$ and for all $t \in T$, $\ell(t) \neq \varepsilon$ (i.e., $T_{uo} = \emptyset$ and all transitions are distinguishable); (ii) *λ-free labeled* if for all $t \in T$, $\ell(t) \neq \varepsilon$ (i.e., $T_{uo} = \emptyset$); (iii) *arbitrarily labeled* otherwise.

2.3 GMECs and Monitor Places

A widely used marking specification is the *generalized mutual exclusion constraint* [11, 15, 25, 18, 22, 20] (GMEC). A GMEC is a pair $(\mathbf{w}, k)$, where $\mathbf{w} \in \mathbb{Z}^m$ and $k \in \mathbb{Z}$, that defines a set of *legal markings*:

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

The quantity $\mathbf{w}^T \cdot M$ is called the *token count* [22] of $(\mathbf{w}, k)$ at marking M.

Definition 1 *Given an LPN G and a GMEC $(\mathbf{w}, k)$ where $\mathbf{w} \in \mathbb{Z}^m$ and $k \in \mathbb{Z}$, the influence vector η (with respect to $(\mathbf{w}, k)$) is $\eta = \mathbf{w}^T \cdot C = [\eta(t_1), \ldots, \eta(t_n)]$. The value $\eta(t_i) = \mathbf{w}^T \cdot C(\cdot, t_i)$ is called the* influence *of t_i.*

In other words, a transition t has influence $\eta(t) = q > 0$ (resp., $\eta(t) = -q < 0$) if its firing increases (resp., decreases) the token count of a quantity $q \in \mathbb{N}$.

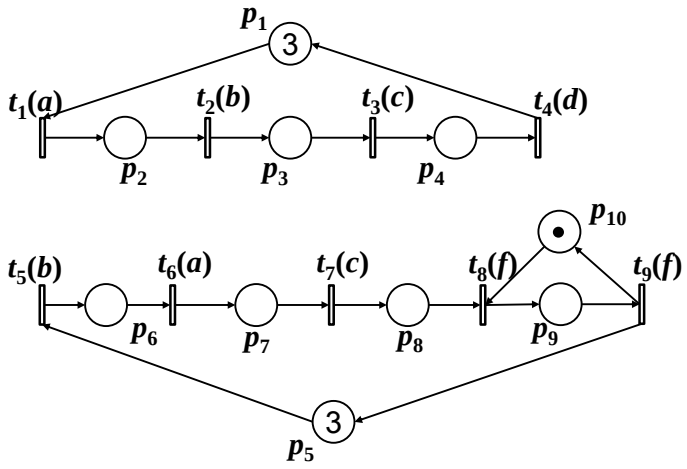

Figure 1: A labeled Petri net used in Examples 1, 2, and 3.

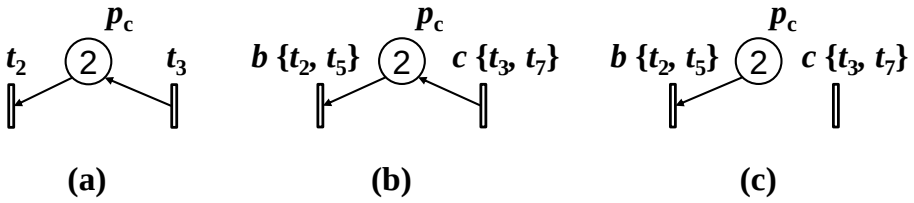

Figure 2: Monitor-based supervisors for the plant net in Figure 1: (a) enforcing $M(p_3) \leq 2$ for the underlying free-labeled net; (b) for the labeled net enforcing $M(p_3) + M(p_6) + M(p_7) \leq 2$; (c) for the labeled net enforcing $M(p_3) \leq 2$.

In this paper we assume that the initial marking is legal, i.e., $M_0 \in \mathcal{L}_{(\mathbf{w},k)}$, otherwise a supervisor that enforces $(\mathbf{w}, k)$ does not exist. In a Petri net $\langle N, M_0 \rangle$ in which all transitions are controllable, a GMEC $(\mathbf{w}, k)$ can be enforced by a *monitor place* [11].

Definition 2 (Monitor place) *[11] Given a marked net $\langle N, M_0 \rangle$, the* monitor place *of $(\mathbf{w}, k)$ is a place p_c with incidence matrix $C'(p_c, \cdot)$ and initial marking $M'_0(p_c)$ which satisfies:*

$$\begin{cases} C'(p_c, \cdot) = -\mathbf{w}^T \cdot C \\ M'_0(p_c) = k - \mathbf{w}^T \cdot M_0 \end{cases} \tag{1}$$

The closed-loop system consisting of marked net $\langle N, M_0 \rangle$ and monitor place p_c is a Petri net with incidence matrix C_{cc} and initial marking $M_{0,cc}$:

$$C_{cc} = \begin{bmatrix} C \\ C'(p_c, \cdot) \end{bmatrix}, \quad M_{0,cc} = \begin{bmatrix} M_0 \\ M'_0(p_c) \end{bmatrix}.$$

Example 1 *Consider the underlying free-labeled Petri net of the LPN in Figure 1, i.e., $\ell(t) = t$ for all $t \in T$, and suppose that all transitions are controllable. Suppose that we want to enforce a GMEC*

$$(\mathbf{w}, k) = ([0, 0, 1, 0, 0, 0, 0, 0, 0, 0]^T, 2).$$

That is, the tokens in p_3 at any reachable marking should not exceed 2. The supervisor can be represented as a monitor place p_c added to the plant net as shown in Figure 2(a).

In a partially controllable LPN, the set T is partitioned into the *set of controllable transitions* T_c and the *set of uncontrollable transitions* T_{uc}, i.e., $T = T_c \cup T_{uc}$.

In general, in a Petri net with uncontrollable transitions, the monitor place of a GMEC, which is defined by Eq. (1), may not be a feasible supervisor, since it may attempt to disable an uncontrollable transition. Structural approaches such as the *GMEC transformation* [25, 19, 2, 23, 20] are proposed to compute a new *controllable* GMEC $(\mathbf{w}', k')$ (or to compute a set of disjunctive/conjunctive GMECs) enforceable by Eq. (1).

3 Monitor-Based Supervisors and Deterministic GMECs

3.1 Monitor-based Supervisors for Free-labeled Nets

We first present some observations on supervisors that are represented by monitor places. One may have noticed that in Example 1, the supervisor represented by p_c only monitors the firings of transition t_2 and t_3 while ignores all other transitions although they are observable. Moreover, given an observation $\sigma \in T^*$, this supervisor applies the following control policy: if $2 - y_{\sigma'}(t_2) + y_{\sigma'}(t_3) \geq 0$ holds, then enable t_2, otherwise disable t_2, where $\sigma' = \sigma t_2$. In other words, the supervisor records the slack of the token count limit (i.e., the quantity $k - \mathbf{w}^T \cdot M$, starting from $k - \mathbf{w}^T \cdot M_0$) by the number of tokens in $M(p_c)$. The supervisor allows t_2 to fire if and only if such slack remains nonnegative after the next firing of t_2.

This observation can be formalized as follows. Given a free-labeled net and a supervisor that is represented as a monitor place p_c, we define its *monitor function* $\zeta : L(N, M_0) \to \mathbb{Z}$ as:

$$\begin{aligned} \zeta(\sigma) = M(p_c) &= M_0(p_c) - \mathbf{w}^T \cdot C \cdot \mathbf{y}_\sigma \\ &= k - \mathbf{w}^T \cdot (M_0 + C \cdot \mathbf{y}_\sigma). \end{aligned}$$

Such a supervisor can be simply described by the following rule: after observing sequence σ, if $\zeta(\sigma t) < 0$, then disable t. Since $\zeta(\sigma t)$ can be easily computed iteratively from $\zeta(\sigma)$, if the plant satisfies the *no concurrency (NC) assumption* [15] (i.e., two transitions cannot fire simultaneously), a corresponding control policy is stated as follows:

Step 1) let $\sigma = \varepsilon$;

Step 2) compute $\zeta(\sigma)$;

Step 3) compute $T_{en} = \{t \in T_c \mid \zeta(\sigma t) \geq 0\}$;

Step 4) enable all transitions in T_{en}, wait until a transition fires in the plant, update $\sigma := \sigma t$, and go to Step 2.

Notice that the value of $\zeta(\sigma)$ is only related to the number of transitions in σ, i.e., $\mathbf{y}_\sigma$, but not their order of occurrence. Such a supervisor is also robust with respect to the initial marking change and the modification of net structures that are irrelevant to t_2, t_3, and p_3.

However, this design of monitor-based supervisors is valid only in free-labeled nets. Although some methods can be generalized to LPNs with unobservable transitions, they all implicitly assume that any two

observable transitions are distinguishable. For LPNs there is no method using structural information analogous to Eq. (1) to design a monitor-based supervisor as far as we know (except our primitive work in [21]). In the rest of this paper we will generalize the intuition above and design a monitor-based supervisor to enforce GMECs in LPNs.

3.2 Problem Set-up in Labeled Petri Nets

We follow the set-up of supervisory control in LPNs in [13]. Given an LPN plant, the set of events E is partitioned into the *set of controllable events* E_c and the *set of uncontrollable events* E_{uc}, i.e., $E = E_c \cup E_{uc}$ and $E_c \cap E_{uc} = \emptyset$. Note that all events in E_c and E_{uc} are observable. Partitioning E into E_c and E_{uc} naturally leads to a partition on the set of transitions $T = T_c \cup T_{uc}$ where $T_c = \{t \in T \mid \ell(t) \in E_c\}$ and $T_{uc} = \{t \in T \mid \ell(t) \in E_{uc} \cup \{\varepsilon\}\}$, i.e., all transitions labeled by a controllable event in E_c are controllable, while others (including all unobservable transitions and all transitions labeled by uncontrollable events in E_{uc}) are uncontrollable. A supervisor ξ runs in parallel with the plant LPN, and for each observation $w \in E^*$ it makes a control decision that is to allow a set of controllable events $\xi(w) \subseteq E_c$ to fire. Note that if the supervisor disables event e, all transitions labeled e are all disabled. In other words, a transition t is firable if it is currently enabled and $\ell(t) \in \xi(w)$.

The specification to be enforced is represented by a GMEC $(\mathbf{w}, k)$. To make this paper concise, we introduce two assumptions that will be used in the sequel, by which we assume that the GMEC to be enforced is *controllable* and *observable* [25].

