Discrete Event Dynamic Systems (2021) 31:373–405
https://doi.org/10.1007/s10626-021-00340-6

On computing the supremal right-closed control invariant subset of a right-closed set of markings for an arbitrary petri net

Roshanak Khaleghi[1] · **Ramavarapu S. Sreenivas**[2]

Received: 27 December 2019 / Accepted: 15 February 2021 / Published online: 25 March 2021

Abstract
A set of non-negative integral vectors is said to be *right-closed* if the presence of a vector in the set implies all term-wise larger vectors also belong to the set. A set of *markings* is *control invariant* with respect to a *Petri Net* (PN) structure if the firing of any *uncontrollable transition* at any marking in this set results in a new marking that is also in the set. Every right-closed set of markings has a unique *supremal control invariant subset*, which is the largest subset that is control invariant with respect to the PN structure. This subset is not necessarily right-closed. In this paper, we present an algorithm that computes the supremal right-closed control invariant subset of a right-closed of markings with respect to an arbitrary PN structure. This set plays a critical role in the synthesis of *Liveness Enforcing Supervisory Policies* (LESPs) for a class of PN structures, and consequently, the proposed algorithm plays a key role in the synthesis of LESPs for this class of PN structures.

Keywords Supervisory Control · Petri Nets · Control Invariance · Right-closed sets

1 Introduction

In this paper we consider *Petri Net* (PN) structures that model *Discrete Event Dynamic Systems* (DEDS). The PN structure consists of a finite set of *places*, *transitions*, and *arcs* that have *weights* associated with them. The PN structure is initialized by placing a non-negative

✉ Roshanak Khaleghi
khalegh2@illinois.edu

Ramavarapu S. Sreenivas
rsree@illinois.edu

[1] Industrial and Enterprise Systems Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

[2] Coordinated Science Laboratory, Industrial and Enterprise Systems Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

number of *tokens* in each of its places, known as the *initial marking*, represented as a non-negative integral vector. If the number of tokens in each input place of a transition is no less than the weight associated with the arc from the input place to transition, the transition is said to be *enabled*. An enabled transition can *fire*, which results in a different set of tokens associated with each place; that is, the firing of the transition can result in a new *marking*. This process can be repeated as often as necessary. Each marking of a PN structure is an integral vector whose dimension is the same as the number of places of the PN.

It is of interest to control the behavior of a PN model by *supervision* such that the markings of the supervised-PN never leaves a desired set. To this end, the set of transitions is partitioned into *controllable*- and *uncontrollable* subsets. A *supervisory policy* can (resp. cannot) prevent the firing of a controllable (resp. uncontrollable) transition that is enabled at a marking of the PN.

A set of integral vectors, which represents different markings of a PN structure, is said to be *control invariant* with respect to the PN structure if the firing of any *uncontrollable transition* at any marking that belongs to this set results in a new marking that also belongs to the same set. Stated differently, only the firing of a controllable transition at a marking belonging to a control invariant set of markings will result in a new marking that does not belong to the set. There is a supervisory policy that ensures the markings of a supervised-PN stays within a set of integral vectors if and only if the desired set is *control invariant* with respect to the PN structure (Ramadge and Wonham 1987).

Every set of integral vectors, corresponding to the markings of a PN, has a unique, largest subset that is control invariant with respect to a given PN structure, identified as its *supremal control invariant subset*. This follows from the fact that the union of two sets of control invariant markings is also control invariant with respect to the PN structure (Ramadge and Wonham 1987). The supervisory policy of permitting a (controllable) transition at a marking that belongs to the supremal control invariant subset *only* if its firing results in a new marking that also belongs to the same set, is the *minimally restrictive* policy that ensures the marking of the PN remains in the original set of markings.

A set of integral vectors is said to be *right-closed* if the presence of a vector in the set implies all integral vectors that are term-wise larger also belong to the same set (Valk and Jantzen 1985). The empty-set is right-closed by definition. Any vector of a right-closed set which is not term-wise larger than other vectors of the set is known as a *minimal element* of the right-closed set. Consequently, a non-empty right-closed set is uniquely identified by its finite set of *minimal elements*.

The supremal control invariant subset of a right-closed set of markings is not necessarily right-closed for a PN structure. However, it always contains a right-closed subset that is control invariant with respect to the PN structure. This follows from the observations that (i) the empty-set is right-closed by definition, (ii) the union of two sets of markings that are control invariant with respect to a PN structure is also control invariant with respect to the PN structure, and (iii) the union of two right-closed sets of markings is also right-closed.

Section 1.1 highlights an important application of supervisory control where it is of interest to identify the supremal right-closed control invariant subset of a right-closed set of markings. In this paper, we present the formal details of an algorithm for computation of the supremal right-closed control invariant subset of a *right-closed* desirable set with respect to an arbitrary PN structure. Our proposed algorithm, discussed in detail in Section 3, follows an exhaustive search approach based on the *Depth-First-Search* (DFS) strategy on a decision tree. The results of this paper present the formal treatment of a critical component of a software tool developed for the application reviewed below (Chandrasekaran et al. 2015).

1.1 Motivation

A Discrete Event System is *live* if, irrespective of the past, all its events can be executed in the future (Alpern and Schneider 1985). Equivalently, a PN is said to be live if all its transitions can be fired, although not immediately, from every reachable marking. A *Liveness Enforcing Supervisory Policy* (LESP) determines the set of controllable transitions that can be permitted to fire at a marking such that the supervised-PN is live. For any PN structure, the set of initial markings for which there is an LESP is control invariant with respect to its structure. The minimally restrictive LESP effectively prevents the firing of a controllable transition at a marking if its firing would result in a new marking for which no LESP exists. The existence of an LESP for an arbitrary PN structure is undecidable (Sreenivas 1997, 2012). However, there are families of PN structures for which the existence of an LESP is decidable, and for each of these families the above mentioned set of initial markings is right closed, and control invariant with respect to the PN structure (cf. Sreenivas 2012; Salimi et al. 2015; Chen et al. 2020). The synthesis of an LESP for any member of these families of PN structures requires the computation of the supremal right-closed control invariant subset of a right-closed set of markings. This paper presents the formal treatment of the algorithm for this important step, which has been used, without formal proof, in the algorithms for LESP-synthesis in reference (Chandrasekaran et al. 2015). Regarding the practical application of our proposed methodology, reference (Khaleghi et al. 2019) provides an example of LESP-synthesis, which necessitates computation of control invariant subset of a right-closed set, in a manufacturing setting. The example in reference (Khaleghi et al. 2019) describes a simplified Petri Net model of an automated system that paints automobile bodies using two paint-booths, and two robots. A more detailed discussion of this practical example is presented in Section 4.3. It should be noted that the LESP-synthesis, as discussed in Khaleghi et al. (2019), involves additional work than just computing the largest right-closed control invariant subset of a given right-closed set.

Prior work that is relevant to the algorithm in this paper is reviewed in the next subsection.

1.2 Prior work

Ramadge and Wonham (1987) considered the synthesis of a supervisory policy that ensures the set of states reachable under supervision is a desired subset of the set of all possible states for a given *Discrete Event System* (DES). The desired supervisory policy exists if and only if the desired subset is control invariant with respect to the set of states of the DES; that is, if the occurrence of an uncontrollable event at any state within the desired subset results in a new state that is also within the desired subset as well.

There can be instances of supervisory control of DES where the objective of supervisory control is to avoid a set of undesirable states. Kumar and Garg (2005) considered PN models of DES where the undesirable set of states (i.e. markings) are right-closed. That is, if a marking is undesirable, then all term-wise larger markings are undesirable, as well. The goal of supervisory control for this class of problems is to ensure the marking of the PN structure stay within a desired set of markings, which is the complement of the right-closed set of forbidden markings. Kumar and Garg showed that the existence of a supervisory policy for this class of problems is decidable. To illustrate the fact that this class of supervisory control problems are different from the problem considered in this paper, we note that the supremal control invariant subset of desired markings for this class of problems is also a complement

of a right-closed set of markings (cf. Lemma 2, Section IV, Kumar and Garg 2005). This is not the case for the class of problems considered in this paper, in that the supremal control invariant subset of the desirable set of right-closed markings is not necessarily right-closed.

There is a large volume of literature on liveness enforcement in PN models of DES systems that do not invoke Ramadge and Wonham's Theory of Supervisory Control (Ramadge and Wonham 1987). Most of these approaches augment the existing PN structure with additional structure, using concepts from the theory of PNs (Murata 1989; Peterson 1981), to arrive at a new PN structure that satisfies the desired property. These approaches involve the use of *linear constraints*, *place-invariants*, and other artefacts from PN theory to arrive at the required structural modifications. References (Moody and Antsaklis 1998; Iordache and Ansaklis 2006) consider situations where linear inequalities, if known in advance to enforce liveness, can be enforced by the use of *monitor-places*. To synthesize minimally restrictive, *closed-loop* liveness for a class of Marked Graph Petri Nets, Basile et al. (2009) provided sufficient conditions enforced by *Generalized Mutual Exclusion Constraints* (GMECs). Proposing a *constraint transformation/constrain reduction* approach, Luo and Nonami (Luo and Nonami 2011) investigated the problem of eliminating redundant constraints in synthesis of supervisory policies, which can result in reduction of the number of monitors in *invariant-based* supervisors. Dideban and Alla (2008) introduced a set of linear constraints that prevents reachability of specific states. They presented the concept of *over-state* in order to build the simplest constraints, which forbid a greater number of states and which reduces the number of monitors in an invariant-based controller for *safe* PNs. Reference (Dideban and Zeraatkar 2018) employs the place-invariant property of Petri Nets for synthesis of controllers for large-scale systems by breaking down the original Petri Net model into smaller models. Their proposed algorithm addresses the problem of forbidden states that results either from deadlocks or from presence of uncontrollable transitions. To design maximally permissive liveness enforcing supervisors, which avoid deadlocks in flexible manufacturing systems, reference (Chen et al. 2011) proposed a computationally efficient approach to design optimal control places. The reachability graph is computed by using a combination of a *vector covering* approach and *binary decision diagrams*. In general, not all liveness enforcement problems involving PNs can be represented as the requirement that the markings satisfy a (finite) set of linear constraints (cf. figure 1, Salimi et al. 2015), and oftentimes there is no choice but to use a supervisory policy to prevent the firing of a controllable transition, as opposed to using the augmented-PN structure to do the same. When applicable, the additional structural enhancements to the original PN can be implicitly interpreted as a supervisory policy. That said, these methods do not explicitly deal with the notion of control invariance, which is the main focus of the algorithm presented in this paper.

The paper is organized as follows: Section 2 introduces the definitions and notations. Section 3 presents the preliminary results and introduces a Depth-First-Search based algorithm for finding the supremal right-closed control invariant subset of a right-closed set. Section 4 contains three PN examples which illustrate the algorithm introduced in Section 3. Section 5 includes main results of this paper. We conclude with some discussion and future research directions in Section 6.

2 Definitions and notations

We use $\mathbb{Z}$ (resp. $\mathbb{Z}^+$) to denote the set of non-negative (resp. positive) integers, and $\mathbb{Z}^n$ denotes the set of n-dimensional integral vectors, where $n \in \mathbb{Z}^+$. We use $\mathbf{x}_i$ to denote the

i-th component of a vector $\mathbf{x} \in \mathbb{Z}^n$. The set of $n \times m$ non-negative integral matrices is represented by $\mathbb{Z}^{n\times m}$. For any $\mathbf{A} \in \mathbb{Z}^{n\times m}$, $\mathbf{A}_{i,j}$ denotes the value of the i-th row and j-th column of the $n \times m$ matrix $\mathbf{A}$. Given two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{Z}^n$, we use the notation $\mathbf{x} \geq \mathbf{y}$ if $\mathbf{x}_i \geq \mathbf{y}_i$ for all $i \in \{1, 2, \ldots n\}$. We use the term $max\{\mathbf{x}, \mathbf{y}\}$ to denote the term-wise maximum of the two vectors in the argument. The cardinality of a set argument is denoted by $card(\bullet)$.

2.1 Petri nets

A *Petri Net structure* $N = (\Pi, T, \mathbf{IN}, \mathbf{OUT})$ is a bipartite weighted directed graph, where $\Pi = \{p_1, p_2, ..., p_n\}$ and $T = \{t_1, t_2, ..., t_m\}$ denote two disjoint sets of vertices called *places* and *transitions*, respectively. $\mathbf{IN} \in \mathbb{Z}^{n\times m}$ (resp. $\mathbf{OUT} \in \mathbb{Z}^{n\times m}$) represents the adjacency relationship ($\Pi \times T$) (resp. ($T \times \Pi$)). Specifically, $\mathbf{IN}_{i,j} = w$ (resp. $\mathbf{OUT}_{i,j} = w$) denotes that there is a directed arc from p_i to t_j (resp. t_j to p_i) with weight w. The *incidence matrix* $\mathbf{C}$ of the PN structure N is an $n \times m$ matrix, where $\mathbf{C} = \mathbf{OUT} - \mathbf{IN}$. We use the notation $\mathbf{IN}_{t_j}$ ($\mathbf{OUT}_{t_j}$) to denote the column of the input (resp. output) matrix corresponding to transition t_j. For purposes of supervisory control, the set of transitions are partitioned as $T = T_u \cup T_c$, where T_u (resp. T_c) represents the subset of *uncontrollable* (reps. *controllable*) transitions.

Each structure N has an *initial marking* $\mathbf{m}^0 \in \mathbb{Z}^n$ associated with it, where the place p_i is assigned $\mathbf{m}_i^0$-many *tokens* at initialization. We will use the term *Petri net* (PN) and the symbol $N(\mathbf{m}^0)$ to denote a PN structure N along with its initial marking $\mathbf{m}^0$.

In graphic representations of PN structures, controllable (resp. uncontrollable) transitions are shown by filled (resp. unfilled) rectangles. Places are represented by circles, and tokens are represented by smaller filled-circles that are inside the circles representing the places. Arcs are represented by directed edges; for the sake of brevity, only non-unitary weights are explicated in graphical representations of PN structures. Any given marking $\mathbf{m}$ can be interpreted as an integral vector in $\mathbb{Z}^n$ which illustrates the distribution of tokens among places at any instant. The supervisory policy $\mathcal{P} : \mathbb{Z}^n \times T \to \{0, 1\}$ is a function that assigns a 0 or 1 for each transition and each marking. Additionally, $\forall \mathbf{m} \in \mathbb{Z}^n, \forall t_u \in T_u, \mathcal{P}(\mathbf{m}, t_u) = 1$.

A transition $t_j \in T$ is said to be *state-enabled* (resp. *control-enabled*) at a marking $\mathbf{m}^k \in \mathbb{Z}^n$ if $\mathbf{m}^k \geq \mathbf{IN}_{t_j}$ (resp. $\mathcal{P}(\mathbf{m}^k, t_j) = 1$). The set of state-enabled transitions at marking $\mathbf{m}^k$ is denoted by $T_e(N, \mathbf{m}^k)$. The set of uncontrollable transitions are control-enabled for all markings. A transition $t_j \in T$ that is state and control-enabled can *fire*, which results in a new marking $\mathbf{m}^{k+1} \in \mathbb{Z}^n$, where $\mathbf{m}^{k+1} = \mathbf{m}^k + \mathbf{C}_{t_j}$. This is represented as $\mathbf{m}^k \xrightarrow{t_j} \mathbf{m}^{k+1}$.

2.2 Right-closed set of markings

A set of markings $\mathcal{M} \subseteq \mathbb{Z}^n$ is *right-closed* if $((\mathbf{m}^1 \in \mathcal{M}) \wedge (\mathbf{m}^2 \geq \mathbf{m}^1) \Rightarrow (\mathbf{m}^2 \in \mathcal{M}))$. Every right-closed set $\mathcal{M}$ contains a finite set of *minimal elements* denoted by $min(\mathcal{M}) \subseteq \mathcal{M}$; the set of minimal elements, $min(\mathcal{M})$, can be used to represent $\mathcal{M}$. The set of markings $\{\widehat{\mathbf{m}}^{k+1} \in \mathbb{Z}^n \mid \widehat{\mathbf{m}}^k \geq \mathbf{m}^k, \widehat{\mathbf{m}}^k \geq \mathbf{IN}_{t_j}, \widehat{\mathbf{m}}^{k+1} = \widehat{\mathbf{m}}^k + \mathbf{C}_{t_j}\}$ is right-closed, and is identified by the minimal element $\max(\mathbf{m}^k, \mathbf{IN}_{t_j}) + \mathbf{C}_{t_j}$. The empty-set is right-closed by definition.

We use the notation $\mathbf{m}^k \overset{t_j}{\rightsquigarrow} \mathbf{m}^{k+1}$ to denote the fact $\mathbf{m}^{k+1} = \max(\mathbf{m}^k, \mathbf{IN}_{t_j}) + \mathbf{C}_{t_j}$. To elaborate, this notation is used to denote that if transition $t_j \in T$ were enabled to fire at a marking which is greater than or equal to $\mathbf{m}^k$, the marking that results from the firing of t_j will be greater than or equal to marking $\mathbf{m}^{k+1}$.

2.3 Control invariance property

A set of marking $\mathcal{M}$ is said to be *control invariant* with respect to a PN structure N if $((\mathbf{m}^1 \in \mathcal{M}) \wedge (t_u \in T_u) \wedge (\mathbf{m}^1 \geq \mathbf{IN}_{t_u}) \wedge (\mathbf{m}^2 = \mathbf{m}^1 + \mathbf{C}_{t_u}) \Rightarrow (\mathbf{m}^2 \in \mathcal{M}))$. That is, the firing of any state-enabled, uncontrollable transition at a marking in $\mathcal{M}$ will result in a new marking that is also in $\mathcal{M}$. If $\mathcal{M}_1, \mathcal{M}_2 \subseteq \mathcal{M}$ are two control invariant subsets of $\mathcal{M}$, then $\mathcal{M}_1 \cup \mathcal{M}_2 \subseteq \mathcal{M}$ is also a control invariant subset of $\mathcal{M}$. Consequently, there is a unique largest (in terms of set-containment) subset of $\mathcal{M}$, the *supremal control invariant subset* $\mathcal{M}^{\uparrow} \subseteq \mathcal{M}$, that is control invariant with respect to N (Section 7, Ramadge and Wonham 1987). We drop the reference to the PN structure N if its identity is unambiguous.

If $\mathcal{M}$ is right-closed, then $\mathcal{M}$ is control invariant if and only if $\forall \mathbf{m}^i \in min(\mathcal{M}), \forall t_u \in T_u, \exists \mathbf{m}^j \in min(\mathcal{M})$ such that:

$$max\{\mathbf{m}^i, \mathbf{IN}_{t_u}\} + \mathbf{C}_{t_u} \geq \mathbf{m}^j \text{(Lemma 5.10, Sreenivas 2012).} \tag{1}$$

That is, $(\mathbf{m}^i \overset{t_u}{\rightsquigarrow} \widehat{\mathbf{m}}^i) \Rightarrow (\widehat{\mathbf{m}}^i \in \mathcal{M})$, where $\widehat{\mathbf{m}}^i = \max(\mathbf{m}^i, \mathbf{IN}_{t_u}) + \mathbf{C}_{t_u}$. This is the *control invariance (CI) condition*, where minimal element $\mathbf{m}^i$ is *covered* by minimal element $\mathbf{m}^j$ for transition $t_u \in T_u$. For a PN structure N and a right-closed set of markings $\mathcal{M}$, the supremal control invariant subset $\mathcal{M}^{\uparrow}$ is not necessarily right-closed. However, there is a unique supremal subset $\mathcal{M}^{\Uparrow} \subseteq \mathcal{M}^{\uparrow}$ that is right-closed, and control invariant with respect to N. This follows directly from the fact that (a) the empty-set is right-closed by definition, (b) the union of two right-closed sets is also right-closed, and (c) the union of two control invariant sets of markings is also control invariant with respect to N. The right-closed set $\mathcal{M}^{\Uparrow}$ is characterized by the property: $\forall \mathbf{m}^1 \in min(\mathcal{M}^{\Uparrow}), \forall t_u \in T_u, (\mathbf{m}^1 \overset{t_u}{\rightsquigarrow} \widehat{\mathbf{m}}^1) \Rightarrow (\widehat{\mathbf{m}}^1 \in \mathcal{M}^{\Uparrow})$.

