A SOFTWARE TOOL FOR LIVE-LOCK AVOIDANCE IN SYSTEMS MODELLED USING A CLASS OF PETRI NETS

E. Salimi[1], N. Somnath[1]and R.S. Sreenivas[1]

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

ABSTRACT

If a manufacturing system enters into a state where a task enters into a state of suspended animation for perpetuity, we say it is in a livelocked state. In contrast, all tasks of the system remain suspended for perpetuity in a deadlocked state of the system. A livelock-free manufacturing system can never experience deadlocks, but the converse is not necessarily true. A livelock-prone manufacturing system can be regulated using a supervisory policy such that the resulting supervised system is livelock-free. If a liveness enforcing supervisory policy (LESP) prevents the occurrence of an event at given state of the manufacturing system, and every other LESP, irrespective of the implementation paradigm, prescribes the same control for that state, we say the original LESP is minimally restrictive.

This paper is about two enhancements to an existing software tool that synthesizes the minimally restrictive LESP for a manufacturing system modelled using a class of weighted Petri nets (PNs). The first enhancement is about broadening the scope of the software tool to a larger class of PNs. The second enhancement is about improving the running time of the software tool using a property identified in this paper.

We identify a class, $\mathcal{H}$, of general Petri net (PN) structures where the existence of a liveness enforcing supervisory policy (LESP) for an instance of this class, initialized at a marking, is sufficient to infer the existence of an LESP when the same instance is initialized at a larger marking. As a consequence, the existence of an LESP for the PN that results when a member of this class is initialized with a marking, is decidable. Additionally, the maximally permissive LESP, when it exists, can be synthesized using a software tool described in an earlier paper. We also highlight a property that plays a critical role in the speed of convergence of the iterative procedure for the synthesis of the minimally restrictive LESP, when it exits, for any instance of $\mathcal{H}$ that uses the aforementioned software package.

KEYWORDS

Manufacturing Systems, Livelock Avoidance &Petri Nets

1. INTRODUCTION

Manufacturing systems belong to a class of systems called *Discrete-Event/Discrete-State* (DEDS) systems. The (discrete-)states of these systems have a logical, as opposed to numerical, interpretation. At each state, there are potential (discrete-)events that can occur, the occurrence of any one of them would change the state of the system, which then results in a new set of potential events, and this process can be repeated as often as necessary. DEDS systems are regulated by a *supervisory policy*, which determines which event is to be permitted at each state, in such a

DOI : 10.5121/ijcsea.2015.5201

manner that some behavioural specification is satisfied. Our focus is on *livelock-avoidance*. A manufacturing system is in a *livelocked-state* if a task enters into a state of suspended animation for perpetuity. If all tasks in the manufacturing system are prevented from progressing to completion, we say the manufacturing system has entered a *deadlocked-state* (cf. [1], for example). A livelock-free manufacturing system does not have deadlocked-states, but a deadlock-free manufacturing system can still experience livelocks. Livelock freedom is harder to achieve, compared to deadlock freedom.

Petri nets (PNs) have been extensively used to model manufacturing systems (cf. [2,3], for example). In this paper we consider the synthesis of *liveness enforcing supervisory policies* (LESPs) in PN models of manufacturing systems. The results in the literature range from heuristic procedures to provably correct methods that can synthesize LESPs for a variety of PN classes. Since the existence of an LESP in an arbitrary PN instance is not even semi-decidable, it is imperative that any provably correct scheme restrictsits attention to a sub-class of PN structures. This paper identifies one such a class, $\mathcal{H}$, which strictly includes all other classes of PNs for which a LESP can be automatically synthesized (cf. [4–6], for example). Additionally, the software tool identified in reference [7] can be used to synthesize the minimally restrictive LESP, when it exists, for any member of the class $\mathcal{H}$, thus broadening its scope. We also identify a characterization of the minimally restrictive LESP for any instance of $\mathcal{H}$, which can be used to improve the running time of the aforementioned software tool.