Assumption 1 *For all $t \in T$, $\eta(t) > 0$ implies $\ell(t) \in E_c$.*

Assumption 2 *For all $t \in T$, $\ell(t) = \varepsilon$ implies $\eta(t) = 0$.*

Assumption 1 requires that the GMEC to be enforced is *controllable*, i.e., any transition whose influence is positive is controllable, which means that a supervisor can always guarantee the control requirement by disabling all controllable events. Assumption 2 requires that the GMEC to be enforced is *observable*, which means that any transition with non-zero influence is observable. With these two assumptions we can focus on the supervisor design problem in LPNs with indistinguishable transitions. On the other hand, if either (or both) of the two assumptions is not satisfied, GMEC transformation techniques in [25] can be used to compute a new GMEC (which although may be suboptimal) that satisfies the assumption. Moreover, the LPNs considered in this work satisfy the widely used *no-concurrency assumption*, i.e., two transitions cannot fire simultaneously. The problem to be studied in this paper is formalized as follows.

Problem 1 *Given an LPN $G = (N, E, \ell, M_0)$ with $E = E_c \cup E_{uc}$ and a GMEC $(\mathbf{w}, k)$ satisfying Assumptions 1 and 2, design a monitor function $\zeta : L_o(G) \to \mathbb{Z}$ such that*

$$\zeta(w) \geq 0 \Rightarrow \forall \sigma \in \ell^{-1}(w), M_0[\sigma\rangle M \in \mathcal{L}_{(\mathbf{w},k)}.$$

Remark 1 *For the sake of simplicity, in this paper we consider the supervisor design problem for a single GMEC. When there are multiple GMECs to be enforced (in the sense of their conjunction), one can design a supervisor for each of them and run all supervisors in parallel with the plant.*

If $M_0[\sigma\rangle M \in \mathcal{L}_{(\mathbf{w},k)} \Leftrightarrow \zeta(w) \geq 0, w = \ell(\sigma)$, the monitor function ζ is said to be *maximally permissive*. Similar to the monitor-based supervisors in free-labeled nets, here we expect that the control decision can be made by simply checking the number of events in the observation w instead of their orders. Hence, we define the *Parikh vector* of $w \in L_o(G)$ as: $\mathbf{z}_w \in \mathbb{N}^{|E|}$ such that $z_w(e) = k$ if event e occurs k times in w.

3.3 Deterministic GMECs

In a free-labeled Petri net, a supervisor is always able to *exactly know* the firing of each transition t to correctly update the value of its monitor function. However, in an LPN it is not always possible due to two types of the nondeterminism:

1. A transition t is *unobservable* if $\ell(t) = \varepsilon$, and hence the supervisor is not able to detect the firing of t;
2. Two transitions t' and t'' are *indistinguishable* if $\ell(t') = \ell(t'') = e \in E$, since their firings produce the same observation. In this case by observing e the supervisor may not be able to determine whether t' or t'' has fired;

The first type of nondeterminism is ruled out by Assumption 2. On the other hand, to characterize the second type of nondeterminism, the following definition is proposed.

Definition 3 *Given an LPN $G = (N, M_0, E, \ell)$, a GMEC $(\mathbf{w}, k)$ is said to be* behaviorally deterministic *(with respect to G) if for all $\sigma, \sigma' \in L(N, M_0)$ such that $\ell(\sigma) = \ell(\sigma')$, the following condition holds:*

$$\mathbf{w}^T \cdot C \cdot \mathbf{y}_\sigma = \mathbf{w}^T \cdot C \cdot \mathbf{y}_{\sigma'}. \tag{2}$$

By Definition 3, a GMEC $(\mathbf{w}, k)$ is behaviorally deterministic if any two sequences $\sigma, \sigma' \in L(G)$ producing the same observation yield markings M and M', i.e., $M_0[\sigma\rangle M, M_0[\sigma'\rangle M'$ with the same token count. Hence for any observation w the token count of a deterministic GMEC can be uniquely determined. Note that this definition is more general than the notion of *distinguishable GMEC* in [21]: for a distinguishable GMEC each transition that modifies its token count is assigned a unique label, and hence a distinguishable GMEC in [21] is always deterministic. Moreover, for a *deterministic LPN* [13], all GMECs are behaviorally deterministic, since $\ell^{-1}(w)$ is a singleton for all observations $w \in L_o(G)$.

To verify the condition in Definition 3 there is no efficient way but to enumerate the reachability space of the plant. On the other hand, the following definition provides us a structural condition that guarantees the determinism of a GMEC.

Definition 4 *Given an LPN $G = (N, M_0, E, \ell)$, a GMEC $(\mathbf{w}, k)$ is said to be* structurally deterministic *(with respect to G) if for all $M \in \mathbb{N}^{|P|}$ and for all $\sigma, \sigma' \in L(N, M)$ such that $\ell(\sigma) = \ell(\sigma')$, $\mathbf{w}^T \cdot C \cdot \mathbf{y}_\sigma = \mathbf{w}^T \cdot C \cdot \mathbf{y}_{\sigma'}$ holds.*

Clearly, a structurally deterministic GMEC is always behaviorally deterministic regardless of the initial marking. A structurally deterministic GMEC satisfies the following two properties: (i) it is observable, (ii) at any marking (not necessarily reachable) any two firing sequences that look alike (i.e., $\ell(\sigma) = \ell(\sigma')$) have the same influence on it, as proved by the following proposition.

Proposition 1 *Given an LPN $G = (N, M_0, E, \ell)$, a GMEC $(\mathbf{w}, k)$ is structurally deterministic (with respect to G) if and only if both of the following conditions hold: (i) for all $t \in T$, $\ell(t) = \varepsilon$ implies $\eta(t) = 0$, and (ii) for all $t, t' \in T$, $\ell(t) = \ell(t') \in E$ implies $\eta(t) = \eta(t')$.*

Proof: (Only if) Suppose that condition (i) is not satisfied. Clearly there exists a marking $M \in \mathbb{N}^{|P|}$ and a transition $t \in T$ such that $M[t\rangle$, $\ell(t) = \varepsilon, \eta(t) \neq 0$. Then we have $\ell(t) = \ell(\varepsilon)$ and $\mathbf{w}^T \cdot C(\cdot, t) \neq \mathbf{w}^T \cdot C \cdot \mathbf{y}_\varepsilon = \mathbf{0}$, which indicates that the GMEC is not structurally deterministic.

On the other hand, suppose that condition (ii) is not satisfied, then there necessarily exists $M \in \mathbb{N}^{|P|}$ and $t, t' \in T$ such that $M[t\rangle, M[t'\rangle$, and $\eta(t) \neq \eta(t')$. Then $\ell(t) = \ell(t')$ and $\mathbf{w}^T \cdot C \cdot \mathbf{y}_t \neq \mathbf{w}^T \cdot C \cdot \mathbf{y}_{t'}$ hold, which implies that the GMEC is not structurally deterministic.

(If) Suppose that both conditions are satisfied. For any sequences $\sigma, \sigma' \in T^*$ such that $\ell(\sigma) = \ell(\sigma')$, the number of occurrences of each event $e \in E$ in σ and σ' must be identical. Since $\eta(t) = \eta(t')$ holds for all t, t' with $\ell(t) = \ell(t') = e$, we denote $\eta(t) = \eta(t') = \eta(e)$. Then, since $\ell(t) = \varepsilon$ implies $\eta(t) = 0$, for any marking M and any sequence σ, σ' firable at M, we have

$$\mathbf{w}^T \cdot C \cdot \mathbf{y}_\sigma = \sum_{t \in T} \eta(t) \cdot y_\sigma(t) = \sum_{e \in E} \eta(e) \cdot z_w(e)$$

and

$$\mathbf{w}^T \cdot C \cdot \mathbf{y}_{\sigma'} = \sum_{t \in T} \eta(t) \cdot y_{\sigma'}(t) = \sum_{e \in E} \eta(e) \cdot z_w(e),$$

which indicates $\mathbf{w}^T \cdot C \cdot \mathbf{y}_\sigma = \mathbf{w}^T \cdot C \cdot \mathbf{y}_{\sigma'}$. □

For LPNs in which each observable transition is assigned a distinct label, the fact that a GMEC is observable also implies that it is structurally deterministic. Since for a structurally deterministic GMEC all transitions with the same label have the same influence, we formally define the *influence* of a label for a structurally deterministic GMEC as follows.

Definition 5 *Given an LPN $G = (N, M_0, E, \ell)$ and a structurally deterministic GMEC $(\mathbf{w}, k)$, the influence of event $e \in E$ is $\eta(e) = \mathbf{w}^T \cdot C(\cdot, t)$ for $t \in T_e$.*

Since in the rest of this section we only consider structurally deterministic GMECs, for simplicity by the term "deterministic GMEC" we refer to a structurally deterministic GMEC. From Definition 4 we have the following result that can be used to compute the token count of a deterministic GMEC $(\mathbf{w}, k)$ for an observation w.

Proposition 2 *Let $G = (N, M_0, E, \ell)$ be an LPN and $(\mathbf{w}, k)$ be a deterministic GMEC. For any observation $w \in L_o(G)$, for all sequences σ such that $\ell(\sigma) = w$ and $M_0[\sigma\rangle M$, the following result holds:*

$$\mathbf{w}^T \cdot M = \mathbf{w}^T \cdot M_0 + \sum_{e \in E} z_w(e) \cdot \eta(e). \tag{3}$$

Proof: Let $\sigma = t_{i_1} t_{i_2} \cdots t_{i_n} \in L(G)$ be an arbitrary sequence such that $\ell(\sigma) = w$. It holds:

$$\mathbf{w}^T \cdot M = \mathbf{w}^T \cdot M_0 + \sum_{j=1}^{n} \mathbf{w}^T \cdot C(\cdot, t_{i_j}). \tag{4}$$

Since $(\mathbf{w}, k)$ is deterministic, $\mathbf{w}^T \cdot C(\cdot, t) = 0$ holds for all t with $\ell(t) = \varepsilon$, and $\mathbf{w}^T \cdot C(\cdot, t) = \eta(e)$ holds for all $t \in T_e$. Hence $\sum_{j=1}^{n} \mathbf{w}^T \cdot C(\cdot, t_{i_j}) = \sum_{e \in E} z_w(e) \cdot \eta(e)$, which implies that Eq. (3) holds. □

Thanks to Proposition 2, the following result provides a monitor function for a deterministic GMEC.