For the PN structure N_1 shown in Fig. 1a, consider the right-closed set $\mathcal{M}_0$ identified by minimal elements $min(\mathcal{M}_0) = \{(1,0)^T, (0,2)^T\}$, where $(1,0)^T$ (resp. $(0,2)^T$) corresponds to transpose of $(1,0)$ (resp. $(0,2)$). For $\mathcal{M}_0$, we have $\mathcal{M}_0^{\uparrow} = \mathcal{M}_1 \cup \{(1,0)^T\}$, where $\mathcal{M}_1$ is the right-closed set identified by minimal elements $min(\mathcal{M}_1) = \{(3,0)^T, (1,1)^T, (0,2)^T\}$. That is, $\mathcal{M}_0^{\uparrow}$ is not right-closed, and $\mathcal{M}_1 (= \mathcal{M}_0^{\Uparrow})$ is the supremal right-closed control invariant subset of $\mathcal{M}_0$.

Our proposed algorithm for finding the supremal right-closed control invariant subset of an initial right-closed set will require elevating the minimal elements of the current estimate of the subset in various ways. We introduce some of the notations pertaining to the elevation process in the next section using the example shown in Fig. 1.

3 Computing the supremal right-closed control invariant subset of a right-closed set

The *Control Invariance Graph (CI-graph)* of the right-closed set $\mathcal{M}$ with respect to the Petri Net structure N is defined as $\mathcal{G}(\mathcal{M}, N) = (min(\mathcal{M}), E)$ where $min(\mathcal{M}) = \{\mathbf{m}^1, \mathbf{m}^2, \ldots, \mathbf{m}^k\}$ are the nodes of the graph and

$$E = \{e_{\mathbf{m}^i, t_u, \mathbf{m}^j} \mid max\{\mathbf{m}^i, \mathbf{IN}_{t_u}\} + \mathbf{C}_{t_u} \geq \mathbf{m}^j\}.$$

That is, there is a directed arc from node $\mathbf{m}^i$ to $\mathbf{m}^j$ with a label $t_u \in T_u$ if $\mathbf{m}^i$ is *covered* by $\mathbf{m}^j$ for $t_u \in T_u$. The markings of the CI-graph may have self-loops since a marking such as $\mathbf{m}^i$ can be covered by itself upon firing of an uncontrollable transition such as t_u. This also implies that $\mathbf{m}^j$ can be the same as $\mathbf{m}^i$ in Eq. 1. The set of minimal elements $min(\mathcal{M})$ can be partitioned into two sets – $min(\mathcal{M}) = \mathcal{A}^{fail} \cup \mathcal{A}^{pass}$, where $\mathcal{A}^{fail} \cap \mathcal{A}^{pass} = \emptyset$.

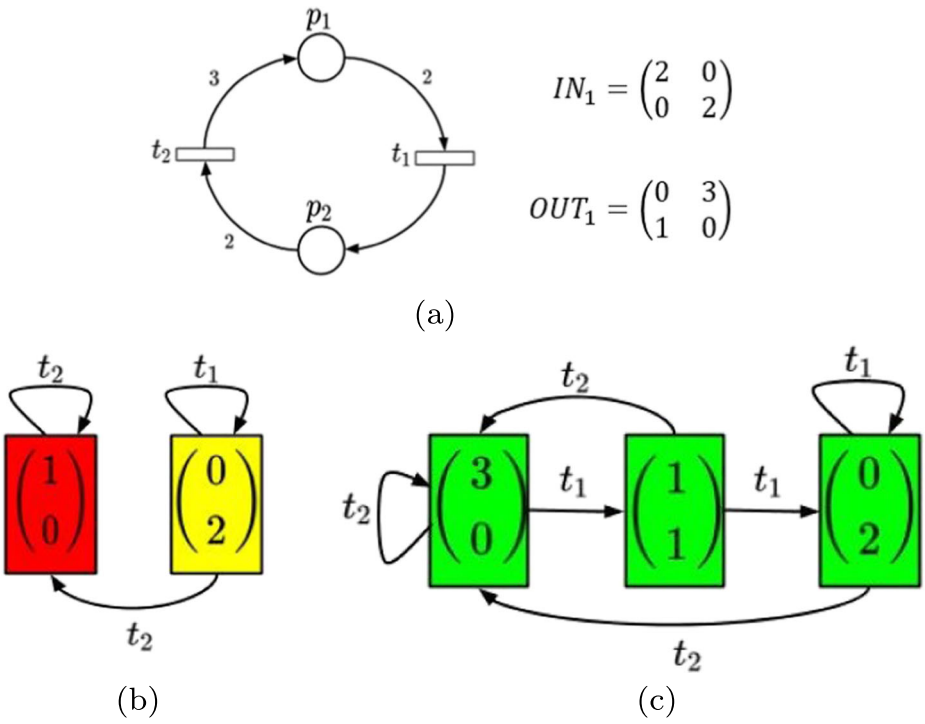

Fig. 1 The PN structure N_1, the CI-graph of N_1 for $min(\mathcal{M}_0) = \{(1\ 0)^T, (0\ 2)^T\}$ and $min(\mathcal{M}_1) = \{(3\ 0)^T, (1\ 1)^T, (0\ 2)^T\}$. **a** N_1 **b** $\mathcal{G}(\mathcal{M}_0, N_1)$ **c** $\mathcal{G}(\mathcal{M}_1, N_1)$

For each $\mathbf{m}^i \in \mathcal{A}^{pass}, \forall t_u \in T_u, \exists \mathbf{m}^j \in \mathcal{A}^{pass}$ where Eq. 1 is satisfied; and $\mathcal{A}^{fail} = min(\mathcal{M}) - \mathcal{A}^{pass}$. Also, for each $\mathbf{m}^i \in \mathcal{A}^{fail}$, the set of *critical transitions*, denoted by $\mathcal{CRT}(\mathbf{m}^i)$, contains all uncontrollable transitions for which $\mathbf{m}^i$ fails the CI condition test; i.e., for all $t_j \in \mathcal{CRT}(\mathbf{m}^i)$, $(\mathbf{m}^i \overset{t_j}{\rightsquigarrow} \widehat{\mathbf{m}}^i)$ and $(\widehat{\mathbf{m}}^i \notin \mathcal{M})$. In graphical representation of a CI-graph, we use *red* color-coding for $\mathbf{m}^i$ if $\mathbf{m}^i \in \mathcal{A}^{fail}$ and $\mathcal{CRT}(\mathbf{m}^i) \neq \emptyset$, *yellow* color-coding if $\mathbf{m}^i \in \mathcal{A}^{fail}$ and $\mathcal{CRT}(\mathbf{m}^i) = \emptyset$, and *green* color-coding if $\mathbf{m}^i \in \mathcal{A}^{pass}$.

As an illustration, consider the PN structure N_1 of Fig. 1a, and the right-closed set $\mathcal{M}_0$, where $min(\mathcal{M}_0) = \{(1\ 0)^T, (0\ 2)^T\}$. For N_1, both transitions are uncontrollable; consequently, the supervisory policy cannot prevent the firing of transitions t_1 or t_2. Figure 1(b) shows the CI-graph of $\mathcal{M}_0$ with respect to the PN structure N_1. The nodes of this directed graph are members of $\mathcal{M}_0$; each directed edge $e_{\mathbf{m}^i, t_u, \mathbf{m}^j}$ originates from $\mathbf{m}^i$ and terminates on $\mathbf{m}^j$ with label $t_u \in T_u$. For brevity, if there are multiple parallel edges between two nodes with different labels, we draw it as a single directed edge with multiple labels.

If we apply (1) to the minimal element $(1\ 0)^T$ for $t_1 \in T_u$, the left-hand-side expression evaluates to $(0\ 1)^T$, which is not in $\mathcal{M}_0$. Consequently, there is no outgoing arc from $(1\ 0)^T$ with a label t_1 in $\mathcal{G}(\mathcal{M}_0, N_1)$; and minimal element $\{(1\ 0)^T\}$ cannot be in $\mathcal{A}^{pass}(\mathcal{M}_0)$. When Eq. 1 is applied to $(1\ 0)^T$ for $t_2 \in T_u$, we get the vector $(4\ 0)^T (\geq (1\ 0)^T)$, which means $(1\ 0)^T$ is covered by itself for transition t_2 and therefore, there is a self-loop from $(1\ 0)^T$ to itself with label t_2. For the minimal element $(1\ 0)^T$, we have $\mathcal{CRT}((1\ 0)^T) = \{t_1\}$ and we use *red* color-coding for this marking as it fails the CI test for at least one uncontrollable transition, i.e. t_1. When Eq. 1 is applied to the minimal element $(0\ 2)^T$ for $t_2 \in T_u$, we

get the vector $(3\ 0)^T (\geq (1\ 0)^T \notin \mathcal{A}^{pass}(\mathcal{M}_0))$, therefore $(0\ 2)^T$ cannot be in $\mathcal{A}^{pass}(\mathcal{M}_0)$ either. However, in this case, we use *yellow* color-coding for this marking to signify the fact that although $\mathcal{CRT}((0\ 2)^T) = \emptyset$, $(0\ 2)^T$ is still a member of $\mathcal{A}^{fail}(\mathcal{M}_0)$. From Fig. 1(b), we infer $\mathcal{A}^{pass}(\mathcal{M}_0) = \emptyset$ and $\mathcal{A}^{fail}(\mathcal{M}_0) = \{(1\ 0)^T, (0\ 2)^T\}$. The right-closed set $\mathcal{M}_0$ is not control invariant with respect to the PN structure N_1. Figure 1c shows the CI-graph of $\mathcal{M}_1$ where $min(\mathcal{M}_1) = \{(3\ 0)^T, (1\ 1)^T, (0\ 2)^T\}$ with respect to the PN structure N_1. From Fig. 1c, we can see that $\mathcal{A}^{fail}(\mathcal{M}_0) = \emptyset$ and $\mathcal{A}^{pass}(\mathcal{M}_0) = \{(3\ 0)^T, (1\ 1)^T, (0\ 2)^T\}$ and thus, all the markings are color-coded as *green*. Therefore, the right-closed set $\mathcal{M}_1$ is control invariant with respect to the PN structure N_1.

We note that there can be multiple outgoing arcs with the same label from a vertex in the CI-graph. In general, if $card(min(\mathcal{M})) = k$ and $card(T_u) = l$, then number of edges in the CI-graph of $\mathcal{M}$ with respect to the PN structure N is bounded by $k^2 l$.

For a right-closed set $\mathcal{M}$, the observations introduced below follow directly from the above discussion.

Observation 1 There is a non-empty (right-closed) control invariant subset of the right-closed set $\mathcal{M}$ if $\mathcal{A}^{pass}(\mathcal{M}) \neq \emptyset$. This is indeed true because based on the definition of $\mathcal{A}^{pass}$, all minimal elements of $\mathcal{A}^{pass}(\mathcal{M})$ will always satisfy the CI test of equation 1, which implies $\mathcal{A}^{pass}(\mathcal{M})$, if not empty, will serve as a non-empty control invariant subset for $\mathcal{M}$.

Observation 2 $\mathcal{M}$ is control invariant if and only if $\mathcal{A}^{fail}(\mathcal{M}) = \emptyset$. Given the definition of $\mathcal{A}^{fail}$, if $\mathcal{A}^{fail}(\mathcal{M})$ is empty, it implies that all minimal elements of $\mathcal{M}$ belong to $\mathcal{A}^{pass}(\mathcal{M})$ and therefore, no minimal element of $\mathcal{M}$ will ever violate the CI test of Eq. 1, which means $\mathcal{M}$ is control invariant. Similarly, if $\mathcal{M}$ is control invariant, it implies that none of its minimal element fail the CI test, resulting in an empty $\mathcal{A}^{pass}(\mathcal{M})$.

Observation 3 $\mathcal{A}^{pass}(\mathcal{M}) \subseteq min(\mathcal{M}^{\Uparrow})$. This is due to the fact that minimal elements of $\mathcal{A}^{pass}(\mathcal{M})$ will always satisfy the CI test of equation 1 and therefore, members of $\mathcal{A}^{pass}(\mathcal{M})$ need not to be removed or replaced by a term-wise larger vector. So, minimal elements of $\mathcal{A}^{pass}(\mathcal{M})$ will be contained in the final supremal right-closed control invariant subset, $min(\mathcal{M}^{\Uparrow})$.

The elevation of selected members of $\mathcal{A}^{fail}(\mathcal{M})$ to term-wise larger vectors forms the basis of the algorithm that computes $min(\mathcal{M}^{\Uparrow})$. The procedural aspects to this operation are described in detail in the remainder of this section.

3.1 Description of the elevation process

Let $\mathcal{M}_0$ be an initial, right-closed set of markings. Suppose $\mathcal{M}_i \subset \mathcal{M}_0$ is the current estimate of a control invariant subset of $\mathcal{M}_0$. Let $\mathbf{m}^1 \in \mathcal{A}^{fail}(\mathcal{M}_i)$, and $t_u \in T_u$ such that $\mathbf{m}^1 \overset{t_u}{\rightsquigarrow} \widehat{\mathbf{m}}$, and $\widehat{\mathbf{m}} \notin \mathcal{M}_i$. This implies that $\mathbf{m}^1$ is a minimal element that fails the CI test for t_u. Since we are looking for a right-closed subset of $\mathcal{M}_i$, the next estimate will have $\mathbf{m}^1$ replaced by a set of minimal elements that are larger than $\mathbf{m}^1$ so they satisfy equation 1. Consequently, $\mathbf{m}^1$ is to be elevated by the smallest vector $\mathbf{x} \in \mathbb{Z}^n$ such that $(\mathbf{m}^1 + \mathbf{x}) \overset{t_u}{\rightsquigarrow} \widehat{\mathbf{m}}^1$, and $\widehat{\mathbf{m}}^1 \geq \mathbf{m}^2$, where $\mathbf{m}^2 \in (min(\mathcal{M}_i) - \{\mathbf{m}^1\})$. The p-th component of $\mathbf{x}$, denoted by $\mathbf{x}(p)$, if elevated, should be such that $\max\{\mathbf{m}^1(p) + \mathbf{x}(p), \mathbf{IN}_{t_u}(p)\} + \mathbf{C}_{t_u}(p) = \mathbf{m}^2(p)$. Let $\overline{\mathbf{m}^1}$ be

the marking that results from elevating $\mathbf{m}^1$ with respect to $\mathbf{m}^2$. As a result of the elevation, for the p-th component of $\overline{\mathbf{m}^1}$, we have:

$$\overline{\mathbf{m}^1}(p) = \begin{cases} \mathbf{m}^2(p) - \mathbf{C}_{t_u}(p), & \text{if } max\{\mathbf{m}^1(p), \mathbf{IN}_{t_u}(p)\} < \mathbf{m}^2(p) - \mathbf{C}_{t_u}(p) \\ \mathbf{m}^1(p), & \text{otherwise} \end{cases} \tag{2}$$

If $\overline{\mathbf{m}^1}$ is term-wise larger than some member of $\min(\mathcal{M}_i) - \{\mathbf{m}^1\}$, the process of elevating $\mathbf{m}^1$ with respect to $\mathbf{m}^2$ for t_u is ignored ($\overline{\mathbf{m}^1}$ is said to be *dismissed*). If $\overline{\mathbf{m}^1}$ is not dismissed, to complete the elevation process of $\mathbf{m}^1$, we also elevate $\mathbf{m}^1$ with respect to the new minimal element $\overline{\mathbf{m}^1}$ by the same process as in Equation 2. The process of elevation of $\mathbf{m}^1$ with respect to the newly generated minimal elements continues till the generated minimal element is dismissed by any of the existing minimal elements. We refer to this whole process as the operation for *elevating* $\boldsymbol{m}^1$ *for control invariance with respect to* t_u *and* $\boldsymbol{m}^2$, and it is formally described in Algorithm 1, RAISE_FOR_CI($\mathbf{M}^1$, $\mathbf{M}^2$, t_u, N) of the Appendix section. We show that Algorithm 1 terminates in finite number of steps in Section 5.

Considering one iteration of Algorithm 1 where $\mathbf{m}^1$ is elevated by $\mathbf{x}$ with respect to $\mathbf{m}^2$ for transition t_u, resulting in the new non-dismissed marking $\overline{\mathbf{m}^1}(= \mathbf{m}^1 + \mathbf{x})$, we introduce the following notations/terminologies:

1. $\overline{\mathbf{m}^1}$ is a *(direct) child* of $\mathbf{m}^2$; likewise, $\mathbf{m}^2$ is a *parent* for $\overline{\mathbf{m}^1}$.
2. The notation $\overline{\mathbf{m}^1} \overset{t_u}{\Rightarrow} \mathbf{m}^2$ is used to show *child-parent* relationship of markings $\overline{\mathbf{m}^1}$ and $\mathbf{m}^2$ through transition t_u.
3. $\overline{\mathbf{m}^1}$ is a *descendant* of $\mathbf{m}^1$. The set of all descendants of $\mathbf{m}^1$ for transition t_u is denoted by $\mathcal{D}(\mathbf{m}^1, t_u)$. The set of all descendants of $\mathbf{m}^1$ raised with respect to $\mathbf{m}^2$ for transition t_u is denoted by $\mathcal{D}(\mathbf{m}^1, \mathbf{m}^2, t_u)$; therefore, $\mathcal{D}(\mathbf{m}^1, \mathbf{m}^2, t_u) \subseteq \mathcal{D}(\mathbf{m}^1, t_u)$.
4. In general, $\mathbf{m}^i$ is a child of $\mathbf{m}^k$ if there is a child-parent sequence of markings in $min(\mathcal{M}_i)$ and transitions in T_u through which $\mathbf{m}^i$ is connected to $\mathbf{m}^k$; e.g. $\mathbf{m}^i \overset{t_1}{\Rightarrow} \mathbf{m}^{i+1} \overset{t_2}{\Rightarrow} ... \overset{t_l}{\Rightarrow} \mathbf{m}^{k-1} \overset{t_u}{\Rightarrow} \mathbf{m}^k$. Note that in this given sequence, $\mathbf{m}^{k-1}$ is the only *direct child* of $\mathbf{m}^k$. The set of all children of a marking $\mathbf{m}^k$ is denoted by $\mathcal{C}(\mathbf{m}^k)$.
5. The combination of all existing child-parent sequences of markings in $min(\mathcal{M}_i)$ forms the collection of *child-parent trees* of $\mathcal{M}_i$ with respect to N defined as $\mathcal{T}(\mathcal{M}_i, N) = (min(\mathcal{M}_i), E)$ where $min(\mathcal{M}_i) = \{\mathbf{m}^1, \mathbf{m}^2, \ldots, \mathbf{m}^k\}$ are the nodes of the tree and

$$E = \{e_{\mathbf{m}^i, t_u, \mathbf{m}^j} \mid \mathbf{m}^i \text{ is a direct child of } \mathbf{m}^j\}.$$

That is, there is a directed arc from node $\mathbf{m}^i$ to $\mathbf{m}^j$ with a label t_u if $\mathbf{m}^i$ is a direct child of $\mathbf{m}^j$ for $t_u \in T_u$.