This paper is organized as follows – section 2 introduces the notations and definitions used in this paper. This section also reviews the supervisory control paradigm for PN models, along with a brief review of the relevant results from the literature. The main results are presented in section 3. Section 4 presents the conclusions.

2. NOTATIONS AND DEFINITIONS AND SOME PRELIMINARY OBSERVATIONS

$\mathbb{N}$ ($\mathbb{N}^+$)denotes the set of non-negative (positive) integers. The cardinality of a set A is represented as *card(A)*. A *Petri net* (PN) structure $N = (\Pi, T, \Phi, \Gamma)$ is an ordered 4-tuple, where$\Pi = \{p_1 \cdots p_n\}$is a set of *nplaces*, $T = \{t_1 \cdots t_m\}$ is a collection of*mtransitions*,$\Phi \subseteq (\Pi \times T) \cup (T \times \Pi)$isasetof*arcs*,and$\Gamma: \Phi \to \mathbb{N}^+$isthe weight associated with each arc. The weight of an arc is represented by an integer that is placed alongside the arc. If an arc has a unitary weight, it is not represented in its graphical representation in this paper.

If all arcs of a PN are unitary, it is said to be an *ordinary* PN, otherwise it is a *general* PN. The *initial marking* of a PN structure N is a function $\boldsymbol{m}^0: \Pi \to \mathbb{N}$, which identifies the number of tokens in each place. A *Petri net* (PN), $N(\mathbf{m}^0)$, is a PN structure N together with its initial marking $\mathbf{m}^0$.

T^*represents the set of all finite-length strings of transitions. For $\sigma \in T^*$, we use$\boldsymbol{x}(\sigma)$ to denote the *Parikh vector* of σ. That is, the i^{th} entry, $\boldsymbol{x_i}(\sigma)$, corresponds to the number of occurrences of transition t_i in σ.

Let $^\bullet x \coloneqq \{y | (y, x) \in \Phi\}$ and $x^\bullet \coloneqq \{y | (x, y) \in \Phi\}$. If $\forall p \in {^\bullet t}, \boldsymbol{m}^i(p) \geq \Gamma((p, t))$for some $t \in T$ and some marking $\mathbf{m}^i$, then $t \in T$ is said to be enabled at marking $\mathbf{m}^i$. The set of enabled transitions at marking $\mathbf{m}^i$ is denoted by the symbol $T_e(N, \boldsymbol{m}^i)$. An enabled transition $t \in T_e(N, \boldsymbol{m}^i)$ can fire, which changes the marking $\mathbf{m}^i$ to $\mathbf{m}^{i+1}$ according to $\boldsymbol{m}^{i+1}(p) = \boldsymbol{m}^i(p) - \Gamma(p, t) + \Gamma(t, p)$.

Asetofmarkings$\mathcal{M} \subseteq \mathbb{N}^n$issaidtobe*right-closed*[8]if$((\boldsymbol{m}^1 \in \mathcal{M}) \wedge (\boldsymbol{m}^2 \geq \boldsymbol{m}^1) \Rightarrow (\boldsymbol{m}^2 \in \mathcal{M}))$, and is uniquely defined by its finite set of minimal-elements.

When the marking is interpreted as a nonnegative integer-valued vector, it is useful to definethe input matrix **IN** (output matrix **OUT**) as an $n\times m$ matrix, where $\mathbf{IN}_{i,j}$ ($\mathbf{OUT}_{i,j}$) equals $\Gamma((p_i,t_j))$ ($\Gamma((p_i,t_j))$) if $p_i\in{}^{\bullet}t_j$, ($p_i\in t^{\bullet}_j$) and is zero-valued otherwise. The incidence matrix **C** of the PN N is an $n\times m$ matrix, where **C** = **OUT**−**IN**.

2.1. Supervisory Control of PNs

Under this paradigm, the set of transitions in the PN is partitioned into a set of *controllabletransitions* ($T_c \subseteq T$)and a set of *uncontrollable transitions* ($T_u \subseteq T$). Thecontrollable (uncontrollable) transitions are represented as filled (unfilled) boxes ingraphical representation of PNs.