Theorem 1 *Given an LPN $G = (N, M_0, E, \ell)$ with $E = E_c \cup E_{uc}$, the following monitor function for a deterministic GMEC $(\mathbf{w}, k)$ is correct and maximally permissive:*

$$\zeta(w) = k - (\mathbf{w}^T \cdot M_0 + \sum_{e \in E} z_w(e) \cdot \eta(e)) \tag{5}$$

where $w \in L_o(G)$ is an observation of G.

Proof: By Proposition 2, for all $\sigma \in \ell^{-1}(we)$ and $M_0[\sigma\rangle M$, $\mathbf{w}^T \cdot M = \mathbf{w}^T \cdot M_0 + \sum_{e \in E} z_w(e) \cdot \eta(e)$ holds. Hence the right-hand-side of Eq. (5) is negative if and only if $\mathbf{w}^T \cdot M > k$. Therefore the supervisor is correct and maximally permissive. □

Example 2 *Let us again consider the net in Figure 1 with the labels, i.e., $\ell(t_1) = \ell(t_6) = a$, $\ell(t_2) = \ell(t_5) = b$, $\ell(t_3) = \ell(t_7) = c$, $\ell(t_4) = d$, and $\ell(t_8) = \ell(t_9) = f$. Suppose that we want to enforce a GMEC $(\mathbf{w}, k) = ([0, 0, 1, 0, 0, 1, 1, 0, 0, 0], 2)$ that defines legal markings satisfying $M(p_3) + M(p_6) + M(p_7) \leq 2$. One can readily verify: for label a, $\mathbf{w}^T \cdot C(\cdot, t_1) = \mathbf{w}^T \cdot C(\cdot, t_6) = 0$; for label b, $\mathbf{w}^T \cdot C(\cdot, t_2) = \mathbf{w}^T \cdot C(\cdot, t_5) = 1$; for label c, $\mathbf{w}^T \cdot C(\cdot, t_3) = \mathbf{w}^T \cdot C(\cdot, t_7) = -1$; for label f, $\mathbf{w}^T \cdot C(\cdot, t_8) = \mathbf{w}^T \cdot C(\cdot, t_9) = 0$. Hence $(\mathbf{w}, k)$ is (structurally) deterministic. We can enforce this GMEC with monitor function:*

$$\zeta(w) = 2 - z_w(b) + z_w(c).$$

The monitor place implementing this monitor function is shown in Figure 2(b).

On the other hand, a non-deterministic GMEC may not be enforced by a monitor place in a similar way: when observing a label, the supervisor may not be able to infer how the token count has changed. Hence, to ensure that the control requirement is not violated, whenever the supervisor observes label e, it has to assume that the transition labeled e with the largest influence fires. However, as shown in the following example, such a primitive control policy may be too conservative to be used in practice. Therefore, in the next two sections we will propose a different method to design monitor functions for non-deterministic GMECs.

Example 3 *Consider the labeled net in Figure 1. Suppose that we want to enforce a GMEC $(\mathbf{w}, k) = ([0, 0, 1, 0, 0, 0, 0, 0, 0, 0], 2)$ requiring that the tokens in p_3 should not exceed 2. This GMEC is controllable and observable but not deterministic.*

Since a monitor-based supervisor does not have knowledge of the consistent markings, once event b occurs, the supervisor cannot determine whether t_2 or t_5 fires. In order to guarantee that the control requirement is not violated, the supervisor has to assume that such an event b comes from the firing of t_2, since the influence of t_2 ($\eta(t_2) = 1$) is larger than that of t_5 ($\eta(t_5) = 0$); on the other hand, once event c occurs, the supervisor cannot determine whether t_3 or t_7 fires, and hence it assumes that such an event c comes from the firing of t_7 due to the similar reason. This reasoning leads to the following monitor function ζ:

$$\zeta(w) = 2 - z_w(b).$$

Clearly, to guarantee $\zeta \geq 0$ a supervisor disables event b forever after it observes b twice, as illustrated in Figure 2(c), and the controlled system inevitably reaches a deadlock.

4 Dependency of Transitions

In this section, we introduce a notion called the *transition dependency* that is a relation between sets of transitions. We first provide some intuitions by the following example.

Example 4 *Consider again the GMEC* $(\mathbf{w}, k) = ([0, 0, 1, 0, 0, 0, 0, 0, 0, 0], 2)$ *in the plant LPN in Figure 1. If we observe event* b *for* $z_w(b)$ *times and event* c *for* $z_w(c)$ *times. If we notice that event* f *occurs for* $z_w(f)$ *times, we can conclude that transition* t_5 *that has the same label as* t_2 *has fired at least* $\lceil z_w(f)/2 \rceil$ *times. Hence we know that* t_2 *has fired at most* $z_w(b) - \lceil z_w(f)/2 \rceil$ *times. On the other hand, we can conclude that* t_3 *must have fired for a total of* $z_w(d)$ *(*$z_w(d) \leq z_w(c)$*) times whenever we observe event* d *for* $z_w(d)$ *times. Hence, a suitable monitor function is:*

$$\zeta(w) = 2 - (z_w(b) - \lceil z_w(f)/2 \rceil) + z_w(d).$$

A supervisor with this monitor function is correct, and the closed-loop system is deadlock-free: after observing two b*'s the supervisor will disable both* t_2 *and* t_5 *until it observes an* f *or a* d *(which means that place* p_3 *must hold no more than one token).*

In the example above, the information of events d and f is used to bound the actual number of firings of t_2 and t_3. This motivates us to use such structural information to obtain a monitor-based supervisor. In the rest of this section, we introduce the notion of *paths*, *flow subnets*, and *transition dependency* that will be used to find a valid monitor function ζ.

Definition 6 *A* path *in a Petri net* $N = (P, T, Pre, Post)$ *is a sequence of nodes* $\pi = x_0 x_1 \cdots x_{k-1} x_k$ *such that* $x_i \in P \cup T, 0 \leq i \leq k$ *and* $x_{i-1} \in {}^\bullet x_i$ *for all* $i \in \{1, \ldots, k\}$*. A path is said to be* anchored *from* x_0 *to* x_k*, if* $x_i \neq x_0$ *for* $1 \leq i \leq k$ *and* $x_i \neq x_k$ *for* $0 \leq i \leq k-1$*, i.e., nodes* x_0 *and* x_k *are not repeated in the path. The set of all anchored paths from* x' *to* x'' *is denoted as* $\Pi(x', x'')$*, and the set of anchored paths from nonempty set* $X' \subset P \cup T$ *to nonempty set* $X'' \subset P \cup T$ *is denoted as* $\Pi(X', X'')$*, i.e.,:*

$$\Pi(X', X'') = \bigcup_{x' \in X', x'' \in X''} \Pi(x', x'')$$

Definition 7 (Flow Subnet) *Given a pair* (T', T'') *where* $T', T'' \subseteq T$ *are both non-empty, the* (T', T'')-flow subnet*, denoted as* $\hat{N}$*, is the subnet of* N *induced by* $\hat{P}$ *and* $\hat{T}$ *where* $\hat{P} = \{p \in P \mid \exists \pi \in \Pi(T', T''), p \in \pi\}$ *and* $\hat{T} = \{t \in T \mid \exists \pi \in \Pi(T', T''), t \in \pi\} \cup {}^\bullet \hat{P}$*.*

The physical meaning of a flow subnet is the following. In the sequel of this paper we will use the occurrence of some event e to bound the number of firings of a set of transitions $\hat{T}$. To do this, set T_e necessarily *depends* (see Definition 8) on $\hat{T}$, otherwise event e can occur infinite times regardless the firing of $\hat{T}$. In such a case, all places and transitions that do not belong to the $(\hat{T}, T_e)$-flow subnet will not affect the result and hence can be removed to simplify the computation. Although the set $\Pi(T', T'')$ can be infinite due to presence of cycles, sets $\hat{P}$ and $\hat{T}$ are finite and can be computed by structural analysis: a place $p \in \hat{P}$ if there exists transitions $t' \in T', t'' \in T''$ such that there exists a path from t' to p that does not pass through t'' and there exists a path from p to t'' that does not pass through t'.

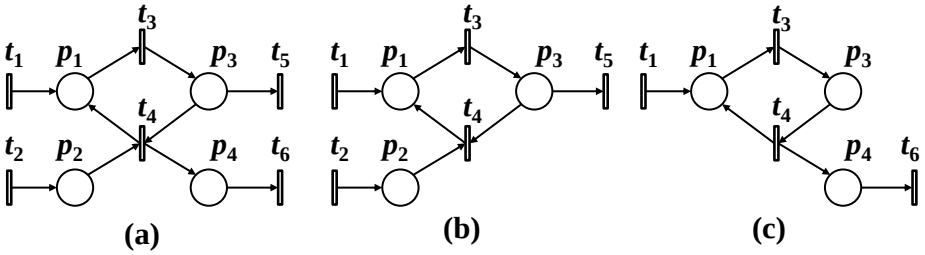

Figure 3: (a) A Petri net for Example 5, and (b), (c) its two flow subnets.

Definition 8 *Given a Petri net N and two non-empty sets of transitions $T', T'' \subseteq T$, let $\hat{N}$ be the (T', T'')-flow subnet. Transition set T'' is* dependent *on transition set T' if for any marking $\hat{M} \in \mathbb{N}^{|P|}$ defined on $\hat{N}$ there exists an integer $K_{\hat{M}} \in \mathbb{N}$ that depends on $\hat{M}$ such that*

$$[\sigma \in L(\hat{N}, \hat{M})] \ \wedge \ [\sigma(T') = 0] \quad \Rightarrow \quad \sum_{t \in T''} \mathbf{y}_\sigma(t) \leq K_{\hat{M}}.$$

We use $T'' \rightarrow_D T'$ (resp., $T'' \nrightarrow_D T'$) to denote that T'' is dependent (resp., not dependent) on T'.

The dependency of transitions is a structural property that does not require that the plant LPN be bounded. This notion will be used to establish a quantitative relation between the firing of transitions in T' and in T''. By Definition 8, if T'' is dependent on T', then in the flow subnet of (T', T''), transitions in T'' cannot fire infinitely often without firing transitions in T', regardless of the initial marking. In such a case, the firings of transitions in T'' can be used to estimate the firings of transitions in T'.