Starting from a right-closed set $\mathcal{M}_i$, Algorithm 1 elaborates the elevation process for a marking such as $\mathbf{m}^1 \in \mathcal{A}^{fail}(\mathcal{M}_i)$ which fails to satisfy the CI condition of Eq. 1 for transition t_u. A valid question here is "what happens to the child(ren) of $\mathbf{m}^1$ given the fact that $\mathcal{C}(\mathbf{m}^1) \neq \emptyset$?".

So, to consider a more general case, for a given right-closed set $\mathcal{M}_i$ with minimal elements $min(\mathcal{M}_i)$ and child-parent tree collection $\mathcal{T}(\mathcal{M}_i, N)$, Algorithm 2 outlines a complete round of elevation process for a marking $\mathbf{m}^p \in min(\mathcal{M}_i)$ which fails to satisfy the CI condition test for transition t_u. This process consists of two main stages. The first stage is the elevation of $\mathbf{m}^p$ for control invariance with respect to transition t_u. In this stage, $\mathbf{m}^p$ is elevated with respect to all $\mathbf{m}^k \in min(\mathcal{M}) - \{\mathbf{m}^p\} - \{\mathcal{C}(\mathbf{m}^p)\}$, using Algorithm 1. The output of this stage would be the descendant markings of $\mathbf{m}^p$ for transition t_u, $\mathcal{D}(\mathbf{m}^p, t_u)$. The second stage involves the process called as *elevation for control invariance forced by the*

parent; for this, all children of $\mathbf{m}^p$ (if any) needs to be raised with respect to all descendant markings of $\mathbf{m}^p$ in $\mathcal{D}(\mathbf{m}^p, t_u)$, again using Algorithm 1. Clearly, the second stage is only performed if $\mathcal{D}(\mathbf{m}^p, t_u)$ is non-empty. If $\mathcal{D}(\mathbf{m}^p, t_u)$ is empty, $\mathbf{m}^p$ along with all its children, $\mathcal{C}(\mathbf{m}^p)$, will be deleted from the current control invariant estimate. Throughout both stages of the elevation process, $\mathcal{T}(\mathcal{M}_i, N)$ will be updated to reflect all newly formed child-parent relations. At the end of the elevation round, both $min(\mathcal{M}_i)$ and $\mathcal{T}(\mathcal{M}_i, N)$ will be checked to remove any dismissed markings along with their children and to obtain the updated right-closed subset $\mathcal{M}_{i+1} \subseteq \mathcal{M}_0$. The Pseudocode of Algorithm 2 is presented in the Appendix section. The first illustrative example of Section 4 elaborates the implementation of the elevation principles discussed in Algorithm 1 and Algorithm 2.

In the next subsection, we present a procedure to compute (the minimal elements of) $\mathcal{M}_0^{\Uparrow}$ using a *Depth-First-Search* (DFS) strategy on an appropriately defined tree-structure.

3.2 A *Depth-First-Search* strategy for finding the supremal right-closed control invariant subset

For a right-closed set $\mathcal{M}_i \subseteq \mathcal{M}_0$, if $\mathcal{M}_i$ is not a control invariant set, multiple minimal elements of $\mathcal{M}_i$ may fail the CI condition test for multiple uncontrollable transitions. This implies that at the start of each elevation round, there might be multiple elevation choices to explore. In fact, at the start of an elevation round with $card(min(\mathcal{M}_i)) = k$ and $card(T_u) = l$, the combination of all possible computational choices to explore will be bounded by kl. In order to explore all possible computational choices, we introduce an algorithm based on the *Depth-First-Search* (DFS) strategy for a decision tree. In this tree structure, each node corresponds to a right-closed set represented by a finite set of minimal elements, where the root node is $min(\mathcal{M}_0)$. Considering the node for $min(\mathcal{M}_i)$, for each elevation process of a marking $\mathbf{m}^k \in min(\mathcal{M}_i)$ with respect to an uncontrollable transition $t_u \in T_u$, a branch with label $(\mathbf{m}^k, t_u, min(\mathcal{M}_i), \mathcal{T}(\mathcal{M}_i, N))$ is created down the node $min(\mathcal{M}_i)$. Starting from any node and given the elevation labels for each branch, the algorithm keeps exploring the path that originated from that node until a termination condition is reached for the path. In fact, for each leaf node of the tree such as $min(\mathcal{M}_f)$, we must have $\mathcal{A}^{fail}(\mathcal{M}_f) = \emptyset$ as the termination condition.

So, given an initial right-closed set $\mathcal{M}_0$, the algorithm for creating the decision tree and performing the *DFS*-based strategy for finding the supremal right-closed control invariant subset of $\mathcal{M}_0$ with respect to PN structure N, denoted by $\mathcal{M}_0^{\Uparrow}$, is briefly described as follows:

1. For the initial iteration, the root node of the decision tree represents the initial set $\mathcal{M}_0$ and its minimal elements. The minimal elements of $min(\mathcal{M}_0)$ are partitioned into $\mathcal{A}^{pass}(\mathcal{M}_0)$ and $\mathcal{A}^{fail}(\mathcal{M}_0)$:
   (a) If $\mathcal{A}^{fail}(\mathcal{M}_0)$ is empty, $\mathcal{M}_0$ is control invariant and the algorithm terminates with $\mathcal{A}^{pass}(\mathcal{M}_0)$ as the supremal right-closed control invariant subset $\mathcal{M}_0^{\Uparrow}$.
   (b) If $\mathcal{A}^{fail}(\mathcal{M}_0)$ is not empty, for each $\mathbf{m}^k \in \mathcal{A}^{fail}(\mathcal{M}_0)$ and for each $t_u \in \mathcal{CRT}(\mathbf{m}^k)$, a new branch is created with label $(\mathbf{m}^k, t_u, min(\mathcal{M}_0), \mathcal{T}(\mathcal{M}_0, N))$. At this point, by convention, the leftmost created branch is selected to be explored for the next elevation round. Each elevation round is implemented by calling Algorithm 2 with input $(\mathbf{m}^k, t_u, min(\mathcal{M}_0), \mathcal{T}(\mathcal{M}_0, N))$. Note that the output from Algorithm 2 will be an updated right-closed set, resulting in creation of a new node in the decision tree.

2. For the n^{th} iteration, suppose the algorithm picks a branch with label $(\mathbf{m}^k, t_u, min(\mathcal{M}_i), \mathcal{T}(\mathcal{M}_i, N))$. In this case, one round of elevation is performed, using algorithm 2, to obtain updated $min(\mathcal{M}_{n+1})$ and $\mathcal{T}(\mathcal{M}_{n+1}, N)$:

   (a) If $min(\mathcal{M}_{n+1})$ is empty, the path terminates with the empty set as a control invariant subset. At this point, the algorithm will backtrack to perform the next elevation round on the last leftmost unexplored branch, if any.

   (b) If $min(\mathcal{M}_{n+1})$ is not empty, $min(\mathcal{M}_{n+1})$ is partitioned into $\mathcal{A}^{pass}(\mathcal{M}_{n+1})$ and $\mathcal{A}^{fail}(\mathcal{M}_{n+1})$:

      i If $\mathcal{A}^{fail}(\mathcal{M}_{n+1})$ is empty, $\mathcal{M}_{n+1}$ is control invariant and the current path terminates with $\mathcal{A}^{pass}(\mathcal{M}_{n+1})$ as a control invariant subset and the algorithm will backtrack to perform the next elevation round on the last leftmost unexplored branch, if any.

      ii If $\mathcal{A}^{fail}(\mathcal{M}_{n+1})$ is not empty, for each $\mathbf{m}^k \in \mathcal{A}^{fail}(\mathcal{M}_{n+1})$ and for each $t_u \in \mathcal{CRT}(\mathbf{m}^k)$, a new branch is created with label $(\mathbf{m}^k, t_u, min(\mathcal{M}_{n+1}), \mathcal{T}(\mathcal{M}_{n+1}, N))$. At this point, the leftmost created branch is selected to be explored for the next elevation round.

Once no unexplored branches are left in the decision tree and all paths reach a termination condition, the algorithm will output the union of all right-closed control invariant subsets found at the termination-point of each path as $\mathcal{M}_0^{\Uparrow}$. This DFS-based elevation process is formally described in Algorithm 3 of the Appendix section. The second PN example of Section 4 illustrates the application of algorithms 3.

4 Illustrative examples

In this section we present a set of examples that illustrate the algorithms introduced in the previous section. The first example illustrates the finite-termination of the elevation process, resulting in a right-closed control invariant subset of the original right-closed set. The right-closed control invariant subset that results from this procedure is dependent on the choices made in due course of the algorithm. While the end-result is right-closed and control invariant (with respect to a PN structure), it is not guaranteed to be the supremal right-closed control invariant subset of the initial right-closed set – for this, one would need to record the control invariant subset arising from each possible choice of the elevation procedure, and present their union as the supremal right-closed control invariant subset. This is effectively done by the DFS-based procedure of Algorithm 3, illustrated in the second example. Regarding the practical application of the proposed algorithm, the third example presents a PN structure in a simplified manufacturing setting and highlights the importance of computation of the largest right-closed control invariant subset of an initial right-closed set, as a prerequisite and necessary step, for the procedure of design and synthesis of LESPs for PN models.

4.1 Example 1: Illustration of the elevation process

Consider the PN structure N_2 of Fig. 2a, and the right-closed set $min(\mathcal{M}_0) = \{(1\,0\,0\,0\,0)^T, (0\,2\,0\,0\,6)^T, (0\,0\,1\,0\,0)^T, (0\,0\,0\,6\,6)^T\}$. In the PN structure N_2 of Fig. 2a, all transitions except t_1 and t_6 are uncontrollable. Figure 2(b) shows the CI-graph $\mathcal{G}(\mathcal{M}_0, N_2)$; $\mathcal{A}^{pass}(\mathcal{M}_0) = \{(1\,0\,0\,0\,0)^T, (0\,0\,0\,6\,6)^T\}$. Also, note that there are no outgoing arcs labeled

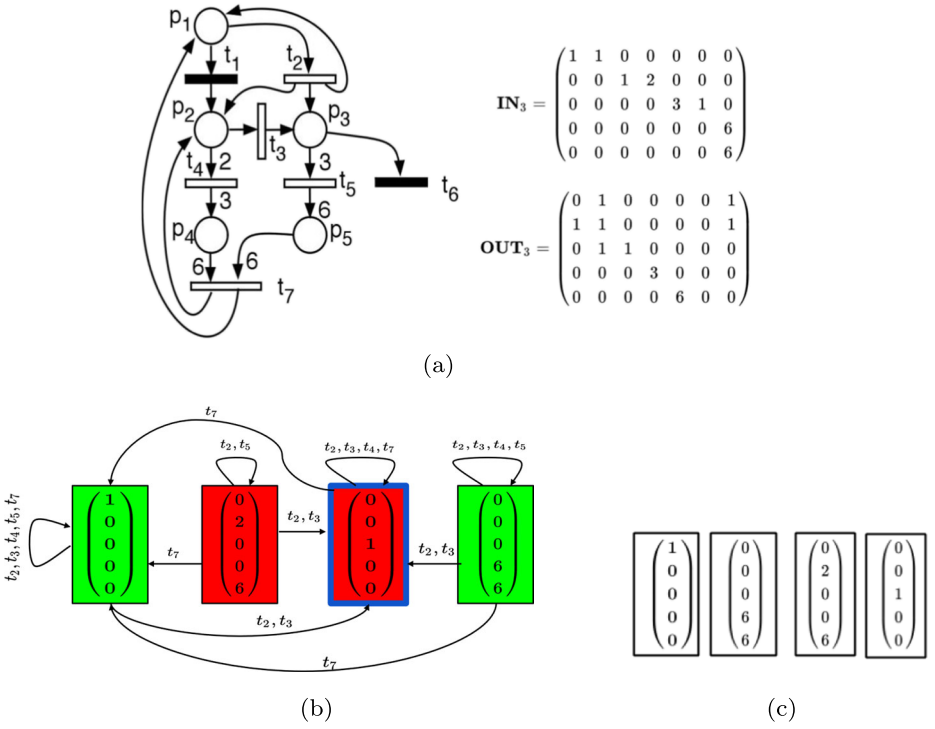

Fig. 2 The PN structure N_2, the CI-graph of $\mathcal{M}_0$ with respect to N_2, and the child-parent tree structure of $\mathcal{M}_0$, where $min(\mathcal{M}_0) = \{(1\ 0\ 0\ 0\ 0)^T, (0\ 2\ 0\ 0\ 6)^T, (0\ 0\ 1\ 0\ 0)^T, (0\ 0\ 0\ 6\ 6)^T\}$, and $\mathcal{A}^{pass}(\mathcal{M}_0) = \{(1\ 0\ 0\ 0\ 0)^T, (0\ 0\ 0\ 6\ 6)^T\}$. **a** N_2 **b** $\mathcal{G}(\mathcal{M}_0, N_2)$ **c** $\mathcal{T}(\mathcal{M}_0, N_2)$

with t_4 and t_5 for minimal elements $(0\ 2\ 0\ 0\ 6)^T$ and $(0\ 0\ 1\ 0\ 0)^T$; therefore, $(0\ 2\ 0\ 0\ 6)^T$ and $(0\ 0\ 1\ 0\ 0)^T$ fail the CI test for transitions t_4 and t_5, respectively. Figure 2c shows the initial child-parent tree collection/structure $\mathcal{T}(\mathcal{M}_0, N_2)$, where all markings of $min(\mathcal{M}_0)$ are considered as root vertices and the tree has no arcs since no child-parent relation is formed yet.

The elevation process can start with either one of the minimal elements that failed the CI test (viz. $(0\ 2\ 0\ 0\ 6)^T$ or $(0\ 0\ 1\ 0\ 0)^T$), and this elevation can be done with respect to either one of the uncontrollable transitions in the set $\{t_4, t_5\}$. Suppose we start by elevating the minimal element $(0\ 0\ 1\ 0\ 0)^T$ for transition t_5 with respect to all members of $(min(\mathcal{M}_0) - \{(0\ 0\ 1\ 0\ 0)^T\} = \{(1\ 0\ 0\ 0\ 0)^T, (0\ 2\ 0\ 0\ 6)^T, (0\ 0\ 0\ 6\ 6)^T\})$. If we perform the elevation process once, we will get the new markings (*descendants*), $\varepsilon_0 = \{(1\ 0\ 1\ 0\ 0)^T, (0\ 0\ 1\ 6\ 0)^T, (0\ 2\ 1\ 0\ 0)^T\}$; it is clear that marking $(1\ 0\ 1\ 0\ 0)^T$ is term-wise larger than minimal element $(1\ 0\ 0\ 0\ 0)^T$ and, therefore, it can be discarded. So, we update ε_0 to $\varepsilon_0 = \{(0\ 0\ 1\ 6\ 0)^T, (0\ 2\ 1\ 0\ 0)^T\}$. Now, to complete the elevation process, $(0\ 0\ 1\ 0\ 0)^T$ needs to be elevated with respect to members of ε_0, resulting in $\varepsilon_1 = \{(0\ 0\ 4\ 6\ 0)^T, (0\ 2\ 4\ 0\ 0)^T\}$; all members of ε_1 are dismissed by members of ε_0. Therefore, the first round of elevation will terminate, and the updated estimate is

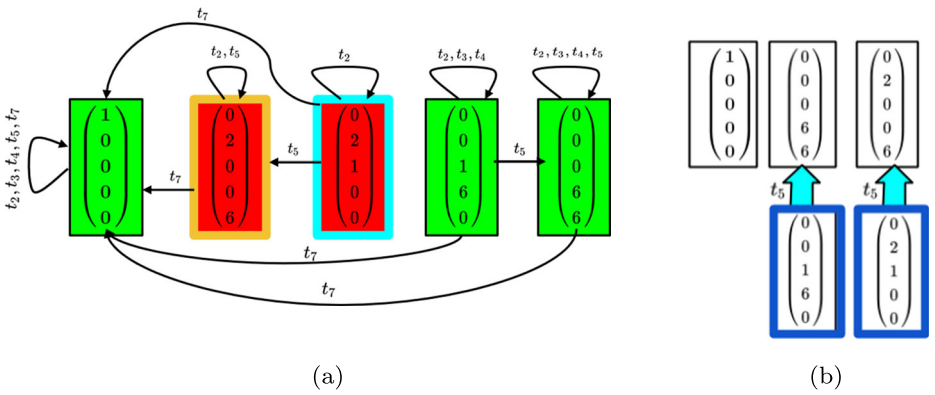

Fig. 3 The CI-graph and the child-parent tree structure of $\widehat{\mathcal{M}}_1$, where $min(\widehat{\mathcal{M}}_1) = \{(1\ 0\ 0\ 0\ 0)^T, (0\ 2\ 0\ 0\ 6)^T, (0\ 0\ 0\ 6\ 6)^T, (0\ 2\ 1\ 0\ 0)^T, (0\ 0\ 1\ 6\ 0)^T\}$, and $\mathcal{A}^{pass}(\widehat{\mathcal{M}}_1) = \{(1\ 0\ 0\ 0\ 0)^T, (0\ 0\ 0\ 6\ 6)^T, (0\ 0\ 1\ 6\ 0)^T\}$. **a** $(\widehat{\mathcal{M}}_1, N_2)$ **b** $\mathcal{T}(\widehat{\mathcal{M}}_1, N_2)$

$min(\widehat{\mathcal{M}}_1) = \{(1\ 0\ 0\ 0\ 0)^T, (0\ 2\ 0\ 0\ 6)^T, (0\ 0\ 0\ 6\ 6)^T, (0\ 2\ 1\ 0\ 0)^T, (0\ 0\ 1\ 6\ 0)^T\}$. The CI-graph $\mathcal{G}(\widehat{\mathcal{M}}_1, N_2)$ and the child-parent tree structure $\mathcal{T}(\widehat{\mathcal{M}}_1, N_2)$ are shown in Fig. 3. Note that the thick blue arrows of Fig. 3b shows the existing *child-parent* relations.