A supervisory policy $\wp: \mathbb{N}^n \times T \to \{0,1\}$, is a function that returns a 0 or 1for each transition and each reachable marking. Transition $t \in T$is *control-enabled*(*state-enabled*) if $\wp(\boldsymbol{m},t) = 1$ ($t \in T_e(N,\boldsymbol{m})$) for some marking **m**. A transition that isstate- and control-enabled can fire, which results in a new marking as indicated in theprevious section. Since uncontrollable transitions cannot be prevented from firing bythe supervisory policy, we require the following condition to be true of all supervisorypolicies: $\forall \boldsymbol{m} \in \mathbb{N}^n, \wp(\boldsymbol{m},t) = 1$, if $t \in T_u$.

A *valid firing string*$\sigma = t_1t_2 \cdots t_k \in T^*$for a marking $\mathbf{m}^i$satisfies the followingconditions: (1)$t_1 \in T_e(N,\mathbf{m}^i), \wp(\mathbf{m}^i,t_1) = 1$,and (2) for $j \in \{1,2,\cdots k-1\}$the firingof transition t_j produces a marking$\mathbf{m}^{i+j}$,$t_{j+1} \in T_e(N,\mathbf{m}^{i+j})$, and $\wp(\mathbf{m}^{i+j},t_{j+1}) = 1$.

$\Re(N,\boldsymbol{m}^0,\wp)$denotes the set of markings that are reachable from $\mathbf{m}^0$ under thesupervision of $\wp$in *N*. We use $\boldsymbol{m}^i \xrightarrow{\sigma} \boldsymbol{m}^j$to denote that $\mathbf{m}^j$results from the firing of$\sigma \in T^*$from $\mathbf{m}^i$.

A transition t_k is *live* under the supervision of $\wp$ if $\forall \boldsymbol{m}^i \in \Re(N,\boldsymbol{m}^0,\wp), \exists \boldsymbol{m}^j \in \Re(N,\boldsymbol{m}^i,\wp)$such that$t_k \in T_e(N,\mathbf{m}^j)$ and $\wp(\mathbf{m}^j,t_k) = 1$. If all transitions in $N(\mathbf{m}^0)$are live under $\wp$, then it is a *liveness enforcing supervisory policy* (LESP) for $N(\mathbf{m}^0)$.The policy $\wp$ is said to be *minimally restrictive* if for every LESP$\hat{\wp}: \mathbb{N}^n \times T \to \{0,1\}$, the following condition holds$\forall \boldsymbol{m}^i \in \mathbb{N}^n, \forall\, t \in T, \wp(\mathbf{m}^i,t) \geq \hat{\wp}(\mathbf{m}^i,t)$.

There is an LESP for $N(\mathbf{m}^0)$if and only if $\boldsymbol{m}^0 \in \Delta(\boldsymbol{N})$, where$\Delta(N) = \{\boldsymbol{m}^0 \in \mathbb{N}^{card(\Pi)} | \exists\ an\ LESP\ for\ N(\boldsymbol{m}^0)\}$ is the set of initial markings $\mathbf{m}^0$ for which there is aLESP for $N(\mathbf{m}^0)$. It follows that $\Delta(N)$ is *control invariant* (cf. [9,10]) with respect to*N*; that is, if $\boldsymbol{m}^1 \in \Delta(N), t_u \in T_u \cap T_e(N,\boldsymbol{m}^1)$ and $\boldsymbol{m}^1 \xrightarrow{t_u} \boldsymbol{m}^2$in *N*, then $\boldsymbol{m}^2 \in \Delta(N)$.Equivalently, only the firing of a controllable transition at any marking in $\Delta(N)$ canresult in a new marking that is not in $\Delta(N)$.