Example 5 *Consider the net in Figure 3(a). Let $T_1 = \{t_1, t_2\}$ and $T_2 = \{t_5\}$. Although there are infinite anchored paths in $\Pi(T_1, T_2)$ (e.g., $t_1 p_1 t_3 p_3 (t_4 p_1 t_3 p_3)^n t_5$ for all $n \geq 0$), the flow subnet of (T_1, T_2) is finite, as depicted in Figure 3(b). Since for any marking $\hat{M}$ in the flow subnet, T_2 can fire no more than $\hat{M}(p_2)$ times without firing transitions in T_1, $T_2 \rightarrow_D T_1$ holds.*

On the other hand, let $T_3 = \{t_1\}$ and $T_4 = \{t_6\}$. The flow subnet of (T_3, T_4) is shown in Figure 3(c). Clearly $T_3 \nrightarrow_D T_4$ since in the flow subnet at marking $M = [1, 0, 0]^T$, transition t_6 can fire infinite times without firing t_1.

In the following we present two propositions to characterize transition dependency. The first provides us a way to determine if T'' is dependent on T', while the second one characterize how the property of dependency extends to subsets/supersets.

Proposition 3 *Given a net N and two sets of transitions $T', T'' \subseteq T$, let $\hat{N}$ be the (T', T'')-flow subnet and $\hat{n}$ be the number of transitions in $\hat{N}$. Transition set T'' is dependent on T' if and only if the following integer system of constraints is not feasible.*

$$\begin{cases} C_{\hat{N}} \cdot \mathbf{y} \geq \mathbf{0} \\ \sum_{t' \in T'} y(t') = 0 \\ \sum_{t'' \in T''} y(t'') \geq 1 \\ \mathbf{y} \in \mathbb{N}^{\hat{n}}. \end{cases} \tag{6}$$

Proof: (If) Suppose that Eq. (6) admits at least one solution $\mathbf{y} \gneq \mathbf{0}$ (clearly $\mathbf{y} = \mathbf{0}$ is not a solution), which implies that $\mathbf{y}$ contains some transitions in T'' but no transition in T'. This indicates that from any marking $\hat{M}$ large enough to enable a firing sequence σ associated to $\mathbf{y}$, the same sequence can repeatedly fire at $\hat{M}$ because the first constraint implies that σ is repetitive. This leads to an arbitrary large number of firings of transitions in T'' without firing any of T'. Hence T'' is not dependent on T'.

(Only if) Suppose that T'' is not dependent on T'. It means that from some marking M there exists a repetitive sequence that contains transitions in T'' but no transition in T'. The firing vector of this sequence is an admissible solution of Eq. (6). □

Proposition 4 *Given a net N and two sets of transitions T', T'', the following conditions hold:*

1. *$T'' \nrightarrow_D T'$ implies that for all $T''_+ \supseteq T''$, $T''_+ \nrightarrow_D T'$ holds;*
2. *$T'' \nrightarrow_D T'$ implies that for all non-empty $T'_- \subseteq T'$, $T'' \nrightarrow_D T'_-$ holds;*
3. *$T'' \rightarrow_D T'$ implies that for all non-empty $T''_- \subseteq T''$, $T''_- \rightarrow_D T'$ holds;*
4. *$T'' \rightarrow_D T'$ implies that for all $T'_+ \supseteq T'$, $T'' \rightarrow_D T'_+$ holds;*

where $\nrightarrow_D$ means "not dependent on".

Proof: (Item 1) Suppose that $T'' \nrightarrow_D T'$. Let $\hat{N}$ be the (T', T'')-flow subnet. By Proposition 3, there exists a marking $\hat{M}$ at which a repetitive firing sequence σ, which contains some transitions in T'' and does not contain any transition in T', can fire. Notice that to add new transitions to T'' will not add any new place p to $\hat{N}$ such that p is an input place of a transition t in $\hat{N}$ (otherwise there is a path from T' to p to t to T'', which indicates that p is already in $\hat{N}$). As a result, σ is also a repetitive sequence at marking $[\hat{M}, \mathbf{0}]$ in the flow net of (T', T''_+)-flow subnet, which implies that at $[\hat{M}, \mathbf{0}]$ transitions in T''_+ can fire infinitely often without firing any transition in T'. Hence $T''_+ \nrightarrow_D T'$.

(Item 2) Suppose $T'' \nrightarrow_D T'$. In the (T', T'')-flow subnet $\hat{N}$, there exists a marking $\hat{M}$ at which a repetitive firing sequence σ, which contains some transitions in T'' and does not contain any transition in T', can fire. Then, let $\tilde{N} = (\tilde{P}, \tilde{T}, \tilde{Pre}, \tilde{Post})$ be the flow net of (T'_-, T''). Net $\tilde{N}$ can be obtained from $\hat{N}$ by removing all places that are not on a path from $T' \setminus T'_-$ to T''. Let $\tilde{M} = \hat{M}_{\uparrow \tilde{P}}$, and $\tilde{\sigma} = \sigma_{\uparrow \tilde{T}}$. Since for any transition t in $\tilde{N}$ the set of its input places is either unchanged or reduced from that in $\hat{N}$, $\tilde{\sigma}$ can repetitively fire at $\tilde{M}$ in $\tilde{N}$. Since $\tilde{\sigma}$ contains transitions in T'' but no transitions in T', $T'' \nrightarrow_D T'_-$ holds.

(Item 3) It is the contrapositive of Item 1.

(Item 4) It is the contrapositive of Item 2. □

5 Monitor-based Supervisor Design

As discussed at the end of Section III, a supervisor forbids a sequence σ if there exists any illegal marking M consistent with $w = \ell(\sigma)$. Given an observation $w \in L_o(G)$, the set of its *consistent markings* is denoted as

$$\mathcal{C}(w) = \{M \in R(N, M_0) \mid (\exists \sigma : \ell(\sigma) = w)\ M_0[\sigma\rangle M\}$$

We denote by $Q(w)$ the maximal increase of quantity $\mathbf{w}^T \cdot M$ from the initial token count (i.e., $\mathbf{w}^T \cdot M_0$) among all markings that are consistent with w, i.e.,

$$\begin{aligned} Q(w) &= \max_{M \in \mathcal{C}(w)} \{\mathbf{w}^T \cdot M\} - \mathbf{w}^T \cdot M_0 \\ &= \max\{\mathbf{w}^T \cdot M \mid (\exists \sigma \in \ell^{-1}(w)) M_0[\sigma\rangle M\} - \mathbf{w}^T \cdot M_0. \end{aligned} \tag{7}$$

Hence, after observing w, an event e must be disabled if $Q(we) > k - \mathbf{w}^T \cdot M_0$.

A control policy based on the precise value of $Q(w)$ would be maximally permissive under the considered partial observation framework. However, there is no general method to compute $Q(w)$ except computing $\mathcal{C}(w)$ at each step by reachability analysis. As a result, in practice it is useful to determine an upper bound for $Q(w)$ which is easily computable to efficiently design a (possibly less permissive) supervisor.

Notice that

$$\begin{aligned} Q(w) &= \max\{\mathbf{w}^T \cdot M \mid (\exists \sigma \in \ell^{-1}(w)) M_0[\sigma\rangle M\} - \mathbf{w}^T \cdot M_0 \\ &\leq \max\{\mathbf{w}^T \cdot M \mid (\exists \mathbf{y}, z_w(e) = \sum_{t \in T_e} y(t)) \\ &\qquad\qquad M = M_0 + C \cdot \mathbf{y}\} - \mathbf{w}^T \cdot M_0 \\ &\leq \sum_{e \in E} [\mathbf{z}_w(e) \cdot \max_{t \in T_e} \eta(t)]. \end{aligned} \tag{8}$$

The right-hand-side of the first inequality in Eq. (8) provides an upper bound for $Q(w)$ using the state equation. However, to verify such a condition is also computationally heavy since for each observed event an *integer linear programming problem* (ILPP) has to be solved. On the other hand, the right-hand-side of the second inequality in Eq. (8), as discussed in Section 3, provides a relaxed upper bound for $Q(w)$ by assuming that each event e in w has been produced by the firing of transitions in T_e with the largest influence. Although this condition can be easily verified, a supervisor designed using this condition may be too restrictive, since such an upper bound is too relaxed. Hence, in the rest of this section we propose a tighter upper bound based on transition dependency. This approach reaches a good trade-off between computational load and permissiveness.

To explain the intuition of the tighter upper bound for $Q(w)$ we design, let us first introduce the following function $Q_e : L_o(G) \to \mathbb{Z}$:

$$Q_e(w) = \max_{\sigma \in \ell^{-1}(w)} \sum_{t \in T_e} \mathbf{y}_\sigma(t) \cdot \eta(t). \tag{9}$$

The value of $Q_e(w)$ for a given w is the maximal value of the influence of transitions belonging to T_e in sequences σ that are consistent with w. Clearly, $Q(w) \leq \sum_{e \in E} Q_e(w) \leq \sum_{e \in E} [\mathbf{z}_w(e) \cdot \max_{t \in T_e} \eta(t)$ holds. Given a transition set $T' \in T$, we denote by $\eta(T')$ the maximal influence of transitions in T', i.e., $\eta(T') = \max_{t \in T'} \eta(t)$. Now, according to the influence of transitions in T_e, the set T_e is partitioned into

$$T_e = T_{e,1} \cup T_{e,2} \cup \cdots \cup T_{e,m} \tag{10}$$

such that for all $t_j, t_{j'} \in T_{e,i}$, $\eta(t_j) = \eta(t_{j'})$ holds, and $\eta(T_{e,i}) > \eta(T_{e,j})$ if $i < j$. For a set $T_{e,i}$, suppose that there exists a different event $\tilde{e}_i \in E$ such that set $T_{\tilde{e}_i}$ is dependent on set $T_{e,i}$. In such a case, if we observe event e for r_1 times, and by observing $\tilde{e}_i$'s we are able to infer that transitions in each $T_{e,i}$ fire r_i times for each $i \in \{2, \ldots, n\}$, respectively, then we obtain the following condition:

$$Q_e(w) \leq (r_1 - \sum_{i=2}^{m} r_i) \cdot \eta(T_{e,1}) + \sum_{i=2}^{m} (r_i \cdot \eta(T_{e,i})). \tag{11}$$

The right-hand-side of Eq. (11) is an upper bound of $Q_e(w)$. In Section 5.2 we will formally define a function $U_e : L_o(G) \to \mathbb{N}$ that is in the form of the right-hand-side of Eq. (11). Moreover, the value of such a function U_e is always in the interval $[Q_e(w), \mathbf{z}_w(e) \cdot \max_{t \in T_e} \eta(t)]$ for any $w \in L_o(G)$ and is easy to compute using transition dependency. Therefore, the quantity of $\sum_{e \in E} U_e(w)$ will be used as tighter upper bound for $Q(w)$ to design a supervisor in the sequel of this paper.