From Fig. 3a, it is clear that both markings $\{(0\ 2\ 1\ 0\ 0)^T, (0\ 2\ 0\ 0\ 6)^T\}$ fail the CI test for transition t_3 and t_4, as there is no outgoing arcs for either of these two markings which is labeled with t_3 and t_4. Our proposed algorithm dictates two rules; based on the first rule, $(0\ 2\ 0\ 0\ 6)^T$ should not be elevated with respect to its children that is, $(0\ 2\ 0\ 0\ 6)^T$ is not elevated with respect to $(0\ 2\ 1\ 0\ 0)^T$. The second rule states that each time a parent marking is elevated and replaced by some non-dismissed descendants, all children of that parent should also be elevated accordingly with respect to the new descendants. However, if the descendants of the parent marking are all dismissed and therefore, deleted, the children should be deleted, as well. We apply the two mentioned guidelines to this example. Let's assume we decide to elevate the parent marking $(0\ 2\ 0\ 0\ 6)^T$ for transition t_4; in fact, $(0\ 2\ 0\ 0\ 6)^T$ needs to be raised with respect to the members of $(min(\widehat{\mathcal{M}}_1) - \{(0\ 2\ 0\ 0\ 6)^T, (0\ 2\ 1\ 0\ 0)^T\} = \{(1\ 0\ 0\ 0\ 0)^T, (0\ 0\ 0\ 6\ 6)^T, (0\ 0\ 1\ 6\ 0)^T\})$. If we perform the elevation process once, we will get the new markings (*descendants*), $\varepsilon_0 = \{(1\ 2\ 0\ 0\ 6)^T, (0\ 2\ 0\ 3\ 6)^T, (0\ 2\ 1\ 3\ 6)^T\}$; it is clear that markings $(1\ 2\ 0\ 0\ 6)^T$ and $(0\ 2\ 1\ 3\ 6)^T$ are term-wise larger than minimal elements $(1\ 0\ 0\ 0\ 0)^T$ and $(0\ 2\ 0\ 3\ 6)^T$ respectively; therefore, we remove these two minimal elements to update ε_0 to $\varepsilon_0 = \{(0\ 2\ 0\ 3\ 6)^T\}$. Now, to complete the elevation process, $(0\ 2\ 0\ 0\ 6)^T$ needs to be elevated with respect to members of ε_0, resulting in $\varepsilon_1 = \{(0\ 4\ 0\ 0\ 6)^T\}$. We repeat the elevation process once more to raise $(0\ 2\ 0\ 0\ 6)^T$ with respect to members of ε_1 to get $\varepsilon_2 = \{(0\ 6\ 0\ 0\ 6)^T\}$. As we can see the only marking of ε_2 is term-wise larger than the only marking of ε_1. So, the final set of descendants for marking $(0\ 2\ 0\ 0\ 6)^T$ will be $\{(0\ 2\ 0\ 3\ 6)^T, (0\ 4\ 0\ 0\ 6)^T\}$. Now, in the next step, we need to raise the child(ren) of $(0\ 2\ 0\ 0\ 6)^T$ with respect to the set of new descendants. As shown in Fig. 3b, $(0\ 2\ 1\ 0\ 0)^T$ is the only child of $(0\ 2\ 0\ 0\ 6)^T$ connected through transition t_5. Therefore, $(0\ 2\ 1\ 0\ 0)^T$ needs to be checked and elevated (if necessary) with respect to members of $\{(0\ 2\ 0\ 3\ 6)^T, (0\ 4\ 0\ 0\ 6)^T\}$ for t_5. This process will result in the set of new markings $\varepsilon_{01} = \{(0\ 2\ 1\ 3\ 0)^T, (0\ 4\ 1\ 0\ 0)^T\}$. Next, to complete the elevation process, $(0\ 2\ 1\ 0\ 0)^T$ is elevated with respect to members of ε_{01}, resulting in $\varepsilon_{11} = \{(0\ 2\ 4\ 3\ 0)^T, (0\ 4\ 4\ 0\ 0)^T\}$; all members of ε_{11} are dismissed by members of ε_{01}, and are discarded. Therefore, the

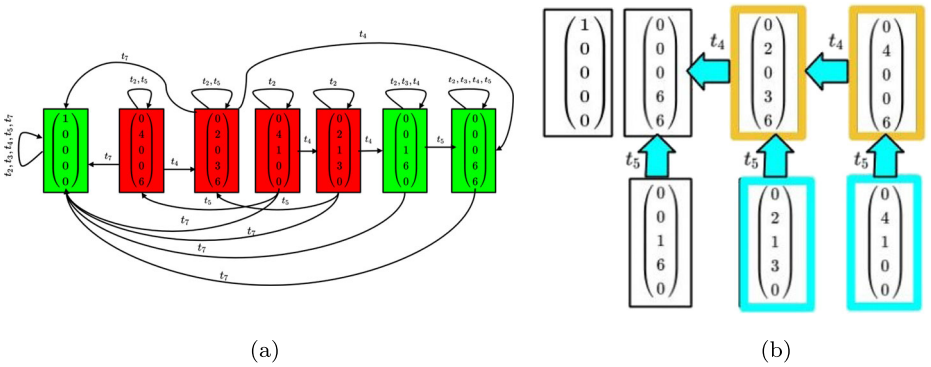

Fig. 4 The CI-graph and the child-parent tree structure of $\widehat{\mathcal{M}}_2$, where $min(\widehat{\mathcal{M}}_2) = \{(1\,0\,0\,0\,0)^T, (0\,0\,0\,6\,6)^T, (0\,0\,1\,6\,0)^T, (0\,2\,0\,3\,6)^T, (0\,4\,0\,0\,6)^T, (0\,2\,1\,3\,0)^T, (0\,4\,1\,0\,0)^T\}$, and $\mathcal{A}^{pass}(\widehat{\mathcal{M}}_2) = \{(1\,0\,0\,0\,0)^T, (0\,0\,0\,6\,6)^T, (0\,0\,1\,6\,0)^T\}$. **a** $\mathcal{G}(\widehat{\mathcal{M}}_2, N_2)$ **b** $\mathcal{T}(\widehat{\mathcal{M}}_2, N_2)$

second round of elevation will terminate, and the updated estimate will be $min(\widehat{\mathcal{M}}_2) = \{(1\,0\,0\,0\,0)^T, (0\,0\,0\,6\,6)^T, (0\,0\,1\,6\,0)^T, (0\,2\,0\,3\,6)^T, (0\,4\,0\,0\,6)^T, (0\,2\,1\,3\,0)^T, (0\,4\,1\,0\,0)^T\}$. The CI-graph $\mathcal{G}(\widehat{\mathcal{M}}_2, N_2)$ and the child-parent tree structure $\mathcal{T}(\widehat{\mathcal{M}}_2, N_2)$ are shown in Fig. 4.

Based on Fig. 4a, markings $\{(0\,2\,0\,3\,6)^T, (0\,4\,0\,0\,6)^T, (0\,2\,1\,3\,0)^T, (0\,4\,1\,0\,0)^T\}$ fails the CI test for transitions t_3. Suppose we decide to elevate $(0\,2\,0\,3\,6)^T$ for transition t_3. Given the rules of the algorithm, $(0\,2\,0\,3\,6)^T$ should be raised with respect to members of $(min(\widehat{\mathcal{M}}_2) - \{(0\,2\,0\,3\,6)^T, (0\,2\,1\,3\,0)^T, (0\,4\,0\,0\,6)^T, (0\,4\,1\,0\,0)^T\} = \{(1\,0\,0\,0\,0)^T, (0\,0\,0\,6\,6)^T, (0\,0\,1\,6\,0)^T\})$. This process will result in the generation of the set $\varepsilon_0 = \{(1\,2\,0\,3\,6)^T, (0\,2\,0\,6\,6)^T, (0\,2\,0\,6\,6)^T\}$, all being term-wise larger than members of $(min(\widehat{\mathcal{M}}_2) - \{(0\,2\,0\,3\,6)^T, (0\,2\,1\,3\,0)^T, (0\,4\,0\,0\,6)^T, (0\,4\,1\,0\,0)^T\}$. At this point, since $(0\,2\,0\,3\,6)^T$ has no non-dismissed descendants, we need to remove it from the current estimate; upon removal of $(0\,2\,0\,3\,6)^T$, all its children, i.e. $\{(0\,2\,1\,3\,0)^T, (0\,4\,0\,0\,6)^T, (0\,4\,1\,0\,0)^T\}$, will also get removed based on the discipline dictated by the algorithm. Therefore, the third round of elevation will terminate, and the updated estimate will be $min(\widehat{\mathcal{M}}_3) = \{(1\,0\,0\,0\,0)^T, (0\,0\,0\,6\,6)^T, (0\,0\,1\,6\,0)^T\}$. The CI-graph $\mathcal{G}(\widehat{\mathcal{M}}_3, N_2)$ and the child-parent tree collection $\mathcal{T}(\widehat{\mathcal{M}}_3, N_2)$ are shown in Fig. 5. Note that $\widehat{\mathcal{M}}_3$ is control invariant with respect to PN structure N_2 as we have $\mathcal{A}^{fail}(\widehat{\mathcal{M}}_3) = \emptyset$.

For this example, the process of elevation resulted in a right-closed set $\widehat{\mathcal{M}}_3 \subset \mathcal{M}_0$ that is control invariant (with respect to the PN structure N_2) that is also the supremal right-closed control invariant subset $\mathcal{M}_0^{\Uparrow}$. In general, the control invariant set that results from this process of repeated elevation is not guaranteed to be the supremal right-closed control invariant subset. The example in the next subsection illustrates the need for the DFS-based procedure of Algorithm 3.

4.2 Example 2: DFS-based approach for computing the supremal right-closed control invariant subset

Consider the PN structure N_3 of Fig. 6a, and the right-closed set $min(\mathcal{M}_0) = \{(1\,0\,0\,0\,0)^T, (0\,1\,0\,0\,0)^T, (0\,0\,1\,0\,0)^T, (0\,0\,0\,1\,1)^T\}$. In the PN structure N_3 of Fig. 6a, all transitions are uncontrollable. Figure 6(b) shows the CI-graph $\mathcal{G}(\mathcal{M}_0, N_3)$; as we can see from the graph, $\mathcal{A}^{pass}(\mathcal{M}_0)$ is empty. Figure 6(c) shows the initial child-parent tree structure $\mathcal{T}(\mathcal{M}_0, N_3)$,

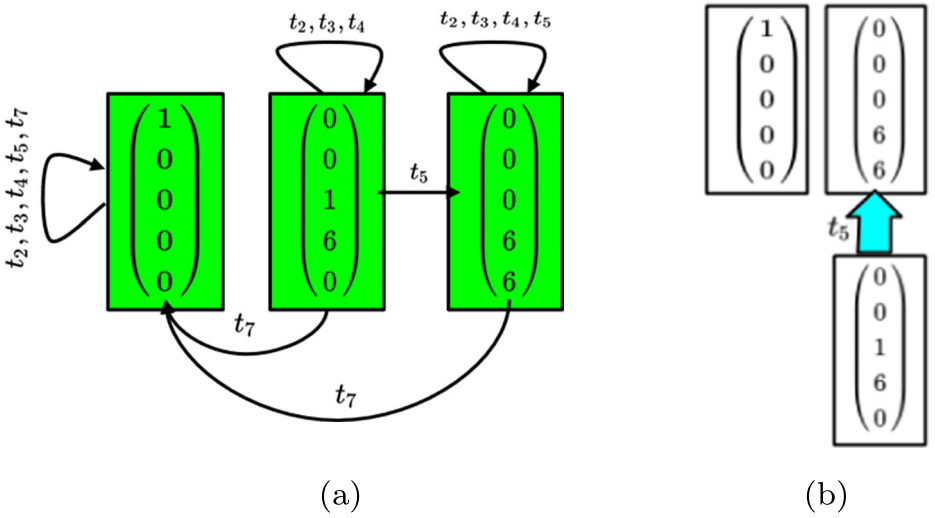

Fig. 5 The CI-graph and the child-parent tree structure of $\widehat{\mathcal{M}}_3$, where $min(\widehat{\mathcal{M}}_3) = \{(1\ 0\ 0\ 0\ 0)^T, (0\ 0\ 0\ 6\ 6)^T, (0\ 0\ 1\ 6\ 0)^T\}$, and $\mathcal{A}^{pass}(\widehat{\mathcal{M}}_3) = \{(1\ 0\ 0\ 0\ 0)^T, (0\ 0\ 0\ 6\ 6)^T, (0\ 0\ 1\ 6\ 0)^T\}$. **a** $\mathcal{G}(\widehat{\mathcal{M}}_3, N_2)$ **b** $\mathcal{T}(\widehat{\mathcal{M}}_3, N_2)$

where all markings of $min(\mathcal{M}_0)$ are considered as root vertices and the tree has no arcs since no child-parent relation is formed yet.

As we can see from Fig. 6b, markings $(1\ 0\ 0\ 0\ 0)^T$ and $(0\ 1\ 0\ 0\ 0)^T$ fail the CI test for transitions t_1 and t_3, respectively. So, there will be two elevation choices to start from. The first one is labeled as $((1\ 0\ 0\ 0\ 0)^T, t_1, min(\mathcal{M}_0), \mathcal{T}(\mathcal{M}_0, N_3))$, which corresponds

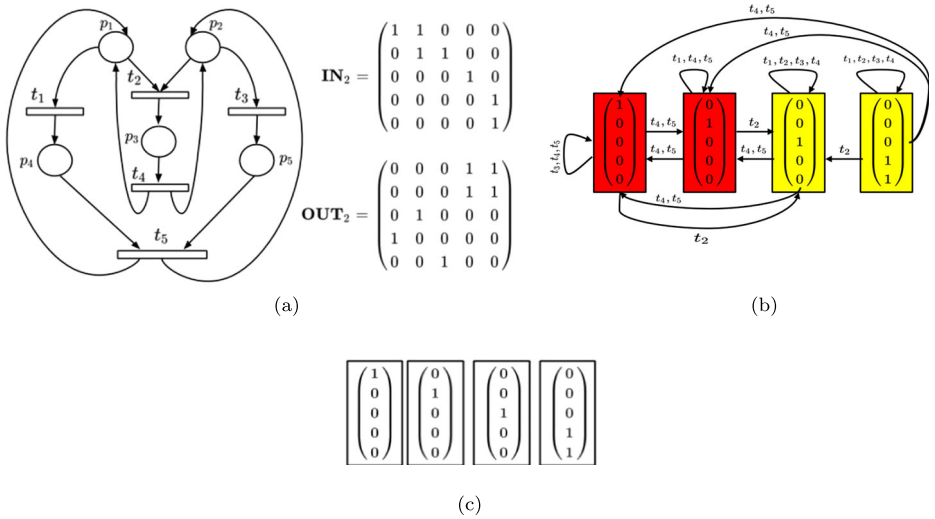

Fig. 6 The PN structure N_3, the CI-graph of $\mathcal{M}_0$ with respect to N_3, and the child-parent tree structure of $\mathcal{M}_0$, where $min(\mathcal{M}_0) = \{(1\ 0\ 0\ 0\ 0)^T, (0\ 1\ 0\ 0\ 0)^T, (0\ 0\ 1\ 0\ 0)^T, (0\ 0\ 0\ 1\ 1)^T\}$, and $\mathcal{A}_{pass}(\mathcal{M}_0) = \emptyset$. **a** N_3 **b** $\mathcal{G}(\mathcal{M}_0, N_3)$ **c** $\mathcal{T}(\mathcal{M}_0, N_3)$

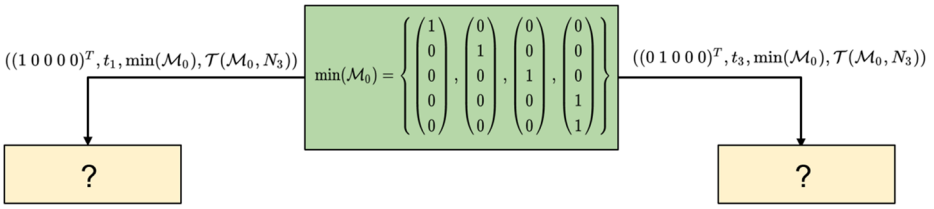

Fig. 7 Summary of different elevation choices starting from $\mathcal{M}_0$, where $min(\mathcal{M}_0) = \{(1\,0\,0\,0\,0)^T, (0\,1\,0\,0\,0)^T, (0\,0\,1\,0\,0)^T, (0\,0\,0\,1\,1)^T\}$

to elevation of $(1\ 0\ 0\ 0\ 0)^T$ with respect to t_1 given $min(\mathcal{M}_0)$ and $\mathcal{T}(\mathcal{M}_0, N_3)$. The second one is labeled as $((0\ 1\ 0\ 0\ 0)^T, t_3, min(\mathcal{M}_0), \mathcal{T}(\mathcal{M}_0, N_3))$, which corresponds to elevation of $(0\ 1\ 0\ 0\ 0)^T$ with respect to t_3 given $min(\mathcal{M}_0)$ and $\mathcal{T}(\mathcal{M}_0, N_3)$. Figure 7 provides a schematic decision tree showing the initial elevation choices for the given example, where each edge label corresponds to a specific elevation choice. This decision tree will get updated as we go through different elevation rounds of the algorithm.

Starting from $((1\ 0\ 0\ 0\ 0)^T, t_1, min(\mathcal{M}_0)), \mathcal{T}(\mathcal{M}_0, N_3))$, first we elevate $(1\ 0\ 0\ 0\ 0)^T$ for t_1 with respect to $(0\ 1\ 0\ 0\ 0)^T$ to get $(1\ 1\ 0\ 0\ 0)^T$, which is term-wise larger than $(0\ 1\ 0\ 0\ 0)^T$; so, it will get deleted. Next, we raise $(1\ 0\ 0\ 0\ 0)^T$ for t_1 with respect to $(0\ 0\ 1\ 0\ 0)^T$ to get $(1\ 0\ 1\ 0\ 0)^T$, which is term-wise larger than $(0\ 0\ 1\ 0\ 0)^T$ and gets deleted. Finally, $(1\ 0\ 0\ 0\ 0)^T$ is elevated for t_1 with respect to $(0\ 0\ 0\ 1\ 1)^T$ resulting in $(1\ 0\ 0\ 0\ 1)^T$, which is not term-wise larger than $(0\ 0\ 0\ 1\ 1)^T$; so, $(1\ 0\ 0\ 0\ 1)^T$ is a *child* of the *parent* marking $(0\ 0\ 0\ 1\ 1)^T$. At this point, we also need to raise $(1\ 0\ 0\ 0\ 0)^T$ for t_1 with respect to the newly generated marking, $(1\ 0\ 0\ 0\ 1)^T$; this results in generation of the marking $(2\ 0\ 0\ 0\ 1)^T$ which is term-wise larger than $(1\ 0\ 0\ 0\ 1)^T$ and is therefore, deleted. So, at the end of this elevation round, we get the updated estimate $min(\widehat{\mathcal{M}}_1) = \{(0\ 1\ 0\ 0\ 0)^T, (0\ 0\ 1\ 0\ 0)^T, (0\ 0\ 0\ 1\ 1)^T, (1\ 0\ 0\ 0\ 1)^T\}$. The CI-graph $\mathcal{G}(\widehat{\mathcal{M}}_1, N_3)$ and the child-parent tree structure $\mathcal{T}(\widehat{\mathcal{M}}_1, N_3)$ is shown in Fig. 8, where $\mathcal{A}^{pass}(\widehat{\mathcal{M}}_1) = \emptyset$. Note that the thick blue arrows of Fig. 8b shows the existing *child-parent* relations.

As we can see from Fig. 8a, marking $(0\ 1\ 0\ 0\ 0)^T$ fails the CI test for transitions t_3. So, at this point, there will be only one elevation choice to perform, which is labeled as $((0\ 1\ 0\ 0\ 0)^T, t_3, min(\widehat{\mathcal{M}}_1), \mathcal{T}(\widehat{\mathcal{M}}_1, N_3))$. If $(0\ 1\ 0\ 0\ 0)^T$ is elevated for t_3 with respect to $(0\ 0\ 1\ 0\ 0)^T$, the resulting marking will be $(0\ 1\ 1\ 0\ 0)^T$, which is term-wise larger than $(0\ 0\ 1\ 0\ 0)^T$; so, it will be discarded. Next, we would raise $(0\ 1\ 0\ 0\ 0)^T$ for t_3 with respect to $(0\ 0\ 0\ 1\ 1)^T$ to get $(0\ 1\ 0\ 1\ 0)^T$, which is not term-wise larger than $(0\ 0\ 0\ 1\ 1)^T$; so, $(0\ 1\ 0\ 1\ 0)^T$ is a *child* of the *parent* marking $(0\ 0\ 0\ 1\ 1)^T$. Since we have a new non-dismissed minimal element, we also need to elevate $(0\ 1\ 0\ 0\ 0)^T$ with respect to $(0\ 1\ 0\ 1\ 0)^T$ for t_3, resulting in $(0\ 2\ 0\ 1\ 0)^T$ which is term-wise larger than $(0\ 1\ 0\ 1\ 0)^T$ and is therefore, discarded. Finally, $(0\ 1\ 0\ 0\ 0)^T$ is raised with respect to $(1\ 0\ 0\ 0\ 1)^T$ for t_3; it results in generation of $(1\ 1\ 0\ 0\ 0)^T$, which is not term-wise larger than $(1\ 0\ 0\ 0\ 1)^T$; so, $(1\ 1\ 0\ 0\ 0)^T$ is a *child* of the *parent* marking $(1\ 0\ 0\ 0\ 1)^T$. Again, $(0\ 1\ 0\ 0\ 0)^T$ needs to be raised with respect to the newly generated marking, $(1\ 1\ 0\ 0\ 0)^T$; the resulting marking will be $(1\ 2\ 0\ 0\ 0)^T$ which is term-wise larger than $(1\ 1\ 0\ 0\ 0)^T$ and is therefore, deleted. At the end of this elevation round, we get the updated estimate $min(\widehat{\mathcal{M}}_2) = \{(0\ 0\ 1\ 0\ 0)^T, (0\ 0\ 0\ 1\ 1)^T, (1\ 0\ 0\ 0\ 1)^T, (0\ 1\ 0\ 1\ 0)^T, (1\ 1\ 0\ 0\ 0)^T\}$. The CI-graph $\mathcal{G}(\widehat{\mathcal{M}}_2, N_3)$ and the child-parent tree structure $\mathcal{T}(\widehat{\mathcal{M}}_2, N_3)$ is shown in Fig. 9, where

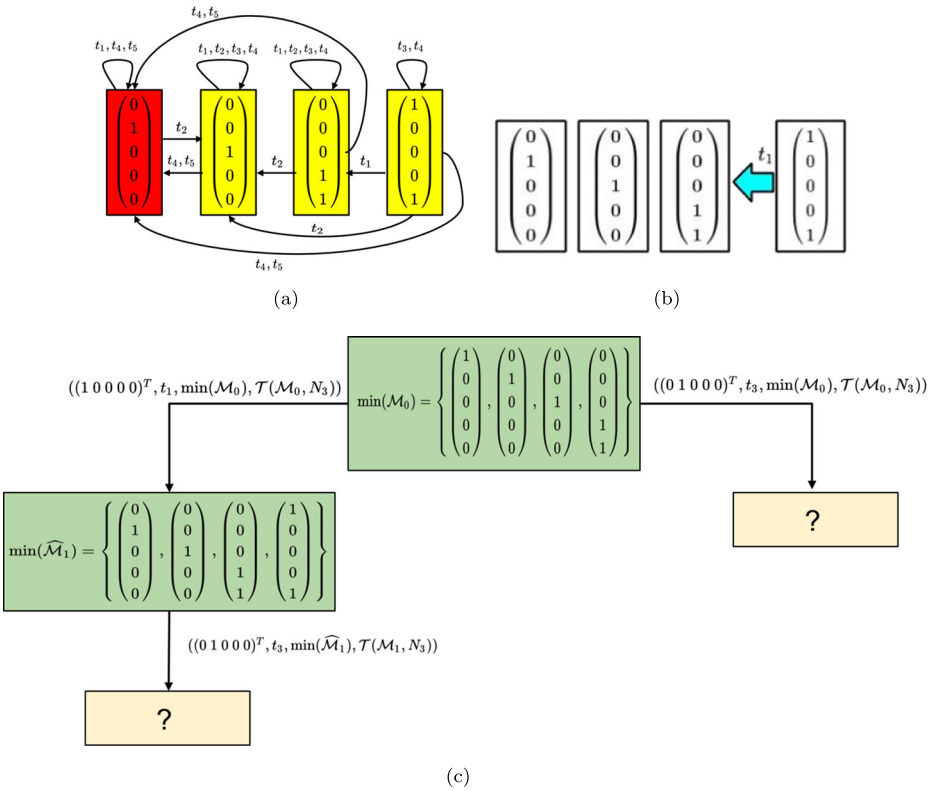

Fig. 8 The CI-graph and the child-parent tree structure of $\widehat{\mathcal{M}}_1$, where $min(\widehat{\mathcal{M}}_1) = \{(0\,1\,0\,0\,0)^T, (0\,0\,1\,0\,0)^T, (0\,0\,0\,1\,1)^T, (1\,0\,0\,0\,1)^T\}$, and $\mathcal{A}_{pass}(\widehat{\mathcal{M}}_1) = \emptyset$. **a** $\mathcal{G}(\widehat{\mathcal{M}}_1, N_3)$ **b** $\mathcal{T}(\widehat{\mathcal{M}}_1, N_3)$ **c** Current Decision Tree of the DFS algorithm

$\mathcal{A}^{fail}(\widehat{\mathcal{M}}_2) = \emptyset$, implying that $\widehat{\mathcal{M}}_2$ is control invariant. Therefore, this path terminates with $\widehat{\mathcal{M}}_2$ as the first control invariant subset found for $\mathcal{M}_0$.

Now, we go back to the last unexplored elevation choice; as shown in Fig. 9c, this choice is labeled as $((0\,1\,0\,0\,0)^T, t_3, min(\mathcal{M}_0), \mathcal{T}(\mathcal{M}_0, N_3))$, corresponding to elevation of $(0\,1\,0\,0\,0)^T$ with respect to t_3 given $min(\mathcal{M}_0)$ and $\mathcal{T}(\mathcal{M}_0, N_3)$ as shown in Fig. 6. If $(0\,1\,0\,0\,0)^T$ is elevated for transition t_3 with respect to $(1\,0\,0\,0\,0)^T$, we get $(1\,1\,0\,0\,0)^T$, which is term-wise larger than $(1\,0\,0\,0\,0)^T$; so, it will get deleted. Next, we would raise $(0\,1\,0\,0\,0)^T$ for t_3 with respect to $(0\,0\,1\,0\,0)^T$ to get $(0\,1\,1\,0\,0)^T$, which is term-wise larger than $(0\,0\,1\,0\,0)^T$ and gets deleted. Finally, $(0\,1\,0\,0\,0)^T$ is raised for t_3 with respect to $(0\,0\,0\,1\,1)^T$, resulting in $(0\,1\,0\,1\,0)^T$, which is not dismissed by $(0\,0\,0\,1\,1)^T$; so, $(0\,1\,0\,1\,0)^T$ is a *child* of the *parent* marking $(0\,0\,0\,1\,1)^T$. At this point, we also need to elevate $(0\,1\,0\,0\,0)^T$ for transition t_3 with respect to the newly generated marking $(0\,1\,0\,1\,0)^T$, which results in $(0\,2\,0\,1\,0)^T$ and this marking is term-wise larger than $(0\,1\,0\,1\,0)^T$ and is therefore, discarded. At the end of this elevation round, we get the updated estimate $min(\widehat{\mathcal{M}}_3) = \{(1\,0\,0\,0\,0)^T, (0\,0\,1\,0\,0)^T, (0\,0\,0\,1\,1)^T, (0\,1\,0\,1\,0)^T\}$. The CI-graph $\mathcal{G}(\widehat{\mathcal{M}}_3, N_3)$ and the child-parent tree structure $\mathcal{T}(\widehat{\mathcal{M}}_3, N_3)$ is shown in Fig. 10, where $\mathcal{A}^{pass}(\widehat{\mathcal{M}}_3) = \emptyset$.

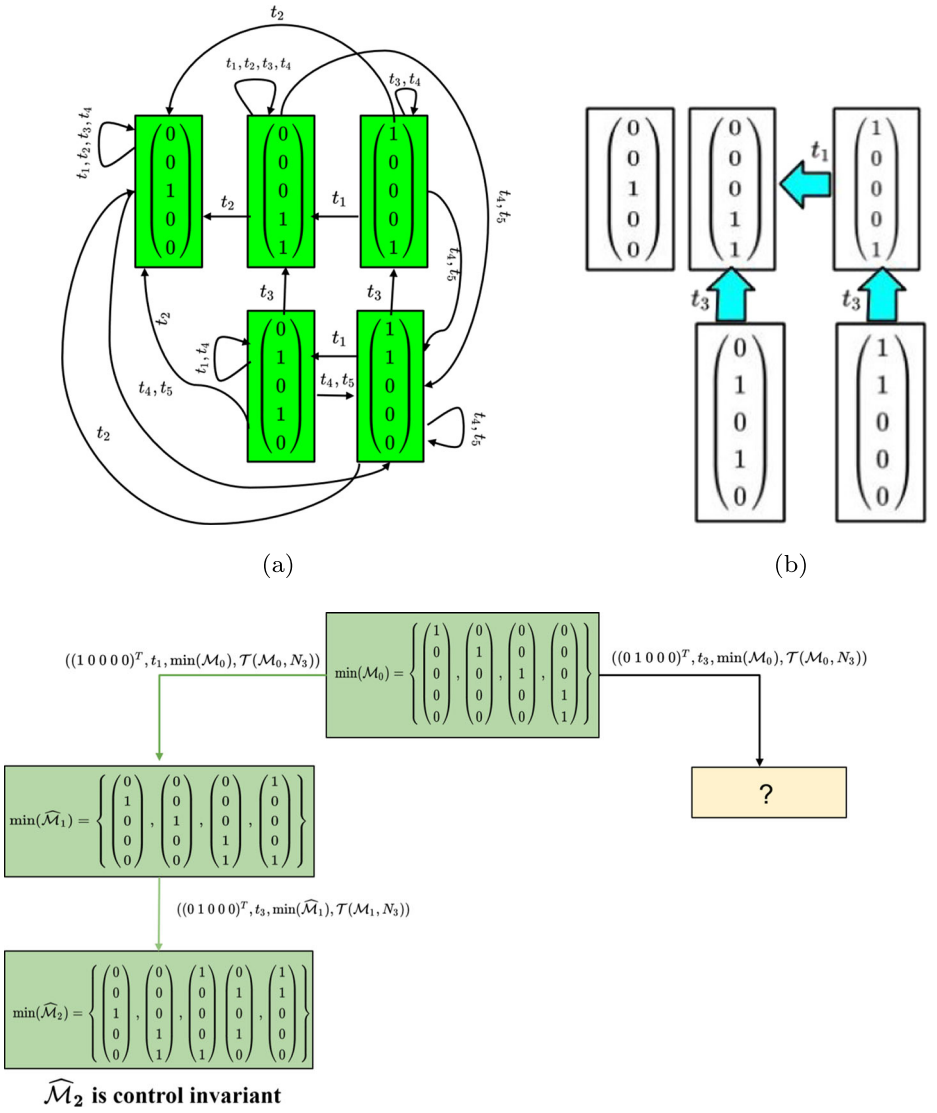

Fig. 9 The CI-graph and the child-parent tree structure of $\widehat{\mathcal{M}}_2$, where $min(\widehat{\mathcal{M}}_2) = \{(0\,0\,1\,0\,0)^T, (0\,0\,0\,1\,1)^T, (1\,0\,0\,0\,1)^T, (0\,1\,0\,1\,0)^T, (1\,1\,0\,0\,0)^T\}$, and $\mathcal{A}_{fail}(\widehat{\mathcal{M}}_2) = \emptyset$. **a** $\mathcal{G}(\widehat{\mathcal{M}}_2, N_3)$ **b** $\mathcal{T}(\widehat{\mathcal{M}}_2, N_3)$ **c** Current Decision Tree of the DFS algorithm

As we can see from Fig. 10a, marking $(1\ 0\ 0\ 0\ 0)^T$ fails the CI test for transitions t_1. So, at this point, there will be only one elevation choice to perform, which is labeled as $((1\ 0\ 0\ 0\ 0)^T, t_1, min(\widehat{\mathcal{M}}_3)), \mathcal{T}(\widehat{\mathcal{M}}_3, N_3))$. If $(1\ 0\ 0\ 0\ 0)^T$ is elevated for transition t_1 with respect to $(0\ 0\ 1\ 0\ 0)^T$, it results in the marking $(1\ 0\ 1\ 0\ 0)^T$, which is term-wise larger than $(0\ 0\ 1\ 0\ 0)^T$; so, it will get deleted. Next, $(1\ 0\ 0\ 0\ 0)^T$ is elevated for transition t_1

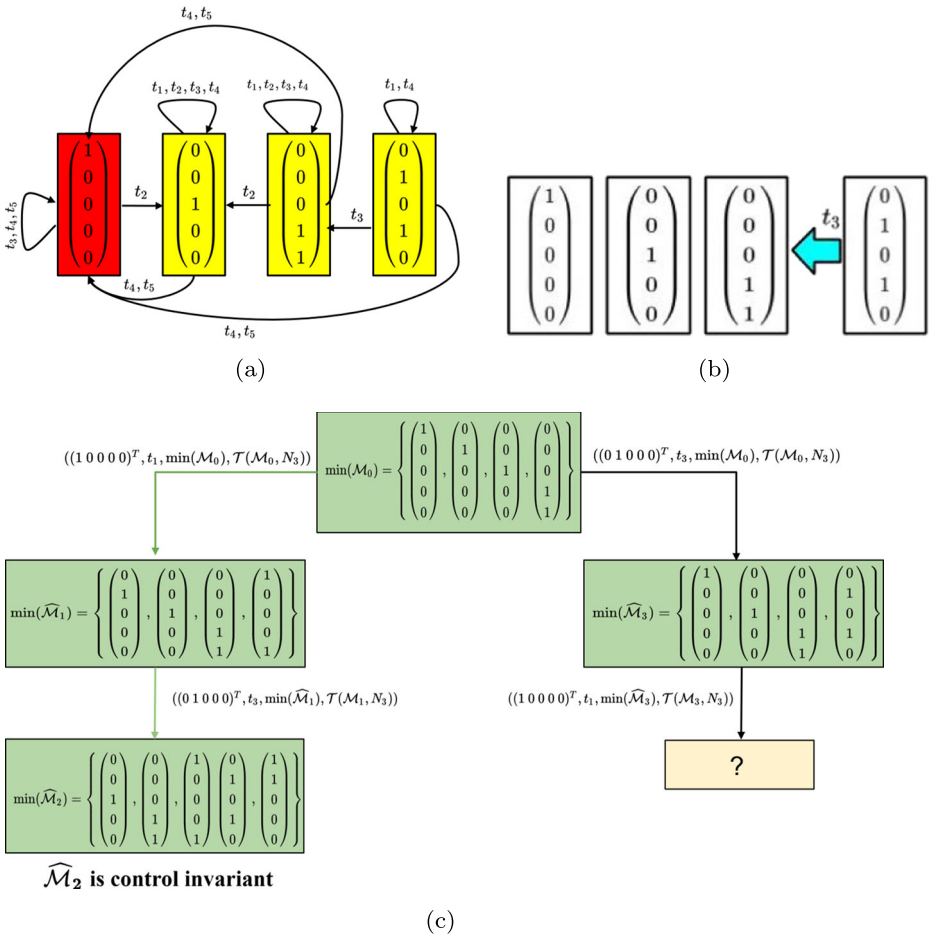

Fig. 10 The CI-graph and the child-parent tree structure of $\widehat{\mathcal{M}}_3$, where $min(\widehat{\mathcal{M}}_3) = \{(1\,0\,0\,0\,0)^T, (0\,0\,1\,0\,0)^T, (0\,0\,0\,1\,1)^T, (0\,1\,0\,1\,0)^T\}$, and $\mathcal{A}_{pass}(\widehat{\mathcal{M}}_3) = \emptyset$. **a** $\mathcal{G}(\widehat{\mathcal{M}}_3, N_3)$ **b** $\mathcal{T}(\widehat{\mathcal{M}}_3, N_3)$ **c** Current Decision Tree of the DFS algorithm

with respect to $(0\,0\,0\,1\,1)^T$ to get $(1\,0\,0\,0\,1)^T$; this marking is not term-wise larger than $(0\,0\,0\,1\,1)^T$; so, $(1\,0\,0\,0\,1)^T$ is a *child* of the *parent* marking $(0\,0\,0\,1\,1)^T$. At this point, $(1\,0\,0\,0\,0)^T$ needs to be elevated for transition t_1 with respect to the newly generated marking $(1\,0\,0\,0\,1)^T$, resulting in $(2\,0\,0\,0\,1)^T$ which is term-wise larger than $(1\,0\,0\,0\,1)^T$ and is therefore, discarded. Finally, we raise $(1\,0\,0\,0\,0)^T$ for t_1 with respect to $(0\,1\,0\,1\,0)^T$ to get $(1\,1\,0\,0\,0)^T$, which is not term-wise larger than $(0\,1\,0\,1\,0)^T$; so, $(1\,1\,0\,0\,0)^T$ is a *child* of the *parent* marking $(0\,1\,0\,1\,0)^T$. If $(1\,0\,0\,0\,0)^T$ is raised with respect to the newly generated marking $(1\,1\,0\,0\,0)^T$, the resulting marking $(2\,1\,0\,0\,0)^T$ will be dismissed by $(1\,1\,0\,0\,0)^T$ and is therefore, deleted. At the end of this elevation round, we get the updated estimate $min(\widehat{\mathcal{M}}_4) = \{(0\,0\,1\,0\,0)^T, (0\,0\,0\,1\,1)^T, (0\,1\,0\,1\,0)^T, (1\,0\,0\,0\,1)^T, (1\,1\,0\,0\,0)^T\}$. The CI-graph $\mathcal{G}(\widehat{\mathcal{M}}_4, N_3)$ and the child-parent tree structure $\mathcal{T}(\widehat{\mathcal{M}}_4, N_3)$ is shown in Fig. 11, where

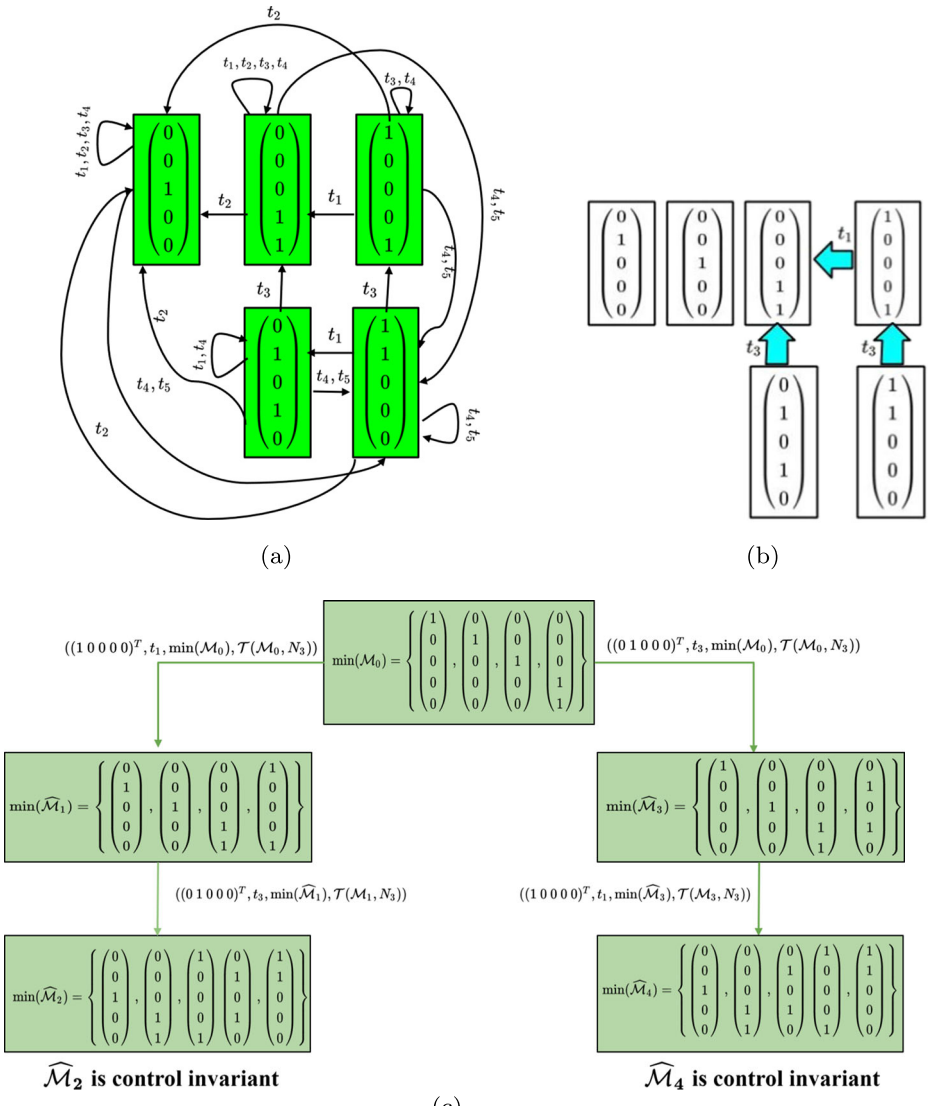

Fig. 11 The CI-graph and the child-parent tree structure of $\widehat{\mathcal{M}}_4$, where $min(\widehat{\mathcal{M}}_4) = \{(0\,0\,1\,0\,0)^T, (0\,0\,0\,1\,1)^T, (0\,1\,0\,1\,0)^T, (1\,0\,0\,0\,1)^T, (1\,1\,0\,0\,0)^T\}$, and $\mathcal{A}_{fail}(\widehat{\mathcal{M}}_4) = \emptyset$. **a** $\mathcal{G}(\widehat{\mathcal{M}}_4, N_3)$ **b** $\mathcal{T}(\widehat{\mathcal{M}}_4, N_3)$ **c** Current Decision Tree of the DFS algorithm

$\mathcal{A}^{fail}(\widehat{\mathcal{M}}_4) = \emptyset$, implying that $\widehat{\mathcal{M}}_4$ is control invariant. Therefore, this path terminates with $\widehat{\mathcal{M}}_4$ as the second control invariant subset found for $\mathcal{M}_0$.