By the discussion above, our target is for each $T_{e,i}$ to find an event $\tilde{e}_i$ such that $T_{\tilde{e}_i}$ is dependent on $T_{e,i}$, and establish a quantitative relation between the observation of $\tilde{e}_i$ and $T_{e,i}$. Recall that to verify $T_{\tilde{e}} \to_D T_{e,i}$ for event $\tilde{e}$ and set $T_{e,i}$ requires solving Eq. (6). We need to test transition dependency for each set $T_{e,i}$ and each set $T_{\tilde{e}}$, hence, Eq. (6) needs to be solved at most $\sum_{e \in E} |T_e| \cdot (|E| - 1) = |T| \cdot (|E| - 1)$ times.

On the other hand, to establish the quantitative relation between the number of firings of transitions in $T_{e,i}$ and $T_{\tilde{e}}$, we need to determine a function $\alpha_{(e,i,\tilde{e})} : \mathbb{N} \to \mathbb{N}$:

$$\sigma \in L(N, M_0), \sigma(T_{\tilde{e}}) = K \quad \Rightarrow \quad \sigma(T_{e,i}) = \alpha_{(e,i,\tilde{e})}(K). \tag{12}$$

This function $\alpha_{(e,i,\tilde{e})}$ indicates that by observing events $\tilde{e}_i$ for K times we can conclude that transitions in $T_{e,i}$ must have fired at least $\alpha_{(e,i,\tilde{e})}(K)$ times. However, Eq. (6) in Proposition 3 does not provide a quantitative relation between the number of occurrences of event $\tilde{e}$ and the number of firings of transitions in $T_{e,i}$. In the following, we show how to compute such a function $\alpha_{(e,i,\tilde{e})}$ using the notion of *Hilbert basis*.

5.1 Hilbert Basis

We first introduce the notion of *Hilbert basis*. In plain words, the Hilbert basis of a linear inequalities system $\mathbf{A} \cdot \mathbf{x} \geq \mathbf{0}$ is a minimal set of integer vectors H such that every feasible solution of the system is a conical combination of the vectors in H with integer coefficients.

Definition 9 (Hilbert Basis) *[10] Given an integer matrix $A \in \mathbb{Z}^{m \times n}$, let $J \subseteq \mathbb{N}^n$ be the set of all nonnegative integer vector solutions of a system of linear inequalities $\mathbf{A} \cdot \mathbf{x} \geq \mathbf{0}$, i.e., $J = \{\mathbf{x} \in \mathbb{N}^n \mid \mathbf{A} \cdot \mathbf{x} \geq \mathbf{0}, \mathbf{x} \geq \mathbf{0}\}$. Set $H \subset J$ is called the* Hilbert basis *of $\mathbf{A} \cdot \mathbf{x} \geq \mathbf{0}$ if H is a minimal set such that every element in J is an integer conical combination (i.e., a linear combination with nonnegative integer coefficients) of vectors in H.*

It has been proved that the Hilbert basis of any system of linear inequalities $\mathbf{A} \cdot \mathbf{x} \geq \mathbf{0} \wedge \mathbf{x} \geq \mathbf{0}$ exists, is finite, and is unique [10]. The complexity of computing the Hilbert basis of a given system is exponential in the number of constraints in it [14], which is similar to the complexity of solving ILPPs that are widely used when solving the state-equation of Petri nets. So far, several algorithms (see [10, 27]) and software tools (such as *Normaliz* [5]) have been developed for computing a Hilbert basis. Now, we present a theorem that provides a quantitative relationship between two dependent transition sets based on Hilbert basis, after the following lemma.

Lemma 1 *For two sequences of nonnegative integers $y_i \in \mathbb{N}, z_i \in \mathbb{N} \setminus \{0\}$ satisfying $y_i/z_i \geq y_{i+1}/z_{i+1}$ for $i \in \{1, \ldots, n-1\}$, the following inequality holds for any $\mathbf{x} = [x_1, \ldots, x_n] \in \mathbb{N}^n \setminus \{\mathbf{0}\}$:*

$$\frac{y_1}{z_1} \geq \frac{\sum_{i=1}^{n} x_i \cdot y_i}{\sum_{i=1}^{n} x_i \cdot z_i}.$$

Proof: Since all x_i, y_i, z_i are nonnegative, $y_1/z_1 \geq y_i/z_i$ implies that $y_1 \cdot (x_i \cdot z_i) \geq z_1 \cdot (x_i \cdot y_i)$ for $i \in \{1, \ldots, n\}$. Therefore by summing them up from $i = 1$ to n we have:

$$y_1 \cdot (x_1 \cdot z_1 + \cdots + x_n \cdot z_n) \geq z_1 \cdot (x_1 \cdot y_1 + \cdots + x_n \cdot y_n).$$

and hence $y_1/z_1 \geq (x_1 \cdot y_1 + \cdots + x_n \cdot y_n)/(x_1 \cdot z_1 + \cdots + x_n \cdot z_n)$ holds. □

Theorem 2 *Given two sets of transitions T' and T'' such that T'' is dependent on T', for a zero-marked subnet $\langle N', \mathbf{0}\rangle$ where N' is the (T', T'')-flow subnet, the following condition holds:*

$$\begin{aligned} &\max_{\sigma \in L(N', \mathbf{0}), \mathbf{y}_\sigma(T'') > 0} \{ \sum_{t \in T''} \mathbf{y}_\sigma(t) / \sum_{t \in T'} \mathbf{y}_\sigma(t) \} \\ &\leq \max_{C_{N'} \cdot \mathbf{y} \geq \mathbf{0}, \mathbf{y} \geq \mathbf{0}, \mathbf{y}(T'') > 0} \{ \sum_{t \in T''} \mathbf{y}(t) / \sum_{t \in T'} \mathbf{y}(t) \} \\ &\leq \max_{\mathbf{y} \in H} \{ \sum_{t \in T''} \mathbf{y}(t) / \sum_{t \in T'} \mathbf{y}(t) \} \end{aligned} \tag{13}$$

where H is the set of Hilbert basis of $C_{N'} \cdot \mathbf{y} \geq \mathbf{0}$.

Proof: Since the net is initially zero-marked and T'' is dependent on T', $\mathbf{y}_\sigma(T'') > 0$ and $\mathbf{y}(T') > 0$ imply $\mathbf{y}_\sigma(T') > 0$ and $\mathbf{y}(T'') > 0$, respectively. Hence, the denominators must be non-zero, and the maximal value of $\sum_{t \in T''} \mathbf{y}(t) / \sum_{t \in T'} \mathbf{y}(t)$ for $\mathbf{y} \in H$ is finite. The first $\leq$ is trivial, and we now prove the second $\leq$. Let J be the set $J = \{\mathbf{y} \in \mathbb{N}^{|T'|} \mid C' \cdot \mathbf{y} \geq \mathbf{0}\}$ and H be its Hilbert basis. For any $\mathbf{y} \in J$ there exists a non-zero integer vector $\mathbf{x} \in \mathbb{N}^{|H|}$ such that $\mathbf{y} = \sum_{i=1}^{|H|} \mathbf{x}(i) \cdot \mathbf{y}_i$, where $\mathbf{y}_i \in H$ is the i-th component of the Hilbert basis. Hence we have:

$$\frac{\sum_{t \in T''} \mathbf{y}(t)}{\sum_{t \in T')} \mathbf{y}(t)} = \frac{\sum_{i=1}^{|H|} \mathbf{x}(i) \cdot \sum_{t \in T''} \mathbf{y}_i(t)}{\sum_{i=1}^{|H|} \mathbf{x}(i) \cdot \sum_{t \in T'} \mathbf{y}_i(t)} \leq \max_{\mathbf{y} \in H} \frac{\sum_{t \in T''} \mathbf{y}(t)}{\sum_{t \in T'} \mathbf{y}(t)}$$

where the $\leq$ step is by Lemma 1. □

When a set T' is dependent on a set T'' and the $(\hat{T}_e, T_{\tilde{e}})$-flow subnet is initially zero-marked, Theorem 2 provides an upper bound for $\sum_{t \in T''} \mathbf{y}_\sigma(t) / \sum_{t \in T'} \mathbf{y}_\sigma(t)$ for all $\sigma \in L(N', \mathbf{0}), \mathbf{y}_\sigma(T'') > 0$. Then, for T', T'' whose transitions have different labels, we establish a quantitative relation between the firing of transitions in them by the following proposition.

Proposition 5 *Given an LPN G, let $\hat{T}_{e,i} \subset T_e$ be a set of transitions labeled by event e according to Eq.* (10)*. Let $\tilde{e} \in E$ be an event such that $\tilde{e} \neq e$ and $T_{\tilde{e}}$ is dependent on $\hat{T}_{e,i}$. Then, in net $\langle N', \mathbf{0}\rangle$ where N' is the $(T_{e,i}, T_{\tilde{e}})$-flow subnet, for any $\sigma \in L(N', \mathbf{0})$ such that $\mathbf{y}_\sigma(T'') > 0$, it holds $\alpha_{(e,i,\tilde{e})}(z_w(\tilde{e})) = \lceil z_w(\tilde{e})/s \rceil$, i.e.:*

$$(\sigma(T_{\tilde{e}}) = z_w(\tilde{e})) \Rightarrow (\sigma(\hat{T}_{e,i}) \geq \lceil z_w(\tilde{e})/s \rceil) \tag{14}$$

where $s = \max_{\mathbf{y} \in H} \frac{\sum_{t \in T_{\tilde{e}}} \mathbf{y}(t)}{\sum_{t \in \hat{T}_{e,i}} \mathbf{y}(t)}$, and "$\lceil \cdot \rceil$" is the ceiling operator *which returns the minimal integer that is not smaller than $(\cdot)$.*