As shown in Fig. 11c, $\widehat{\mathcal{M}}_4 = \widehat{\mathcal{M}}_2$, since there are no other elevation choices to be explored, the algorithm will terminate and will return $\mathcal{M}_0^{\Uparrow} = \widehat{\mathcal{M}}_4 \cup \widehat{\mathcal{M}}_2 = \widehat{\mathcal{M}}_2$; and, $min(\mathcal{M}_0^{\Uparrow}) = \{(0\,0\,1\,0\,0)^T, (0\,0\,0\,1\,1)^T, (0\,1\,0\,1\,0)^T, (1\,0\,0\,0\,1)^T, (1\,1\,0\,0\,0)^T\}$. In the Section 5 we formally consolidate the various observations and results introduced via illustrative examples in this section.

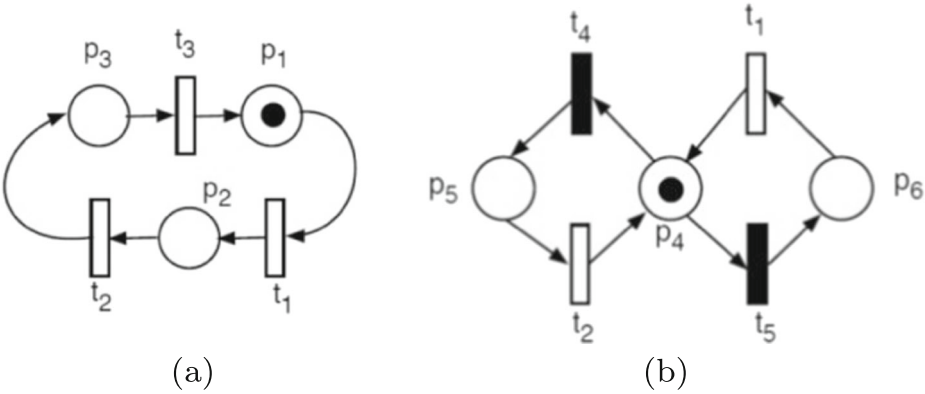

Fig. 12 The PN structures representing task flows for Robot 1 and Robot 2. **a** PN N_{4a} representing the task ow for Robot 1. **b** PN N_{4b} representing the task flow for Robot 2

4.3 Example 3: A practical example on control invariance property and its implication in LESP-synthesis

In this section, a practical PN example is presented to illustrate the importance of control invariance property in synthesizing LESPs for Discrete Event Systems (Khaleghi et al. 2019). Computation of the *minimally restrictive* LESP for PN models requires computation of the supremal right-closed control invariant subset of a right-closed set of markings.

Let us consider an automated system that paints automobile bodies using Booth 1 and Booth 2 and Robot 1 and Robot 2, where Robot 1 and Robot 2 are used for transportation tasks and painting tasks, respectively. A base-coat is applied to an unfinished body in Booth 1, following this it is transferred to Booth 2, where a coat of clear resin is applied. After sufficient time has elapsed for the solvents to evaporate, the finished automobile frame is transported out of Booth 2, and a new unfinished frame is placed in Booth 1 for painting. The base-coat operation can commence only after Robot 1 has placed an unfinished automobile body in Booth 1, and the base-coat paint nozzle is affixed to Robot 2. After the completion of this task, the automobile body with the base-coat is ready to be transported to Booth 2 by Robot 1. At this stage, Robot 2 can either be directed to do another base-coat operation in Booth 1; or, it can be commissioned to commence a clear-coat operation in Booth 2. The clear-coat operation can start only after Robot 1 has placed the body with the base-coat in Booth 2, and the clear-coat nozzle is affixed to Robot 2. After this task is completed, the finished automobile body is ready to be transported out of Booth 2 by Robot 1, which then places a new unfinished automobile body in Booth 1 for its base-coat.

Figure 12 shows the PN model representing the task sequence for Robot 1 and Robot 2. For N_{4a} representing the task flow of Robot 1, transition t_1 corresponds to the operation of putting the base-coat in Booth 1. Transition t_2 represents the operation of putting the clear coat in Booth 2. Transition t_3 denotes the operation of removing the painted automobile body from Booth 2, along with the process of placing a new, unfinished body on the fixture in Booth 1. The presence of a token in place p_1 (resp. p_2) denotes the resource state fact that an automobile body is on the fixture in Booth 1 (resp. Booth 2). A token in place p_3 denotes the resource state that a finished automobile body is at the fixture in Booth 2 to be removed. For N_{4b} representing the task sequence for Robot 2, similar to N_{4a}, transitions t_1 and t_2 corresponds to the operations of putting the base-coat and the clear-coat in Booth 1

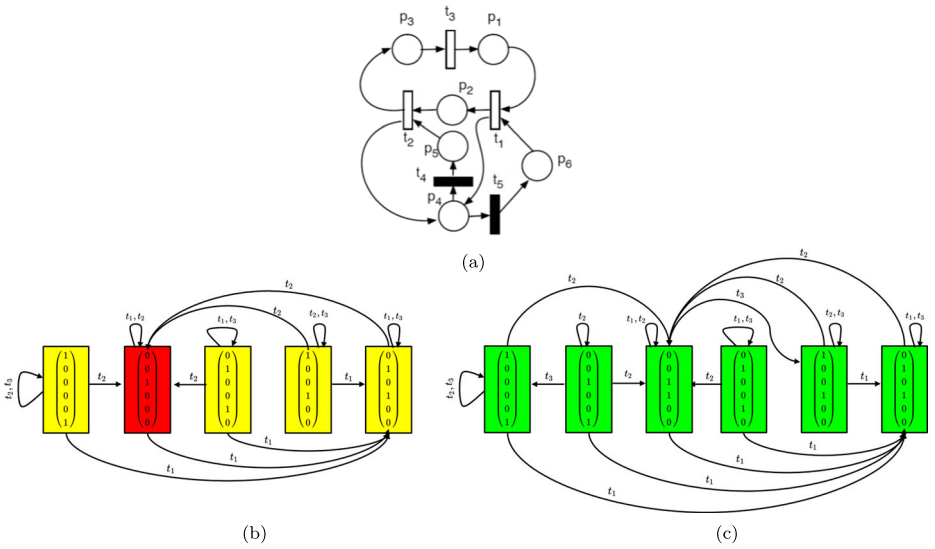

Fig. 13 The PN structure N_4, the CI-graph of N_4 for $min(\mathcal{M}_0) = \{(1\ 0\ 0\ 0\ 0\ 1)^T, (0\ 0\ 1\ 0\ 0\ 0)^T, (0\ 1\ 0\ 0\ 1\ 0)^T, (1\ 0\ 0\ 1\ 0\ 0)^T, (0\ 1\ 0\ 1\ 0\ 0)^T\}$ and $min(\mathcal{M}_1) = \{(1\,0\,0\,0\,0\,1)^T, (0\,0\,1\,0\,0\,1)^T, (0\,0\,1\,1\,0\,0)^T, (0\,1\,0\,0\,1\,0)^T, (1\,0\,0\,1\,0\,0)^T, (0\,1\,0\,1\,0\,0)^T\}$. **a** N_4 **b** $\mathcal{G}(\mathcal{M}_0, N_4)$ c $\mathcal{G}(\mathcal{M}_1, N_4)$

and Booth 2, respectively. Transition t_4 (resp. t_5) represents the operation where Robot 2 is commissioned to the clear-coat (resp. base-coat) paint-operation. The presence of a token in place p_4 indicates Robot 2 is ready to be commissioned for any of the two paint tasks. The presence of a token in place p_5 (resp. place p_6) indicates Robot 2 is ready to apply the clear-coat (resp. the base-coat) in Booth 2 (resp. Booth 1).

Figure 13a shows the PN structure N_4 that results from merging the two PNs of Fig. 12. PN N_4 belongs to the family of PNs for which the liveness property can be enforced by a *proper* right-closed set of minimal elements (cf. (Sreenivas 2012; Salimi et al. 2015; Chen et al. 2020)); as long as N_4 operates within the boundary defined by this proper right-closed set, the liveness property is guaranteed to be satisfied. Consequently, to synthesis a liveness policy for N_4, computation of the control invariant subset of its corresponding proper right-closed set is required in order to ensure that firing of no uncontrollable transition would result in a marking which is outside the boundary of the proper right-closed set. For N_4, transitions t_4 and t_5 are controllable, in that the decisions about how Robot 2 is commissioned is to be made by the supervisory policy which enforces the liveness property. On the other hand, all the transportation tasks assigned to Robot 1, represented by transitions t_1, t_2, and t_3, are uncontrollable; this means that as soon as a body part becomes available in any of the places p_1, p_2, or p_3, the transportation task can be started by Robot 1, without any restrictions from the liveness enforcing supervisory policy. Therefore, to reflect the fact that the supervisory policy has no control over the commencement of uncontrollable transitions, the proper right-closed set which enforces the liveness policy is required to satisfy the control invariance property, or, otherwise, commencement of transportation tasks by Robot 1 (i.e. firing of any of the transitions t_1, t_2, or t_3) may result in a state which violates the liveness property.

The LESP-synthesis, as discussed in (Khaleghi et al. 2019; Chandrasekaran et al. 2015), requires additional work than computing the largest right-closed control invariant subset of a given right-closed set. Without getting into the details of LESP-synthesis procedure, which is beyond the scope of this paper, let us assume, for N_4, the right-closed set $\mathcal{M}_0$, where $min(\mathcal{M}_0) = \{(1\,0\,0\,0\,0\,1)^T, (0\,0\,1\,0\,0\,0)^T, (0\,1\,0\,0\,1\,0)^T, (1\,0\,0\,1\,0\,0)^T, (0\,1\,0\,1\,0\,0)^T\}$ denotes the candidate right-closed set for enforcing the liveness property. Figure 13b shows the CI-graph of $\mathcal{M}_0$ with respect to the PN structure N_4. If we apply Eq. 1 to the minimal element $(0\,0\,1\,0\,0\,0)^T$ for t_3, the left-hand-side expression evaluates to $(1\,0\,0\,0\,0\,0)^T$, which is not in $\mathcal{M}_0$. This means that if a finished automobile body is at the fixture in Booth 2 (i.e. $(0\,0\,1\,0\,0\,0)^T$), Robot 1 can start the transportation task to remove it from Booth 2 and then, put an unfinished body in the fixture of Booth 1 (i.e. $(1\,0\,0\,0\,0\,0)^T$). As a result of this operation, the net will get trapped in state $(1\,0\,0\,0\,0\,0)^T$, which is outside the boundary of the set defined by $min(\mathcal{M}_0)$, and where no other operations can take place (*deadlock* state). This implies that the right-closed set $\mathcal{M}_0$ is not control invariant with respect to the PN structure N_4. Following the procedure dictated by the proposed algorithm for computing the supremal control invariant subset of a right-closed set, marking $(0\,0\,1\,0\,0\,0)^T$ will be replaced by two new minimal elements $\{(0\,0\,1\,0\,0\,1)^T, (0\,0\,1\,1\,0\,0)^T\}$, resulting in the right-closed set $\mathcal{M}_1$, where $min(\mathcal{M}_1) = \{(1\,0\,0\,0\,0\,1)^T, (0\,0\,1\,0\,0\,1)^T, (0\,0\,1\,1\,0\,0)^T, (0\,1\,0\,0\,1\,0)^T, (1\,0\,0\,1\,0\,0)^T, (0\,1\,0\,1\,0\,0)^T\}$. Figure 13b shows the CI-graph of $\mathcal{M}_1 (= \mathcal{M}_0^{\Uparrow})$, the largest right-closed control invariant subset of $\mathcal{M}_0$ with respect to the PN structure N_4. In fact, $\mathcal{M}_1$ represents the largest controllable right-closed set which enforces the *minimally restrictive* liveness policy for the manufacturing system represented by PN N_4 (Khaleghi et al. 2019).

5 Main results

Observation 4 follows directly from the elevation process and Algorithm1, and is stated without proof. Theorem 1 establishes the fact that any elevation process terminates in finite time.

Observation 4 $\overline{\mathbf{m}^1}$, as obtained from Eq. 1 and algorithm 1, is the smallest marking (termwise) greater than or equal to $\mathbf{m}^1$ such that $\overline{\mathbf{m}^1} \overset{t_u}{\Rightarrow} \mathbf{m}^2$.

Theorem 1 *Algorithm 1,* RAISE_FOR_CI($\mathbf{m}^1$, $\mathbf{m}^2$, t_u, N)*, will terminate in finite time.*

Proof : Following the notation of algorithm 1, let $\overline{\mathbf{m}}^1(p)$ be the marking that results from elevation of $\mathbf{m}^1(p)$ with respect to $\mathbf{m}^2(p)$ for transition t_u. If $\forall p \in \Pi, \overline{\mathbf{m}}^1(p) \geq \mathbf{m}^2(p)$, then $\overline{\mathbf{m}}^1(p)$ is dismissed by $\mathbf{m}^2$ and the algorithm terminates. If this is not the case, $\mathbf{m}^1$ will be elevated with respect to $\overline{\mathbf{m}}^1$. Let $\overline{\mathbf{m}}^1_1$ be the marking that results from this elevation process. Let us divide the vector $\mathbf{C}_{t_u}$ into three partitions for $p \in \Pi$ and consider the following three cases:

1. $\mathbf{C}_{t_u}(p) = 0$: using the following two scenarios, we show that for this case $\overline{\mathbf{m}}^1_1(p) = \overline{\mathbf{m}}^1(p)$:

1.1. If $\overline{\mathbf{m}}^1(p) = \mathbf{m}^1(p)$, which implies $max\{\mathbf{m}^1(p), \mathbf{IN}_{t_u}(p)\} \geq \mathbf{m}^2(p) - \mathbf{C}_{t_u}(p)$. Now, if $\mathbf{m}^2(p)$ is replaced by $\overline{\mathbf{m}}^1(p) = \mathbf{m}^1(p)$ on the right-hand side and given the fact that $\mathbf{C}_{t_u}(p) = 0$, we get $max\{\mathbf{m}^1(p), \mathbf{IN}_{t_u}(p)\} \geq \mathbf{m}^1(p) - 0 = \mathbf{m}^1(p)$. Therefore, we will have $\overline{\mathbf{m}}^1_1(p) = \overline{\mathbf{m}}^1(p)$.

1.2. If $\overline{\mathbf{m}}^1(p) = \mathbf{m}^2(p) - \mathbf{C}_{t_u}(p)$, which implies $max\{\mathbf{m}^1(p), \mathbf{IN}_{t_u}(p)\} < \mathbf{m}^2(p) - \mathbf{C}_{t_u}(p)$. Now, if $\mathbf{m}^2(p)$ is replaced by $\overline{\mathbf{m}}^1(p) = \mathbf{m}^2(p) - \mathbf{C}_{t_u}(p)$ on the right-hand side and given the fact that $\mathbf{C}_{t_u}(p) = 0$, we get $max\{\mathbf{m}^1(p), \mathbf{IN}_{t_u}(p)\} < \mathbf{m}^2(p) - 2\mathbf{C}_{t_u}(p) = \mathbf{m}^2(p) - 2 * 0 = \mathbf{m}^2(p)$. Therefore, we will have $\overline{\mathbf{m}}^1_1(p) = \overline{\mathbf{m}}^1(p)$.

2. $\mathbf{C}_{t_u}(p) < 0$: using the following two scenarios, we show that for this case $\overline{\mathbf{m}}^1_1(p) \geq \overline{\mathbf{m}}^1(p)$:

2.1. $\overline{\mathbf{m}}^1(p) = \mathbf{m}^1(p)$, which implies $max\{\mathbf{m}^1(p), \mathbf{IN}_{t_u}(p)\} \geq \mathbf{m}^2(p) - \mathbf{C}_{t_u}(p)$. Now, if $\mathbf{m}^2(p)$ is replaced by $\overline{\mathbf{m}}^1(p) = \mathbf{m}^1(p)$ on the right-hand side and given the fact that $\mathbf{C}_{t_u}(p) < 0$, $\overline{\mathbf{m}}^1_1(p)$ will either be equal to $\mathbf{m}^1(p)$ or $\mathbf{m}^1(p) - \mathbf{C}_{t_u}(p)$. Therefore, we will have $\overline{\mathbf{m}}^1_1(p) \geq \overline{\mathbf{m}}^1(p)$.

2.2. $\overline{\mathbf{m}}^1(p) = \mathbf{m}^2(p) - \mathbf{C}_{t_u}(p)$, which implies $max\{\mathbf{m}^1(p), \mathbf{IN}_{t_u}(p)\} < \mathbf{m}^2(p) - \mathbf{C}_{t_u}(p)$. Now, if $\mathbf{m}^2(p)$ is replaced by $\overline{\mathbf{m}}^1(p) = \mathbf{m}^2(p) - \mathbf{C}_{t_u}(p)$ on the right-hand side and given the fact that $\mathbf{C}_{t_u}(p) < 0$, we get $max\{\mathbf{m}^1(p), \mathbf{IN}_{t_u}(p)\} < \mathbf{m}^2(p) - 2\mathbf{C}_{t_u}(p)$. Therefore, $\overline{\mathbf{m}}^1_1(p) = \mathbf{m}^2(p) - 2\mathbf{C}_{t_u}(p) > \overline{\mathbf{m}}^1(p) = \mathbf{m}^2(p) - \mathbf{C}_{t_u}(p)$.

3. $\mathbf{C}_{t_u}(p) > 0$:

3.1. $\overline{\mathbf{m}}^1(p) = \mathbf{m}^1(p)$, which implies $max\{\mathbf{m}^1(p), \mathbf{IN}_{t_u}(p)\} \geq \mathbf{m}^2(p) - \mathbf{C}_{t_u}(p)$. Now, if $\mathbf{m}^2(p)$ is replaced by $\overline{\mathbf{m}}^1(p) = \mathbf{m}^1(p)$ on the right-hand side and given the fact that $\mathbf{C}_{t_u}(p) > 0$, we get $max\{\mathbf{m}^1(p), \mathbf{IN}_t(p)\} > \mathbf{m}^1(p) - \mathbf{C}_t(p)$. Therefore, we will have $\overline{\mathbf{m}}^1_1(p) = \overline{\mathbf{m}}^1(p)$.

3.2. $\overline{\mathbf{m}}^1(p) = \mathbf{m}^2(p) - \mathbf{C}_{t_u}(p)$, which implies $max\{\mathbf{m}^1(p), \mathbf{IN}_{t_u}(p)\} < \mathbf{m}^2(p) - \mathbf{C}_{t_u}(p)$. Now, if $\mathbf{m}^2(p)$ is replaced by $\overline{\mathbf{m}}^1(p) = \mathbf{m}^2(p) - \mathbf{C}_{t_u}(p)$, $\overline{\mathbf{m}}^1_1(p)$ will either be equal to $\mathbf{m}^2(p) - 2\mathbf{C}_{t_u}(p)$ or $\mathbf{m}^1(p)$. Also, note that since $max\{\mathbf{m}^1(p), \mathbf{IN}_{t_u}(p)\} < \mathbf{m}^2(p) - \mathbf{C}_{t_u}(p)$, we can conclude $\mathbf{m}^1(p) < \mathbf{m}^2(p) - \mathbf{C}_{t_u}(p)$. Therefore, we will have $\overline{\mathbf{m}}^1_1(p) \leq \overline{\mathbf{m}}^1(p)$.

The three cases explained above can be generalized to when $\mathbf{m}^1(p)$ is to be raised with respect to $\overline{\mathbf{m}}^1_k(p)$ in the k^{th} iteration of the algorithm. However note that for the first two cases, we will always have $\overline{\mathbf{m}}^1_k(p) \geq \overline{\mathbf{m}}^1_{k-1}(p)$. For the third case, if scenario [3.1] happens, then it is clear that $\overline{\mathbf{m}}^1_k(p) = \overline{\mathbf{m}}^1_{k-1}(p) = \mathbf{m}^1(p)$ and the algorithm will terminate as $\overline{\mathbf{m}}^1_{k-1}(p)$ will dismiss $\overline{\mathbf{m}}^1_k(p)$. Regarding scenario [3.2], we might observe a non-increasing trend/token load from $\overline{\mathbf{m}}^1_{k-1}(p)$ to $\overline{\mathbf{m}}^1_k(p)$. However, this trend will turn into a constant non-decreasing one in finite number of iterations. For the k^{th} iteration of algorithm **??**, the right-hand side of step 7 will be equal to $\mathbf{m}^2(p) - k\mathbf{C}_{t_u}(p)$. If $\mathbf{m}^2(p) - k\mathbf{C}_{t_u}(p)$ turns out to be a negative value, the value of $\overline{\mathbf{m}}^1_k(p)$ for the current and subsequent iterations will be fixed at $\mathbf{m}^1(p)$; that is due to the fact that $max\{\mathbf{m}^1(p), \mathbf{IN}_{t_u}(p)\} \geq 0$. Therefore, as soon as the fixed value of $\mathbf{m}^1(p)$ is obtained for all $p \in \Pi$ where $\mathbf{C}_{t_u}(p) > 0$, the algorithm will terminate. Consequently, the number of iterations of the Algorithm 1 will be bounded by $\max_{\mathbf{C}_{t_u}(p) \geq 0} \lceil \frac{\mathbf{m}^2(p)}{\mathbf{C}_{t_u}(p)} \rceil$.

□

The directives listed below play an important role in establishing the correctness of Algorithm 3 and its constituent components.

Directive 1 If during an iteration of Algorithm 3, a parent marking such as $\mathbf{m}^k$ fails the *CI* test for some uncontrollable transition such as t_u, no member of $\mathcal{C}(\mathbf{m}^k)$ (where $\mathcal{C}(\mathbf{m}^k)$ denotes the set of children of $\mathbf{m}^k$) should be used to elevate $\mathbf{m}^k$ for t_u.

Directive 2 If during an iteration of Algorithm 3, a parent marking such as $\mathbf{m}^k$ fails the *CI* test for some uncontrollable transition and is replaced by the descendant set $\mathcal{D}(\mathbf{m}^k)$ as the result of the elevation process, then all its children, $\mathcal{C}(\mathbf{m}^k)$, might have to be elevated accordingly. This process is referred to as "elevation for control invariance forced by the parent" and it seeks to maintain the existing child-parent relations. In the simple $\overline{\mathbf{m}^1} \overset{t_u}{\Rightarrow} \mathbf{m}^2$ example, if $\mathbf{m}^2$ fails the *CI* test for some uncontrollable transition and is replaced by $\mathcal{D}(\mathbf{m}^2)$, we need to perform RAISE_FOR_CI($\overline{\mathbf{m}^1}$, $\mathcal{D}(\mathbf{m}^2)$, t_u, N) as dictated by Algorithm 1.

Directive 3 If during an iteration of Algorithm 3, as $\mathbf{m}^k$ fails the *CI* test for some uncontrollable transition and when elevated, $\mathcal{D}(\mathbf{m}^k)$ turns out to be empty (i.e. all descendants of $\mathbf{m}^k$ are dismissed), as $\mathbf{m}^k$ gets removed from the current control invariant estimate, so do all children of $\mathbf{m}^k$, $\mathcal{C}(\mathbf{m}^k)$. The reasoning behind this policy is explained in more details in Theorem 2.

Theorem 2 *Suppose $\widehat{\mathcal{M}}$ is an estimate of the right-closed control invariant subset of the initial set. Also, let's assume we have $\mathbf{m}_2^1, \mathbf{m}^2 \in \min(\widehat{\mathcal{M}})$ and $\mathbf{m}_2^1 \in \mathcal{D}(\mathbf{m}^1, t_1)$ such that $\mathbf{m}_2^1 \overset{t_1}{\Rightarrow} \mathbf{m}^2$, which implies $\mathbf{m}_2^1$ is a child of $\mathbf{m}^2$. Let there exists a transition $t_2 \in T_u$ such that $\mathbf{m}^2 \overset{t_2}{\rightsquigarrow} \widehat{\mathbf{m}}^2$ and $\widehat{\mathbf{m}}^2 \notin \widehat{\mathcal{M}}$. Next, suppose that if $\mathbf{m}^2$ is elevated for t_2, $\nexists \overline{\mathbf{m}}^2 \in \mathcal{D}(\mathbf{m}^2, t_2)$ such that (a) $\overline{\mathbf{m}}^2 \not\geq \mathbf{m}^i$ for any $\mathbf{m}^i \in \min(\widehat{\mathcal{M}}) - \{\mathbf{m}^2\}$; and, (b) $\overline{\mathbf{m}}^2 \overset{t_2}{\rightsquigarrow} \underline{\mathbf{m}}^2$, $\underline{\mathbf{m}}^2 \geq \mathbf{m}^i$ for some $\mathbf{m}^i \in \min(\widehat{\mathcal{M}}) - \{\mathbf{m}^2\}$; i.e. $\mathcal{D}(\boldsymbol{m}^2, t_2) = \emptyset$. We seek to prove that once the parent marking $\boldsymbol{m}^2$ is dropped without having any descendants, there is no need to keep its child, and $\boldsymbol{m}_2^1$ can also be removed from the current control invariant estimate that is, $\nexists \overline{\mathbf{m}}_2^1 \in \mathcal{D}(\mathbf{m}_2^1)$ such that:(a) $\overline{\mathbf{m}}_2^1 \not\geq \mathbf{m}^i$ for any $\mathbf{m}^i \in \min(\widehat{\mathcal{M}}) - \{\mathbf{m}^2\} - \{\mathbf{m}_2^1\}$; and, (b) $\overline{\mathbf{m}}_2^1 \overset{t_1}{\rightsquigarrow} \underline{\mathbf{m}}_2^1$ and $\overline{\mathbf{m}}_2^1 \overset{t_2}{\rightsquigarrow} \underline{\underline{\mathbf{m}}}_2^1$ such that $\underline{\mathbf{m}}^2, \underline{\underline{\mathbf{m}}}_2^1 \geq \mathbf{m}^i$ for some $\mathbf{m}^i \in \min(\widehat{\mathcal{M}}) - \{\mathbf{m}^2\} - \{\mathbf{m}_2^1\}$.*

Proof . Since $\mathbf{m}^2$ is the only parent of $\mathbf{m}_2^1$ for t_1, if we re-write $\mathbf{m}_2^1$ as $\mathbf{m}^1 + \mathbf{x}$ for a vector $\mathbf{x} \geq \mathbf{0}$, we used to have $max\{\mathbf{m}^1 + \mathbf{x}, \mathbf{IN}_{t_1}\} + \mathbf{C}_{t_1} \geq \mathbf{m}^2$. It is clear that upon removal of $\mathbf{m}^2$ from $\widehat{\mathcal{M}}$, $\mathbf{m}_2^1$ will fail the requirement of *CI* test for transition t_1. Now, let's suppose $\mathbf{m}_2^1$ is to be raised for t_1 with respect to a marking $\mathbf{m}^3 \in \min(\widehat{\mathcal{M}}) - \{\mathbf{m}^2\}$. In other words, we seek to find vector $\mathbf{z} \geq 0$ such that: $max\{\mathbf{m}_2^1 + \mathbf{z}, \mathbf{IN}_{t_1}\} + \mathbf{C}_{t_1} \geq \mathbf{m}^3$; let's call this newly formed marking $\mathbf{m}_2^{1\uparrow}$ (where $\mathbf{m}_2^{1\uparrow} = \mathbf{m}^1 + \mathbf{x} + \mathbf{z}$). We will have either of the two following scenarios:

1. $\exists \mathbf{m}_3^1 \in \min(\widehat{\mathcal{M}}) - \{\mathbf{m}^2\}$ and $\mathbf{m}_3^1 \in \mathcal{D}(\mathbf{m}^1, t_1)$ such that $\mathbf{m}_3^1 \overset{t_1}{\Rightarrow} \mathbf{m}^3$, which implies $\mathbf{m}_3^1$ is a child of $\mathbf{m}^3$. In other words, if we re-write $\mathbf{m}_3^1$ as $\mathbf{m}^1 + \mathbf{y}$ for a vector $\mathbf{y} \geq \mathbf{0}$, we have $max\{\mathbf{m}^1 + \mathbf{y}, \mathbf{IN}_{t_1}\} + \mathbf{C}_{t_1} \geq \mathbf{m}^3$. It is not hard to show that $\mathbf{x}(p) + \mathbf{z}(p) \geq \mathbf{y}(p), \forall p \in \Pi$, viz. $\mathbf{m}_2^{1\uparrow} = \mathbf{m}^1 + \mathbf{x} + \mathbf{z} \geq \mathbf{m}_3^1 = \mathbf{m}^1 + \mathbf{y}$. Therefore, $\mathbf{m}_3^1$ will dismiss $\mathbf{m}_2^{1\uparrow}$ (as $\mathbf{m}_2^{1\uparrow}$ is term-wise larger than $\mathbf{m}_3^1$) and there is no need to elevate $\mathbf{m}_2^1$ with respect to $\mathbf{m}^3$ for transition t_1.

2. $\nexists \mathbf{m}_3^1 \in \min(\widehat{\mathcal{M}}) - \{\mathbf{m}^2\}$ and $\mathbf{m}_3^1 \in \mathcal{D}(\mathbf{m}^1, t_1)$ such that $\mathbf{m}_3^1 \overset{t_1}{\Rightarrow} \mathbf{m}^3$. The primary reason for non-existence of $\mathbf{m}_3^1$ could be that during a previous iteration of the algorithm, $\mathbf{m}_3^1$ got dismissed by a marking such as $\mathbf{m}^4$:
   (a) If $\mathbf{m}^4 \in \min(\widehat{\mathcal{M}}) - \{\mathbf{m}^2\}$, it means that $\mathbf{m}^4$ is still a minimal element of the current control invariant subset estimate. In this case, since $\mathbf{m}_3^1 \geq \mathbf{m}^4$ and from scenario 1, $\mathbf{m}_2^{1\uparrow} \geq \mathbf{m}_3^1$, then we can conclude $\mathbf{m}_2^{1\uparrow} \geq \mathbf{m}^4$, which implies that $\mathbf{m}^4$ will dismiss $\mathbf{m}_2^{1\uparrow}$ and there is no need to elevate $\mathbf{m}_2^1$ with respect to $\mathbf{m}^3$ for transition t_1.
   (b) If $\mathbf{m}^4 \notin \min(\widehat{\mathcal{M}}) - \{\mathbf{m}^2\}$, it means that $\mathbf{m}^4$ is no longer a minimal element of the current control invariant subset. In other words, during a previous iteration of the algorithm with estimate $\mathcal{M}^{current}$, $\mathbf{m}^4$ must have failed in some transition such as t_3 and when elevated for transition t_3, $\nexists \overline{\mathbf{m}}^4 \in \mathcal{D}(\mathbf{m}^4, t_3)$ such that: (a) $\overline{\mathbf{m}}^4 \not\geq \mathbf{m}^i$ for any $\mathbf{m}^i \in \min(\mathcal{M}^{current}) - \{\mathbf{m}^4\})$; and, (b) $\overline{\mathbf{m}}^4 \overset{t_3}{\rightsquigarrow} \underline{\mathbf{m}}^4$ and $\underline{\mathbf{m}}^4 \geq \mathbf{m}^i$ for some $\mathbf{m}^i \in \min(\mathcal{M}^{current}) - \{\mathbf{m}^4\}$. We know that $\mathbf{m}_3^1 \geq \mathbf{m}^4$ and from scenario 1, $\mathbf{m}_2^{1\uparrow} \geq \mathbf{m}_3^1$; So, we get $\mathbf{m}_2^{1\uparrow} \geq \mathbf{m}^4$. Note that here, $\mathbf{m}_2^{1\uparrow}$ may or may not fail the *CI* test for transition t_3. In the latter case, it will get dismissed by the same marking which covers $\mathbf{m}_2^{1\uparrow}$ for t_3 (or the same marking that dismissed the elevated version of $\mathbf{m}^4$ for t_3). In the former case, $\mathbf{m}_2^{1\uparrow}$ does not satisfy the requirement of *CI* test for transition t_3 and there cannot exist any non-dismissed descendant for $\mathbf{m}_2^{1\uparrow}$ which satisfies the *CI* test for t_3 (i.e. $\mathcal{D}(\mathbf{m}_2^{1\uparrow}, t_3) = \emptyset$); or otherwise, $\exists \overline{\mathbf{m}}^4 \in \mathcal{D}(\mathbf{m}^4, t_3)$ such that $\overline{\mathbf{m}}^4 \overset{t_3}{\rightsquigarrow} \underline{\mathbf{m}}^4$ and $\underline{\mathbf{m}}^4 \geq \mathbf{m}^i$ for some $\mathbf{m}^i \in \min(\mathcal{M}^{current})$, which is a contradiction. So, there is no need to elevate $\mathbf{m}_2^1$ with respect to $\mathbf{m}^3$ for transition t_1.

□

The following Theorem notes that Algorithm 3 will raise a minimal element at most once with respect to an uncontrollable transitions. In fact, repeated elevation of a minimal element can only occur via Directive 2.

Theorem 3 *Algorithm 3 ensures that each marking (and its descendants) is at most elevated once with respect to the same transition, unless through Directive 2.*

Proof . Suppose $\widehat{\mathcal{M}}$ is an estimate of the right-closed control invariant subset of the initial set. Also, let's assume we have $\mathbf{m}^i \in \min(\widehat{\mathcal{M}})$. First it should be noted that if $\mathbf{m}^i$ is not the child of any other minimal element of $\min(\widehat{\mathcal{M}})$, then it means it has never been elevated before with respect to any transitions. Now, let us assume there exist $\mathbf{m}^j \in \min(\widehat{\mathcal{M}})$ such that $\mathbf{m}^i \overset{t_u}{\Rightarrow} \mathbf{m}^j$, which implies $\mathbf{m}^i$ is a child of $\mathbf{m}^j$ through transition t_u.

1. If $\mathbf{m}^j \in \mathcal{A}^{pass}(\widehat{\mathcal{M}})$, it implies that the parent marking for $\mathbf{m}^i$ will never get dropped or elevated/updated:
   (a) If $\mathbf{m}^i \in \mathcal{A}^{pass}(\widehat{\mathcal{M}})$, then $\mathbf{m}^i \overset{t_u}{\Rightarrow} \mathbf{m}^j$ will hold for the remaining elevation rounds of the algorithm (if any).
   (b) If $\mathbf{m}^i \in \mathcal{A}^{fail}(\widehat{\mathcal{M}})$, then it means $\mathbf{m}^i$ fails/might fail in at least one more transition except t_u; consider one such transition t_l:
      (i) . If $\mathcal{D}(\mathbf{m}^i, t_l) \neq \emptyset$,then $\mathbf{m}^i$ will be replaced by some larger descendants such as $\overline{\mathbf{m}}^i$ and, therefore, $\overline{\mathbf{m}}^i \overset{t_u}{\Rightarrow} \mathbf{m}^j$ will still hold.

(ii) If $\mathcal{D}(\mathbf{m}^i, t_l) = \emptyset$,then $\mathbf{m}^i$ will be removed and no future elevation for $\mathbf{m}^i$ with respect to t_u is ever needed.

2. If $\mathbf{m}^j \in \mathcal{A}^{fail}(\widehat{\mathcal{M}})$, it implies that $\mathbf{m}^j$ fails/might fail in at least one transition; consider one such transition t_s:
   (a) If $\mathcal{D}(\mathbf{m}^j, t_s) = \emptyset$, then by Theorem 2, as $\mathbf{m}^j$ gets removed, so does $\mathbf{m}^i$ and no future elevation for $\mathbf{m}^i$ with respect to t_u is ever needed.
   (b) If $\mathcal{D}(\mathbf{m}^j, t_s) \neq \emptyset$, then $\mathbf{m}^i$ will be replaced by some larger descendants such as $\overline{\mathbf{m}}^i$ resulting from Directive 2. Therefore, $\overline{\mathbf{m}}^i \overset{t_u}{\Rightarrow} \overline{\mathbf{m}}^j$ will still hold, where $\overline{\mathbf{m}}^j \in \mathcal{D}(\mathbf{m}^j, t_s)$.

□

The following observation identifies conditions under which Algorithm 3 terminates, and follows directly from its constituent procedures.

Observation 5 At the point of termination for each of the computational paths of the DFS-tree structure of Algorithm 3, either of the following termination conditions is reached for the corresponding leaf node denoted by $min(\mathcal{M}_i)$: (1) $min(\mathcal{M}_i) = \emptyset$; (2) $\mathcal{A}^{fail}(\mathcal{M}_i) = \emptyset$.

Theorem 4 notes that Algorithm 3 terminates in finite-time, a sketch of the proof of this claim in presented below, as well.

Theorem 4 *Algorithm 3 will terminate in finite time.*

Proof (*Sketch*). This Theorem can be inferred based on the following results: (1) Theorem 1 which proves each elevation round will eventually converge; (2) Theorem 3 which implies no marking is elevated more than once for the same uncontrollable transition; (3) Number of uncontrollable transitions of the PN structure N being finite; (4) Number of *descendants* generated at each elevation round for a marking which fails the test for CI condition being finite. □

As the culmination of the various observations and Theorem in this section, the following observation notes that the union of the control invariant subsets obtained at the conclusion of each branch of the DFS-based approach is the supremal right-closed control invariant subset of the original right-closed set.