Proof: By letting $T' = \hat{T}_{e,i}$ and $T'' = T_{\tilde{e}}$ in Theorem 2, the following condition holds:

$$\begin{aligned} &\max_{\sigma \in L(N', \mathbf{0}), \mathbf{y} \geq \mathbf{0}, \mathbf{y}_\sigma(T'') > 0} \{ \sum_{t \in T_{\tilde{e}}} \mathbf{y}_\sigma(t) / \sum_{t \in \hat{T}_{e,i}} \mathbf{y}_\sigma(t) \} \\ &\leq \max_{\mathbf{y} \in H} \{ \sum_{\hat{t} \in T_{\tilde{e}}} \mathbf{y}(t) / \sum_{t \in \hat{T}_{e,i}} \mathbf{y}(t) \} \end{aligned} \tag{15}$$

where H is the set of Hilbert basis of $C_{N'} \cdot \mathbf{y} \geq \mathbf{0}$. Then the statement trivially holds: otherwise, the firing vector $\mathbf{y}_\sigma$ of a firing sequence σ with $\sigma(T_{\tilde{e}}) = z_w(\tilde{e})$ and $\sigma(\hat{T}_{e,i}) \geq \lceil z_w(\tilde{e})/s \rceil$ is a solution of $C_{N'} \cdot \mathbf{y} \geq \mathbf{0}$ but cannot be obtained by integer conical combination of elements in H, which is a contradiction. □

It is worth noting that if the subnet N' is not zero-marked, Theorem 2 holds by replacing the inequality $C_{N'} \cdot \mathbf{y} \geq \mathbf{0}$ with $M_0 + C_{N'} \cdot \mathbf{y} \geq \mathbf{0}$, and hence the results in Proposition 5 are still applicable in these cases. However, the resulted control policy will be more conservative, since the firing vector in $\mathbf{y} \in H$ with the maximal value of $\sum_{t \in T_{\tilde{e}}} \mathbf{y}(t) / \sum_{t \in \hat{T}_e} \mathbf{y}(t)$ may not satisfy $C_{N'} \cdot \mathbf{y} \geq \mathbf{0}$. Such a situation requires to be handled with a special care and this is an issue that will be explored in our future work. However, we point out that in many Petri net models of real systems tokens are initially distributed only in some *idle* places while the working zone (where those transitions with $\eta \neq 0$ are associated to) is zero-marked. Proposition 5 (and results in the rest of this section) are well applicable to these cases.

5.2 Estimation of Influence

Now we define the *event estimation function* $U_e : E^* \to \mathbb{N}$ that provides an upper bound for $Q_e(w)$. The physical interpretation of Eq. (16) is similar to Eq. (11), and the $\max$ operator is used for finding the maximal number of firings of $T_{e,i}$ according to all its dependent events.

Definition 10 (Event Estimation Function) *Given an LPN $G = (N, E, \ell, M_0)$ and event $e \in E$, the* estimation function $U_e : E^* \to \mathbb{N}$ *(of event e) is defined as:*

$$\begin{aligned} U_e(w) =& z_w(e) \cdot \eta(T_e) \\ & - [\sum_{i=2}^{m} \max_{\tilde{e} \in E} \alpha_{(e,i,\tilde{e})}(z_w(\tilde{e}))] \cdot \eta(T_e) \\ & + \sum_{i=2}^{m} [\max_{\tilde{e} \in E} \alpha_{(e,i,\tilde{e})}(z_w(\tilde{e})) \eta(T_{e,i})]. \end{aligned} \tag{16}$$

where $T_e = T_{e,1} \cup T_{e,2} \cup \cdots \cup T_{e,m}$ such that for all $t, t' \in T_{e,i}$, $\eta(t) = \eta(t')$, and $\eta(T_{e,i}) > \eta(T_{e,j})$ if $i < j$.

As mentioned at the beginning of this section (see Eq. (12)), functions $\alpha_{(e,i,\tilde{e})}$ provides a quantitative relation between the observation of $\tilde{e}_i$ and the firing of transitions in $T_{e,i}$ based on transition dependency: by observing events $\tilde{e}$ for K times, we conclude that transitions in $T_{e,i}$ must have fired at least $\alpha_{(e,i,\tilde{e})}(K)$ times. Note that for $\tilde{e}$ and $T_{e,i}$ such that $T_{\tilde{e}}$ is not dependent on $T_{e,i}$, term $\alpha_{(e,i,\tilde{e})}$ in Eq. (16) is treated as zero, i.e., $\alpha_{(e,i,\tilde{e})}(K) = 0$ for all $K \in \mathbb{N}$, and hence can be practically neglected.

Proposition 6 *Given an observation w, for all $\sigma \in L(N, M_0)$ such that $\ell(\sigma) = w$, it holds:*

$$Q_e(w) \leq U_e(w) \leq \mathbf{z}_w(e) \cdot \max_{t \in T_e} \eta(t). \tag{17}$$

Proof: The first $\leq$ is trivial since $U_e(w)$ is the right-hand-side of Eq. (11). For the second $\leq$, since $T_{e,i}$'s are in the descending order of η and $\alpha_{(\cdot)}$ is always positive, the right-hand-side of Eq. (16) is necessarily less than or equal to $\mathbf{z}_w(e) \cdot \eta(T_e)$. □

By Proposition 6, $U_e(w)$ is an upper bound of $Q_e(w)$ and can be to design a control policy. Now we propose an algorithm to compute the estimation function $U_e : E^* \to \mathbb{Z}$ in the form of Eq. (16) for event

Algorithm 1 Computation of estimation function of an event

Input: An LPN $G = (N, M_0, E, \ell)$, a GMEC $(\mathbf{w}, k)$ satisfying Assumptions 1 and 2, and an event $e \in E$

Output: Estimation function $U_e : E^* \to \mathbb{Z}$

1: Let $T_e = T_{e,1} \cup T_{e,2} \cup \cdots \cup T_{e,m}$ such that for all $t, t' \in T_{e,i}$, $\eta(t) = \eta(t')$, and $\eta(T_{e,i}) > \eta(T_{e,j})$ for $i < j$;
2: let $\tilde{E} = \emptyset, i = 2$;
3: **while** $i \leq m$, **do**
4: **for all** $\tilde{e} \in E, \tilde{e} \neq e$, **do**
5: **if** $T_{\tilde{e}} \to_D T_{e,i}$, **then**
6: let $\tilde{E} = \tilde{E} \cup \{\tilde{e}\}$;
7: **end if**
8: **end for**
9: **for all** $\tilde{e} \in \tilde{E}$, **do**
10: compute $\alpha_{(e,i,\tilde{e})}$ according to Proposition 5;
11: **end for**
12: let $i = i + 1, \tilde{E} = \emptyset$;
13: **end while**
14: Output U_e according to Eq. (16).

$e \in E$. Since U_e is in the form of Eq. (16), for any observation $w \in L_o(G)$ the value of $U_e(w)$ can be easily computed (an example can be found in Section 6).

In Step 1 of Algorithm 1, according to the influence of transitions in T_e, the set T_e is partitioned into $T_{e,1} \cup \cdots \cup T_{e,m}$ according to the influences of transitions. Step 2 initializes the set $\tilde{E}$ to temporarily record event $\tilde{e}$'s depending on each $T_{e,i}$. The external loop (Steps 3 to 13) finds a set of events $\tilde{e} \in E, \tilde{e} \neq e$ such that $T_{\tilde{e}}$ is dependent on set $T_{e,i}$, and puts all such events $\tilde{e}$ in set $\tilde{E}$ (Steps 4 to 8). In Steps 9 to 10, function $\alpha_{(e,i,\tilde{e})}$ is computed such that $z_w(\tilde{e}) = r$ implies for all $\sigma \in L(N, M_0), \ell(\sigma) = w$, $\mathbf{y}_\sigma(T_{e,i}) \leq \alpha_{(e,i,\tilde{e})}(r)$ holds. The estimation function U_e in the form of Eq. (16) is outputted in Step 14. Finally, a control policy ζ can be designed by the following theorem.

Theorem 3 *Given an LPN $G = (N, E, \ell, M_0)$ and a nondeterministic GMEC $(\mathbf{w}, k)$, the following monitor function*

$$\zeta(w) = k - (\mathbf{w}^T \cdot M_0 + \sum_{e \in E} U_e(w)) \tag{18}$$

satisfies: $\zeta(w) \geq 0 \Rightarrow \forall \sigma \in \ell^{-1}(w), M_0[\sigma\rangle M \in \mathcal{L}_{(\mathbf{w},k)}$, where $w \in L_o(G)$ is an observation of G.

Proof: According to Proposition 6, $U_e(w) \geq Q_e(w)$ holds, which implies $\sum_{e \in E} U_e(w) \geq \sum_{e \in E} Q_e(w) \geq Q(w)$. Hence, for an observation $w \in L_o(G)$ such that $\zeta(w) \geq 0$, for all sequences $\sigma \in \ell^{-1}(w)$ such that $M_0[\sigma\rangle M$, $\mathbf{w}^T \cdot M \leq \mathbf{w}^T \cdot M_0 + Q(w) \leq \mathbf{w}^T \cdot M_0 + \sum_{e \in E} U_e(w) = k - \zeta(w) \leq k$ holds. Therefore, the statement is true. □

5.3 Permissiveness of Monitor-based Supervisors

Our approach estimates the token count of a given nondeterministic GMEC by using transition dependency. In other words, our supervisor first performs a worst-case estimation and then improves it from other observed

events thereafter. On the other hand, a marking-estimation-based supervisor makes control decisions by processing all information from the history of the observation so far. Hence, the control decision made by of our supervisor may be overrestrictive with respect to marking-estimation-based approaches in some cases due to two reasons: (i) some transitions with nonzero influences may not have dependent events, which implies that the numbers of their firings cannot be estimated; (ii) the number of firings of a transition can be estimated only after its dependent transitions fire, i.e., with a "delay". However, our monitor-based supervisor requires a significant lower online computational effort than a marking-estimation-based supervisor. For example, let us again consider the LPN in Figure 1 and the legal set $\mathcal{L}_{(\mathbf{w},k)} = \{M \mid M(p_3) \leq 2\}$. This net has 600 reachable markings. By using a monitor function $\zeta(w) = 2 - (z_w(b) - \lceil z_w(f)/2 \rceil) + z_w(d)$, the controlled plant can reach 140 markings such that both workflows can operate normally. Moreover, the control policy is quite simple. On the other hand, a maximally permissive estimation-based supervisor permits 570 markings in this example. To enforce such a estimation-based supervisor requires to compute a very large *observer automaton* [8] that has 1665 states among which the largest state contains 256 plant markings.

Furthermore, we point out that in some particular cases a monitor-based supervisor designed by our method is maximally permissive. For example, a monitor-based supervisor has the same permissiveness as an estimation-based supervisor if a plant LPN satisfies the following conditions C1 and C2:

C1) All transitions $t \in T$ with $\eta(t) > 0$ are uniquely labeled.

C2) For all $t', t'' \in T$ with $\ell(t') = \ell(t'')$ and $\eta(t'), \eta(t'') \leq 0$, $\eta(t') + \eta(t'') \neq 0$:

- C2.1) The *downstream unobservable subnet* of t', t'', denoted as $N' = (P', T', Pre', Post')$ and $N'' = (P'', T'', Pre'', Post'')$, respectively, are acyclic and disjoint. The downstream unobservable subnet of a transition $t \in T$ is the subnet of N_{uo} by removing all places/transitions x such that in N_{uo} there is no path from t to x.
- C2.2) ${}^\bullet(P') \setminus T' = \{t'\}$ and ${}^\bullet(P'') \setminus T'' = \{t''\}$, i.e., t' and t'' are the unique source transitions of N' and N'', respectively.
- C2.3) $(P')^\bullet \setminus T' = \{\hat{t}'\}$, $(P'')^\bullet \setminus T'' = \{\hat{t}''\}$, and $\hat{t}', \hat{t}''$ are uniquely labeled. In other words, $\hat{t}'$ and $\hat{t}''$ are the unique sink transitions of N' and N'', respectively.

Conditions C1 and C2 depend on both the net structure of G and GMEC $(\mathbf{w}, k)$. The two conditions can both be verified by inspecting the net structure and the GMEC. Condition C1 means that all increases of the token count can be exactly detected by a monitor-based supervisor immediately. On the other hand, Condition C2 means that the firings of any two transitions t' and t'' that look alike and with different negative influences can be exactly distinguished by observing the next firing of their downstream transitions, respectively.

Proposition 7 *Given an LPN G and a GMEC $(\mathbf{w}, k)$ that satisfies Conditions C1 and C2, the monitor-based supervisor designed by Algorithm 1 has the same permissiveness as an estimation-based supervisor.*

Proof: Condition C1 means that both our monitor-based supervisor and an estimation-based supervisor can precisely recognize any increase of the token count of $(\mathbf{w}, k)$ with no delay. On the other hand, for the decrease of the token count, Condition C2.1, C2.2, and C2.3 jointly indicate that for any transitions t' and t'' that look alike and with different negative influences, the firing of t' and that of t'' can be distinguished by observing the firing of $\hat{t}'$ and $\hat{t}''$. This can be proved with the following argument.

Since the only unobservable transitions in N' are t' and $\hat{t}'$, both marking-estimation-based supervisor and our monitor-based supervisor will be able to infer that t' has fired only when observing $\ell(t')$. Since the flow-subnet N' from $\{t'\}$ to $\{\hat{t}'\}$ is acyclic, for any firing vector $\mathbf{y} \in H$ there exists a firing sequence $\sigma \in L(N', \mathbf{0})$ whose firing vector is $\mathbf{y}_\sigma = \mathbf{y}$. Hence, in Eq. (13) the equality holds, i.e.,

$$\max_{\sigma \in L(N', \mathbf{0}), \mathbf{y}_\sigma(\hat{t}') > 0} \{\mathbf{y}_\sigma(\hat{t}')/\mathbf{y}_\sigma(t')\} = \max_{\mathbf{y} \in H} \{\mathbf{y}(\hat{t}')/\mathbf{y}(t')\}.$$

The left-hand side and the right-hand side of the equality above are the estimations made by an marking-estimation-based supervisor and by our monitor-based supervisor, respectively. This argument also holds for transitions t'' and $\hat{t}''$. Therefore, both supervisors always have the same knowledge of the token count of the GMEC and hence have the same control law. □

5.4 Monitor-based Supervisor Design

In the following we propose a control policy based on the previously presented results. Algorithm 2 consists in an offline stage (Steps 1–3) and an online one (Steps 4–15). In the offline stage, Step 2 calls Algorithm 1 to compute the estimation function U_e for each event e in E. Then, in the online stage, the supervisor enters in listening mode (Steps 4–15) where for any observation w, it computes the value of $\zeta(we)$ for all controllable events e according to Eq. (18) and disables those events e whose occurrence may violate the specification (i.e., event e is disabled if $\zeta(we) < 0$).

Algorithm 2 Monitor-based supervisor design

Input: An LPN $G = (N, M_0, E, \ell)$, a GMEC $(\mathbf{w}, k)$ that satisfies Assumptions 1 and 2

Offline:

1: **for all** $e \in E$, **do**
2: Call Algorithm 1 to compute $U_e : E^* \to \mathbb{Z}$ for all e such that there exists transition $t \in T_e$ with $\eta(t) \neq 0$;
3: **end for**

Online:

4: let $w = \varepsilon$;
5: let $Ctrl = E$;
6: **for all** $e \in E_c$, **do**
7: compute $\zeta(we)$ according to Eq. (18);
8: **if** $\zeta(we) < 0$, **then**
9: $Ctrl = Ctrl \setminus \{e\}$;
10: **end if**
11: **end for**
12: Disable all $t \in T$ such that $\ell(t) \notin Ctrl$;
13: Wait until an event e is observed;
14: let $w = we$;
15: Goto Step 5;

It is worth noticing that the monitor function designed in this section does not forbid any uncontrollable transition. A monitor function ζ disables event e after observation w if and only if the quantity of $\zeta(we)$ is

negative. In other words, event e is forbidden if and only if $\zeta(w) \geq 0$ and $\zeta(we) \leq 0$. This implies there exists a transition $t \in T_e$ such that $\eta(t) > 0$, since the quantity of $\zeta(w)$ defined by Eq. (18) is an upper bound of $k - \mathbf{w}^T \cdot M$ when observing w. By Assumption 1, $\ell(t) \in E_c$ holds, i.e., event e is controllable.

At the end of this section we discuss the complexity of our approach and compare it with marking-estimation-based approaches.

- **complexity of supervisor synthesis**: For the offline computation, the `while` loop in Algorithm 1 executes at most $m - 1$ times and the two `for` loop (Steps 4-11) execute $|E| - 1$ times. Hence the computational complexity of Algorithm 1 is $(m-1)(|E|-1)$, i.e., $O(|T_e| \cdot |E|)$. The offline stage of Algorithm 2 calls Algorithm 1 $|E|$ times. Hence, Algorithm 2 needs to solve Hilbert basis for at most $\sum_{e \in E}(|T_e|-1)(|E|-1) = (|T|-|E|)(|E|-1)$ times. Since the complexity of solving a Hilbert Basis in system $C_{N'} \cdot \mathbf{y} \geq \mathbf{0}$ is exponential in the number of constraints [14] which is equal to the number of places in a plant net, the complexity of our approach to design a supervisor is $O(|T| \cdot |E| \cdot 2^{|P|})$.

  On the other hand, to design a marking-estimation-based supervisor, an *observer* of a plant LPN needs to be computed offline (otherwise the online computational load would be very high, see the third bullet). Although some state-abstraction techniques [28, 24] may be used, the worst-case complexity of them is $R(N, M_0)$, which implies that the corresponding *observer automaton* [8] has a structural complexity $O(2^{|R(N,M_0)|})$. In general, we have $|P| \ll |R(N, M_0)|$, which indicates that our method is more efficient than marking-estimation-based approaches.

- **structural complexity of a supervisor**: In $U_e(w)$ there are at most $|T_e| \cdot (|E|-1)$ non-zero α's. Since the maximal number of α's in ζ is $\sum_{e \in E} |T_e| \cdot (|E|-1)$, the structural complexity of our supervisor ζ is $O(|T| \cdot |E|)$,

  In contrast, as discussed above, the structural complexity of a marking-estimation-based supervisor is $O(2^{|R(N,M_0)|})$ that is still much higher than that of ours.

- **complexity of online control decision making**: The control decision after observing w is made by computing $\zeta(wt)$ for each transition t in T, the online complexity of our method is $O(|T|^2 \cdot |E|$. This can be swiftly done since $\zeta(wt)$ can be computed by simple arithmetical operations.

  In comparison, to make a control decision online, a marking-estimation-based supervisor needs to computes all consistent markings $\mathcal{C}(we)$ for all events $e \in E$. When the observer is constructed (whose structural complexity is $O(2^{|R(N,M_0)|})$), to compute $\mathcal{C}(we)$ from $\mathcal{C}(w)$ can be done in $O(1)$, which indicates that the complexity of online control decision making is $O(|E|)$. However, if the observer is not computed offline, the supervisor either preforms a brute-force reachability search or solves a series of ILPPs [7] at each step online, which is computationally heavy.

We point out that the Hilbert basis does not depend on the initial marking: hence our method does not depend on the size of the state space and needs not be recomputed as the initial marking changes. On the other hand, as we have mentioned in Section 5.3, our supervisor may not guarantee maximally permissiveness, while a marking-estimation-based approaches can do so. However, our approach reaches good trade-off between computational load and permissiveness.

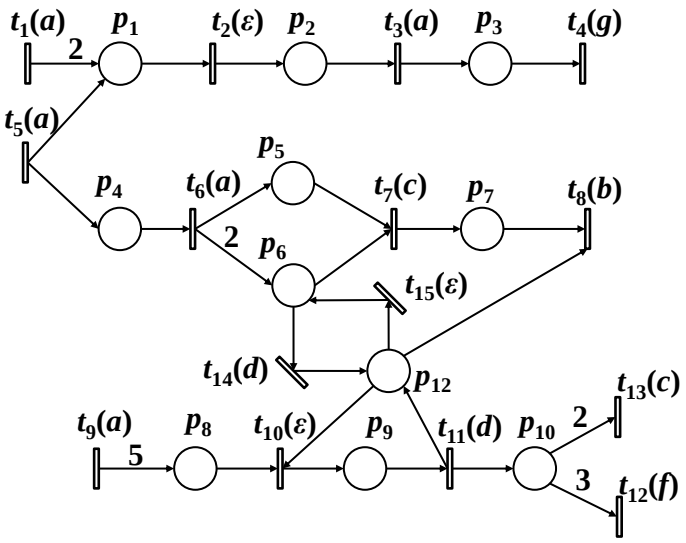

Figure 4: The example for Section 6.

6 An Illustrative Example

Consider the LPN in Figure 4 with $E_c = E$ and a GMEC $(\mathbf{w}, k)$ whose set of legal markings is $\mathcal{L}_{(\mathbf{w},k)} = \{M \mid M(p_1) + M(p_2) \leq 12\}$. This GMEC satisfies Assumptions 1 and 2. We use Algorithm 2 to design a monitor-based supervisor that enforces $(\mathbf{w}, k)$. In this LPN there are five transitions t_1, t_3, t_5, t_6, t_9 labeled with a: $\eta(t_1) = 2, \eta(t_3) = -1, \eta(t_5) = 1, \eta(t_6) = \eta(t_9) = 0$. On the other hand, each of the other events is assigned to a unique transition that has zero influence on $(\mathbf{w}, k)$. Hence $\zeta = 12 - U_a(w)$ where function U_a is computed as follows.