Observation 6 Algorithm 3 returns the largest right-closed control invariant subset of the initial set $\mathcal{M}_0$. This result can be argued using Observation 4 and the brute-force approach implemented by the DFS-based strategy which checks all possible computational paths for the elevation process exhaustively.

6 Discussion

The supremal right-closed control invariant subset of a right-closed set $\mathcal{M} \subseteq \mathbb{Z}^n$ (with respect to a PN structure N), denoted by $\mathcal{M}^{\Uparrow}$ ($\mathcal{M}^{\Uparrow} \subseteq \mathcal{M}^{\uparrow} \subseteq \mathcal{M}$), satisfies the requirement – $\forall \mathbf{m}^1 \in min(\mathcal{M}^{\Uparrow}), \exists \mathbf{m}^2 \in min(\mathcal{M})$ such that $\mathbf{m}^1 \geq \mathbf{m}^2$. In addition, $\forall \mathbf{m}^1 \in min(\mathcal{M}^{\Uparrow}), \forall t_u \in T_u, (\mathbf{m}^1 \overset{t_u}{\rightsquigarrow} \widehat{\mathbf{m}}^1) \Rightarrow (\widehat{\mathbf{m}}^1 \in \mathcal{M}^{\Uparrow})$.

If a right-closed set $\mathcal{M} \subseteq \mathbb{Z}^n$ is not control invariant with respect to a PN structure N, then $\exists \mathbf{m}^1 \in min(\mathcal{M}), \exists t_u \in T_u$, such that $\mathbf{m}^1 \overset{t_u}{\rightsquigarrow} \widehat{\mathbf{m}}^1$, and $\widehat{\mathbf{m}}^1 \notin \mathcal{M}$. The process of computing the minimal elements of $\mathcal{M}^{\Uparrow}$ is about developing a systematic approach to elevating $\mathbf{m}^1$ with respect to the other minimal elements, for each $t_u \in T_u$ where the CI-test (cf. Equation 1) has been violated. At any given stage, there are many decision choices involved in this elevation process (i) the choice of the other minimal element, and (ii) the choice of the uncontrollable transition that featured in the violation of Eq. 1. Using a DFS-based approach, the algorithm outlined in this paper explores *all* such choices. At its termination point, each computational path of this DFS-based approach computes a right-closed control invariant subset of the original right-closed set, and the termination is shown to happen in finite-time. The set $\mathcal{M}^{\Uparrow}$ is the union of the control invariant right-closed sets generated at the termination of each path of the DFS-based tree. Reference (Chandrasekaran et al. 2015) contains an implementation of a single-branch of the DFS-based procedure outlined in this paper.

The $\mathcal{H}$-class of PN structures is identified by the following structural properties: (1) for each place, the weights associated with the outgoing arcs that terminate on uncontrollable transitions must be the smallest of all outgoing arc-weights; (2) the set of input places to each uncontrollable transition is no larger than the set of input places of any transition which shares a common input place with it (cf. Salimi et al. 2015; Chen et al. 2020). All PN examples in this paper belong to the $\mathcal{H}$-class of PN structures. In all our experiments of PN structures from this class, we have found that each branch of the DFS-based procedure outputs the same control invariant subset. As an illustration, $min(\widehat{\mathcal{M}}_2) = min(\widehat{\mathcal{M}}_4)$ in the example shown in Fig. 11c. This leads us to hypothesize that for the $\mathcal{H}$-class of PN structures, the supremal right-closed control invariant subset can be identified as soon as a single branch of the DFS-procedure terminates. We suggest formal investigations into this hypothesis as a future research topic.

In terms of running time, the total number of nodes of the DFS-tree is a measure of the upper-bound of execution time of the proposed algorithm, which depends on the depth-bound (α) and the breadth-bound (β) of the DFS-structure (which in turn depends on the minimal elements of $\mathcal{M}_0$ and the PN structure N). As a consequence of Directive 1, Directive 2 and Theorem 3, we have $\alpha \leq O(|min(\mathcal{M}_0)||T_u|$. For a given node, $min(\mathcal{M}_i)$, the max number of branches that originates from this node at the breadth level is bounded by $|min(\mathcal{M}_i)||T_u|$. Consequently, $\beta \leq \max_i O(|min(\mathcal{M}_i)||T_u|)$, and the (conservative) upper-bound of the execution-time is $O(\alpha^{\beta})$, which is the conservative estimate of the total number of nodes in the aforementioned DFS-structure. We hypothesize that if the PN structure belongs to the $\mathcal{H}$-class, then all branches of the DFS-structure yield the same right-closed set, if this hypothesis is true, then the upper-bound of the execution-time is $O(\alpha)$.

7 Conclusion

In context of District Event Systems modeled by Petri Net structures, a Liveness Enforcing Supervisory Policy (LESP) enforces the liveness property by disabling controllable events which result in states which are not live. Therefore, the set of initial markings for which an LESP exists needs to be control invariant with respect to its corresponding PN structure. The control invariance property ensures that firing of no uncontrollable transition would result in a state that violates the liveness property, which is intended to be enforced by the LESP. As a prerequisite step in synthesis of an LESP for certain classes of PN structures, this

paper presented a formal algorithm to compute the largest (aka the supremal) right-closed control invariant subset of a right-closed set with respect to its corresponding PN structure. The *supremal* property is, in particular, necessary for computation of *minimally restrictive* LESPs. The proposed algorithm works based on a step-by-step elevation framework performed on the minimal elements of the initial right-closed set which fail to satisfy the control invariance property. During each elevation iteration of the algorithm, a target marking which fails the control invariance property for at least one uncontrollable transition is investigated, resulting in either its removal from the current control invariant estimate or its replacement by a set of elevated descendants. To obtain the supremal property, a tree-based Depth-First-Search (DFS) strategy is implemented which ensures for each pair of minimal elements and transitions of a given right-closed set, all necessary elevation procedures are tested exhaustively. The termination point of each path of the DFS tree corresponds to a right-closed control invariant subset of the initial right-closed set. Once, all paths of the DFS tree reach the termination condition, the algorithm returns the union of all right-closed control invariant subsets found at each leaf node as the final supremal right-closed control invariant subset of the initial right-closed set. The elevation process is designed in such a way that the algorithm is guaranteed to terminate in finite number of iterations. The paper presented several PN examples to illustrate the fundamental elevation process which lies at the heart of the DFS strategy. All illustrative examples of this paper belong to $\mathcal{H}$-class of PN structures, where the output of each leaf node of the DFS tree outputs the same control invariant subset. For future work, we suggest a deeper investigation into the hypothesis that for the $\mathcal{H}$-class of PN structures, as soon as a single path of the DFS tree terminates, the supremal right-closed control invariant subset of the initial right-closed set can be obtained.

Appendix

Algorithm 1 RAISE_FOR_CI($\mathbf{m}^1$, $\mathbf{m}^2$, t_u, N).

Create an empty array $\mathcal{D}^{\mathbf{m}^1}$ to store descendants of $\mathbf{m}^1$
Create a tree $\mathcal{T}^{\mathbf{m}^1}$ with root $\mathbf{m}^2$
for all $p \in P$ **do**
    if $max\{\mathbf{m}^1(p), \mathbf{IN}_{t_u}(p)\} \geq \mathbf{m}^2(p) - \mathbf{C}_{t_u}(p)$ **then**
        $\overline{\mathbf{m}^1}(p) = \mathbf{m}^1(p)$
    else
        $\overline{\mathbf{m}^1}(p) = \mathbf{m}^2(p) - \mathbf{C}_{t_u}(p)$
    end if
end for
while $\overline{\mathbf{m}^1}$ is not dismissed by $\mathbf{m}^2$ **do**
    Update $\mathcal{D}^{\mathbf{m}^1}$ to include $\overline{\mathbf{m}^1}$
    Create a vertex $\overline{\mathbf{m}^1}$
    Create a new arc such that it starts from $\overline{\mathbf{m}^1}$ and ends at $\mathbf{m}^2$ with label t_u
    $\mathbf{m}^2 \leftarrow \overline{\mathbf{m}^1}$ and repeat steps 3 to 9
end while
Return $\mathcal{D}^{\mathbf{m}^1}$ and $\mathcal{T}^{\mathbf{m}^1}$

Algorithm 2 ELEVATION_FUNCTION($\mathbf{M}^p, t_u, min(\mathcal{M}_i), \mathcal{T}(\mathcal{M}_i, N)$).

Store the most recent collection of child-parent trees in $\mathcal{T}_0$: $\mathcal{T}_0 \leftarrow \mathcal{T}(\mathcal{M}_i, N)$
Remove vertex $\mathbf{m}^p$ and all its children from $\mathcal{T}(\mathcal{M}_i, N)$
$min(\mathcal{M}_i) \leftarrow min(\mathcal{M}_i) - \{\mathbf{m}^p\} - \{\mathcal{C}(\mathbf{m}^p)\}$
if $min(\mathcal{M}_i)$ is empty **then**
    Return an empty set
else
    Create an empty dictionary $desDict$ to store markings as key and the array containing their descendants as values
    Create an empty array $\mathcal{D}^{all}$ to store newly formed descendants of $\mathbf{m}^p$ as well as descendants of its children
    Create an empty stack $\mathcal{T}^{collection}$ to store collection of newly formed child-parent trees
    for all $\mathbf{m}^k \in min(\mathcal{M}_i)$ **do**
        Call RAISE_FOR_CI($\mathbf{M}^p, \mathbf{M}^k, t_u, N$) to find $\mathcal{D}^{\mathbf{m}^p}$ and $\mathcal{T}^{\mathbf{m}^p}$
        Update $\mathcal{D}^{all}$ to include $\mathcal{D}^{\mathbf{m}^p}$
        Update $\mathcal{T}^{collection}$ to include $\mathcal{T}^{\mathbf{m}^p}$
    end for
    Update $desDict$ to include $desDict[\mathbf{m}^p] = \mathcal{D}^{all}$
    Create an empty list $\mathcal{S}$ and add all direct children of $\mathbf{m}^p$ from $\mathcal{T}_0$ to the list: each child is stored as a 3-tuple $(\mathbf{m}^c, \mathbf{m}^p, t_l)$, where the first argument is the child's marking, the second one is the parent's marking, and the third one is the label of the arc connecting the child to the parent
    while $\mathcal{S}$ is not empty **do**
        Remove and return the first element from $\mathcal{S}$, stored as $(\mathbf{m}^c, \mathbf{m}^p, t_l)$
        Add all direct children of $\mathbf{m}^c$ to the end of $\mathcal{S}$ as in step 16
        Create an empty array $\mathcal{D}^{temp}$ to store descendants of $\mathbf{m}^c$
        for all $\mathbf{m}^j \in desDict[\mathbf{m}^p]$ **do**
            Call RAISE_FOR_CI($\mathbf{M}^c, \mathbf{M}^j, t_l, N$) to find $\mathcal{D}^{\mathbf{m}^c}$ and $\mathcal{T}^{\mathbf{m}^c}$
            Update $\mathcal{D}^{temp}$ to include $\mathcal{D}^{\mathbf{m}^c}$
            Update $\mathcal{D}^{all}$ to include $\mathcal{D}^{\mathbf{m}^c}$
            Update $\mathcal{T}^{collection}$ to include $\mathcal{T}^{\mathbf{m}^c}$
        end for
        Update $desDict$ to include $desDict[\mathbf{m}^c] = \mathcal{D}^{temp}$
    end while
end if
Return the updated child-parent tree collection $\mathcal{T}(\mathcal{M}_{i+1}, N)$ and the updated set $min(\mathcal{M}_{i+1})$: find $\mathcal{T}(\mathcal{M}_i, N) \cup \mathcal{T}^{collection}$ and remove all dismissed vertices along with their children to obtain $\mathcal{T}(\mathcal{M}_{i+1}, N)$. $min(\mathcal{M}_{i+1})$ is an array containing markings stored in vertices of the updated $\mathcal{T}(\mathcal{M}_{i+1}, N)$

Algorithm 3 FIND_CONTROLINVARIANT_SUBSET($min(\mathcal{M}_0)$, N).

Create an empty list $\mathcal{E}$ to store all CI estimates found
Create an empty list $\mathcal{S}$ to store all possible elevation combinations: each combination is stored as a 4-tuple $(\mathbf{m}^k, t_u, min(\mathcal{M}_i), \mathcal{T}(\mathcal{M}_i, N))$ where $\mathbf{m}^k$ is the marking to be elevated, t_u is the transition for which $\mathbf{m}^k$ is to be elevated, and $min(\mathcal{M}_i)$ and $\mathcal{T}(\mathcal{M}_i, N)$ are the corresponding CI estimate and child-parent tree collections, respectively
Initialization (n=0):
Create two empty arrays, $\mathcal{A}^{pass}$ and $\mathcal{A}^{fail}$ to partition markings of $min(\mathcal{M}_0)$ into $\mathcal{A}^{pass}$ and $\mathcal{A}^{fail}$
if $\mathcal{A}^{fail}$ is empty **then**
    $\mathcal{M}_0$ is control invariant: add $min(\mathcal{M}_0)$ to $\mathcal{E}$
else
    Create a dictionary $FailDict$ and store markings of $\mathcal{A}^{fail}$ along with their critical transitions: the key is the marking and the value is an array containing their critical transitions
    for all $\mathbf{m}^k \in FailDict.keys()$ **do**
        for all $t_u \in FailDict[\mathbf{m}^k]$ **do**
            Store $(\mathbf{m}^k, t_u, min(\mathcal{M}_0), \mathcal{T}(\mathcal{M}_0, N))$ as the first element of $\mathcal{S}$
        end for
    end for
end if
Iterative step ($n \geq 1$):
while $\mathcal{S}$ is not empty **do**
    n ← n+1
    Remove and return the first element from $\mathcal{S}$, stored as $(\mathbf{m}^k, t_u, min(\mathcal{M}_i), \mathcal{T}(\mathcal{M}_i, N))$
    Call ELEVATION_FUNCTION($\mathbf{m}^k, t_u, min(\mathcal{M}_i), \mathcal{T}(\mathcal{M}_i, N)$) and store its output as $min(\mathcal{M}_n)$ and $\mathcal{T}(\mathcal{M}_n, N)$
    if $min(\mathcal{M}_n) = \emptyset$ **then**
        $\mathcal{M}_n$ is control invariant: add $min(\mathcal{M}_n)$ to $\mathcal{E}$
    else
        Create two empty arrays, $\mathcal{A}^{pass}$ and $\mathcal{A}^{fail}$ to partition markings of $min(\mathcal{M}_n)$ into $\mathcal{A}^{pass}$ and $\mathcal{A}^{fail}$
        if $\mathcal{A}^{fail}$ is empty **then**
            $\mathcal{M}_n$ is control invariant: add $min(\mathcal{M}_n)$ to $\mathcal{E}$
        else
            Create a dictionary $FailDict$ and store markings of $\mathcal{A}^{fail}$ along with their critical transitions: the key is the marking and the value is an array containing their critical transitions
            for all $\mathbf{m}^k \in FailDict.keys()$ **do**
                for all $t_u \in FailDict[\mathbf{m}^k]$ **do**
                    Store $(\mathbf{m}^k, t_u, min(\mathcal{M}_n), \mathcal{T}(\mathcal{M}_n, N))$ as the first element of $\mathcal{S}$
                end for
            end for
        end if
    end if
end while
Return the final control invariant estimate $\mathcal{M}_0^{\Uparrow}$: find the union of arrays stored in $\mathcal{E}$

References

Alpern B, Schneider FB (1985) Defining Liveness. Inf Process Lett 21(4):181–185
Basile F, Recalde L, Chiacchio P, Silva M (2009) Closed-Loop Live Marked Graphs Under Generalized Mutual Exclusion Constraint Enforcement. Discret Event Dyn Syst 19(1):1–30
Chandrasekaran S, Somnath N, Sreenivas RS (2015) A Software Tool for the Automatic Synthesis of Minimally Restrictive Liveness Enforcing Supervisory Policies for a Class of General Petri Net models of Manufacturing- and Service-Systems. J Intell Manuf 26(5):945–958
Chen Y, Li ZW, Khalgui M, Mosbahi O (2011) Design of a Maximally Permissive Liveness-Enforcing Petri Net Supervisor for Flexible Manufacturing Systems. IEEE Trans Autom Sci Eng 8(2):374–393
Chen C, Raman A, Hu H, Sreenivas RS (2020) On Liveness Enforcing Supervisory Policies for Arbitrary Petri Nets. IEEE Transactions on Automatic Control, to appear 2020
Dideban A, Alla H (2008) Reduction of Constraints for Controller Synthesis Based on Safe Petri Nets. Automatica 44(7):1697–1706
Dideban A, Zeraatkar H (2018) Petri Net Controller Synthesis Based on Decomposed Manufacturing Models. ISA Trans 77:90–99
Iordache M, Ansaklis PJ (2006) Supervisory Control of Concurrent Systems: A Petri Net Structural Approach. Birkhaüser, Boston
Khaleghi R, Raman A, Sreenivas RS (2019) Designing Supervisory Policies that Avoid Livelocks in Manufacturing- and Service-Systems Modeled using General Petri Nets: A Tutorial using an Illustrative Example. In: Proceedings of the 9th International Conference on Industrial Engineering and Operations Management (IEOM), pp 895–902
Kumar R, Garg VK (2005) On Computation of State Avoidance Control for Infinite State Systems in Assignment Program Framework. IEEE Trans Autom Sci Eng 2(1):87–91
Luo J, Nonami K (2011) Approach for Transforming Linear Constraints on Petri Nets. IEEE Trans Autom Control 56(12):2751–2765
Moody JO, Antsaklis PJ (1998) Supervisory Control of Discrete Event Systems Using Petri Nets. Kluwer, Norwell
Murata T (1989) Petri Nets: Properties, Analysis and Applications. Proc IEEE 77(4):541–580
Peterson JL (1981) Petri Net Theory and the Modeling of Systems. Englewood Cliffs, NJ
Ramadge PJ, Wonham WM (1987) Modular Feedback Logic for Discrete Event Systems. SIAM J Control Optim 25(5):1202–1218
Salimi E, Somnath N, Sreenivas RS (2015) A Software Tool for Live-Lock Avoidance in Systems Modeled Using a Class of Petri Nets. Int J Comput Sci Appl 5(2):1–13
Sreenivas RS (1997) On the Existence of Supervisory Policies that Enforce Liveness in Discrete-Event Dynamic Systems Modeled by Controlled Petri Nets. IEEE Trans Autom Control 42(7):928–945
Sreenivas RS (2012) On the Existence of Supervisory Policies that Enforce Liveness in Partially Controlled Free-Choice Petri Nets. IEEE Trans Autom Control 57(2):435–449
Valk R, Jantzen M (1985) The Residue of Vector Sets with Applications to Decidability Problems in Petri Nets. Acta Inform 21(6):643–674

Roshanak Khaleghi is a graduate student pursuing her Ph.D. in Industrial Engineering at the Department of Industrial and Enterprise Systems Engineering at the University of Illinois at Urbana-Champaign. She completed her Bachelor of Science and Master of Science degrees in Industrial Engineering from the University of Tehran, Iran. Her research is in the area of supervisory control of DEDS systems.

Ramavarapu S. Sreenivas received the B.Tech. degree in Electrical Engineering from the Indian Institute of Technology, Madras, India, and the M.S. and Ph.D. degrees in Electrical and Computer Engineering from Carnegie Mellon University, Pittsburgh, PA. He was a Postdoctoral Fellow in Decision and Control at the Division of Applied Sciences, Harvard University, Cambridge, MA, before he joined the University of Illinois at Urbana-Champaign in 1992, where he is a Professor of Industrial and Enterprise Systems Engineering.