According to the influence of transitions, set T_a is partitioned into four sets: $T_a = T_{a,1} \cup T_{a,2} \cup T_{a,3} \cup T_{a,4}$, where $T_{a,1} = \{t_1\}$ ($\eta(t_1) = 2$), $T_{a,2} = \{t_5\}$ ($\eta(t_5) = 1$), $T_{a,3} = \{t_6, t_9\}$ ($\eta(t_6) = \eta(t_9) = 0$), and $T_{a,4} = \{t_3\}$ ($\eta(t_3) = -1$).

For $T_{a,2} = \{t_5\}$, by calling Algorithm 1 and solving ILPP (6), we understand that event d is not suitable to recognize the firing or $T_{e,2}$, since T_d is not dependent on $T_{a,2}$. On the other hand, event b can be used to recognize the firing of $T_{a,2}$. The Hilbert basis H contains 13 firing vectors, among which vector $\mathbf{y}^*$ representing $1 \cdot t_5 + 1 \cdot t_6 + 1 \cdot t_7 + 1 \cdot t_8 + 1 \cdot t_{14}$ has the maximal quantity of $[y(t_8)/y(t_5)] = 1$. By Propositions 5 and 6 whenever event b is observed $z_w(b)$ times, at least $z_w(b)$ occurrences of event a are generated by $T_{e,2}$, i.e., t_5.

For $T_{a,3} = \{t_6, t_9\}$, by calling Algorithm 1 and solving ILPP (6) we find that both events c and f can be used to recognize the firing of $T_{a,3}$. For event c, the Hilbert basis H contains 90 firing vectors among which vector $\mathbf{y}^*$ representing $2 \cdot t_9 + 10 \cdot t_{10} + 10 \cdot t_{11} + 5 \cdot t_{13}$ has the maximal quantity of $[[y(t_7) + y(t_{13})]/[y(t_6) + y(t_9)]] = 5/2$. For event f, the Hilbert basis H contains 48 firing vectors among which vector $\mathbf{y}^*$ representing $3 \cdot t_9 + 15 \cdot t_{10} + 15 \cdot t_{11} + 5 \cdot t_{12}$ has the maximal quantity of $[y(t_{12})/[y(t_6) + y(t_9)]] = 5/3$. Hence by observing c for $z_w(c)$ times and f for $z_w(f)$ times, at least $\max\{\lceil 2/5 \cdot z_w(c) \rceil, \lceil 3/5 \cdot z_w(f) \rceil\}$ occurrences of event a are generated by $T_{e,3} = \{t_6, t_9\}$ instead of t_1.

Finally, the firing of $T_{a,4} = \{t_3\}$ can be identified by event g (i.e., t_4) in a 1:1 ratio ($|H|$ = 2; the flow-

subnet contains t_3, p_3, t_4 and is not drawn). Hence the estimation function U_a is $U_a(w) = K \cdot 2 + z_w(b) \cdot 1 + z_w(g) \cdot (-1)$ where

$$K = z_w(a) - z_w(b) - \max\{\lceil \frac{2}{5} z_w(c) \rceil, \lceil \frac{3}{5} z_w(f) \rceil\}.$$

Therefore, the monitor function is:

$$\begin{aligned} \zeta(w) = 12 - & 2 \cdot z_w(a) + z_w(b) \\ & + 2 \cdot \max\{\lceil \frac{2}{5} z_w(c) \rceil, \lceil \frac{3}{5} z_w(f) \rceil\} + z_w(g) \end{aligned} \tag{19}$$

7 Conclusion

In this paper, we have introduced deterministic GMECs and propose a method to design a monitor function for structurally deterministic ones. For non deterministic GMECs, we have proposed a method to design monitor-based supervisors. Our method is based on the *transition dependency* and *Hilbert basis* which solely depends on the net structure. A supervisor obtained in our method is robust with respect to the change of the irrelevant part of the plant net. We aim to improve the permissiveness of the monitor-based supervisor in our future work.

References

[1] F. Basile, M. P. Cabasino, and C. Seatzu. State estimation and fault diagnosis of labeled time Petri net systems with unobservable transitions. *IEEE Transactions on Automatic Control*, 60(4):997–1009, 2015.

[2] 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*, 49(11):3432–3439, 2013.

[3] F. Basile, R. Cordone, and L. Piroddi. A branch and bound approach for the design of decentralized supervisors in Petri net models. *Automatica*, 52:322–333, 2015.

[4] P. Bonhomme. Marking estimation of P-time Petri nets with unobservable transitions. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(3):508–518, 2015.

[5] W. Bruns and B. Ichim. Normaliz: Algorithms for affine monoids and rational cones. *Journal of Algebra*, 324(5):1098–1113, 2010.

[6] M. Cabasino, C. N. Hadjicostis, and C. Seatzu. Probabilistic marking estimation in labeled Petri nets. *IEEE Transactions on Automatic Control*, 60:528–533, 2015.

[7] Maria Paola Cabasino, Christoforos N. Hadjicostis, and Carla Seatzu. Marking observer in labeled Petri nets with application to supervisory control. *IEEE Transactions on Automatic Control*, 62(4):1813–1824, 2017.

[8] C. G. Cassandras and S. Lafortune. *Introduction to discrete event systems*. Springer, 2008.

[9] Y. F. Chen, Z. W. Li, K. Barkaoui, and A. Giua. On the enforcement of a class of nonlinear constraints on Petri nets. *Automatica*, 55:116–124, 2015.

[10] D. Chubarov and A. Voronkov. Basis of solutions for a system of linear inequalities in integers: Computation and applications. In *Proceedings of the 30th International Symposium on Mathematical Foundations of Computer Science*, pages 260–270. Springer Berlin Heidelberg, 2005.

[11] A. Giua, F. DiCesare, and M. Silva. Generalized mutual exclusion constraints for Petri nets with uncontrollable transitions. In *Proceedings of the IEEE Int. Conf. on Systems, Man, and Cybernetics*, pages 947–949, Chicago, USA, 1992.

[12] A. Giua, S. Lafortune, and C. Seatzu. Divergence properties of labeled Petri nets and their relevance for diagnosability analysis. *IEEE Transactions on Automatic Control*, page to appear. DOI: 10.1109/TAC.2019.2947650, 2020.

[13] Alessandro Giua. *Supervisory Control of Petri Nets with Language Specifications*, pages 235–255. Springer, London, 2013.

[14] Miki Hermann, Laurent Juban, and Phokion G. Kolaitis. On the complexity of counting the hilbert basis of a linear diophantine system. In Harald Ganzinger, David McAllester, and Andrei Voronkov, editors, *Logic for Programming and Automated Reasoning*, pages 13–32, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[15] L. E. Holloway, B. H. Krogh, and A. Giua. A survey of Petri net methods for controlled discrete event systems. *Discrete Event Dynamic Systems: Theory and Applications*, 7(2):151–190, 1997.

[16] M. V. Iordache and P. J. Antsaklis. Petri net supervisors for disjunctive constraints. In *Proceedings of the 26th American Control Conference*, pages 4951–4956, New York, USA, 2007.

[17] M. V. Iordache, P. Wu, F. Zhu, and P. J. Antsaklis. Efficient design of Petri-net supervisors with disjunctive specifications. In *Proceedings of the IEEE Int. Conf. on Automation Science and Engineering*, pages 936–941, Madison, USA, 2013.

[18] J. L. Luo, H. J. Ni, W. M. Wu, S. G. Wang, and M. C. Zhou. Simultaneous reduction of Petri nets and linear constraints for efficient supervisor synthesis. *IEEE Transactions on Automatic Control*, 60(1):88–103, 2015.

[19] J. L. Luo, W. M. Wu, H. Y. Su, and J. Chu. Supervisor synthesis for enforcing a class of generalized mutual exclusion constraints on Petri nets. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 39(6):1237–1246, 2009.

[20] J. L. Luo and M. C. Zhou. Petri-net controller synthesis for partially controllable and observable discrete event systems. *IEEE Transactions on Automatic Control*, 62(3):1301–1313, 2017.

[21] Z. Ma, Z. He, Z. Li, and A. Giua. Design of monitor-based supervisors in labelled Petri nets. In *Proceedings of the 14th International Workshop on Discrete Event Systems*, pages 374–380, Sorrento, Italy, 2018.

[22] Z. Ma, Z. Li, and A. Giua. Design of optimal Petri net controllers for disjunctive generalized mutual exclusion constraints. *IEEE Transactions on Automatic Control*, 60(7):1774–1785, 2015.

[23] Z. Ma, Z. Li, and A. Giua. Characterization of admissible marking sets in Petri nets with conflicts and synchronizations. *IEEE Transactions on Automatic Control*, 62(3):1329–1341, 2017.

[24] Z. Ma, Y. Tong, Z. Li, and A. Giua. Basis marking representation of Petri net reachability spaces and its application to the reachability problem. *IEEE Transactions on Automatic Control*, 62(3):1078–1093, 2017.

[25] J. Moody and P. Antsaklis. Petri net supervisors for DES with uncontrollable and unobservable transitions. *IEEE Transactions on Automatic Control*, 45(3):462–476, 2000.

[26] A. Nazeem and S. Reveliotis. Maximally permissive deadlock avoidance for resource allocation systems with r/w-locks. *Discrete Event Dynamic Systems*, 25(1):31–63, 2015.

[27] Dmitrii V. Pasechnik. On computing hilbert bases via the elliot-macmahon algorithm. *Theoretical Computer Science*, 263(1):37–46, 2001. Combinatorics and Computer Science.

[28] Y. Ru, M. P. Cabasino, A. Giua, and C. N. Hadjicostis. Supervisor synthesis for discrete event systems under partial observation and arbitrary forbidden state specifications. *Discrete Event Dynamic Systems*, 24(3):275–307, 2014.

[29] Y. Tong, H. Lan, and J. Guo. Verification of detectability in labeled Petri nets. In *Proceedings of the 2019 American Control Conference*, pages 5627–5632, July 2019.

[30] X. Yin. Verification of prognosability for labeled Petri nets. *IEEE Transactions on Automatic Control*, 63(6):1828–1834, 2018.