If $\wp$ is an LESP for $N(\mathbf{m}^0)$, then $\Re(N,\boldsymbol{m}^0,\wp) \subseteq \Delta(N)$. Additionally, the LESP$\wp^*$, that prevents the firing of a controllable transition at any marking when its firingwould result in a new marking that is not in $\Delta(\boldsymbol{N})$, is the minimally restrictive LESPfor $N(\mathbf{m}^0)$. That is, there can be no other LESP, independent of the implementationparadigm, that can be better than $\wp^*$.

Neither the existence, nor the non-existence, of an LESP for an arbitrary PN issemi-decidable; the existence of an LESP is decidable if all transitions in the PNare controllable, or if the PN structure belongs to specific classes identified in theliterature [4–6,11]. The process of deciding the existence of an LESP in an arbitraryinstance from these classes is NP-hard.

2.2. Review of Relevant Prior Work

Monitors were introduced into supervisory control of PNS by Giua [12], and wereused by Moody et al. [13], and Iordache et al. [14] to derive sufficient conditions forthe existence of certain classes of PNs. Sufficient conditions for minimally restrictive,closed-loop liveness of a class Marked Graph PNs supervised by monitors were derivedby Basile et al. [15]. There are necessary and sufficient conditions for classes ofPNs known as S^3PR and S^4PR nets that can be used to synthesize liveness enforcingenhancements in instances of these classes [16,17]. Reveliotis et al. [18] and Ghaffariet al. [19] used the theory of regions to identify minimally restrictive LESPs forResource Allocation Systems. Marchetti et al. [20] presented a polynomial time sufficient condition for liveness, for the class of Unitary Weighted Event Graphs. Ferrariniet al [21] compare the performance of a selection of deadlock avoidance policies inPN models of flexible manufacturing systems. Chen et al. [22] use Integer LinearProgramming to synthesize invariant based monitors that enforce liveness in a classof PNs. Hu et al. [23,24] use a set of inequalities to characterize insufficiently markedsiphons that is subsequently used to develop an algebraic LESP-synthesis procedure.Li et al [25] develop an iterative siphon-based control scheme for preventing deadlocksin PN models of manufacturing systems using a mixed integer programmingapproach involving what are known as necessary siphons.

3. Main Results

Let,$\Omega(t) = \{\hat{t} \in T \mid {}^{\bullet}t \cap {}^{\bullet}\hat{t} \neq \emptyset\}$, denote the set of transitions that share a common inputplace with $t \in T$for a PN structure $N = (\Pi, T, \Phi, \Gamma)$. Consequently,$(t_1 \in \Omega(t_2)) \Rightarrow (t_2 \in \Omega(t_1))$. Let $\widetilde{\mathcal{H}}$ denote a class of PN structures where the following property is true:

$$\forall \boldsymbol{m} \in \Delta(N), \forall t_u \in T_u, \forall t \in \Omega(t_u), \big(t \in T_e(N, \boldsymbol{m})\big) \Rightarrow \big(t_u \in T_e(N, \boldsymbol{m})\big). \quad (1)$$

That is, $\widetilde{\mathcal{H}}$ is a class of PN structures where, if a transition *t* is state-enabled, then all uncontrollable transitions that share a common input place with *t* are also state enabledat any marking in $\Delta(N)$. The following lemma finds use in the proof of theorem2.

Lemma 1: (*Lemma 5.1, [5]*) Let $\wp$ be a LESP for $N(\mathbf{m}^0)$, where $\boldsymbol{m}^0 \in \Delta(N)$, for aPN structure $N \in \widetilde{\mathcal{H}}$. Suppose $\boldsymbol{m}^0 \xrightarrow{\sigma} \boldsymbol{m}^i$ under the supervision of $\wp$, and$\widehat{\boldsymbol{m}}^0 \xrightarrow{\hat{\sigma}} \widehat{\boldsymbol{m}}^j$ without supervision in *N*, where the number of occurrences of each controllabletransition in σ and $\hat{\sigma}$ are identical, and$\widehat{\boldsymbol{m}}^0 \geq \boldsymbol{m}^0$. There exists strings $\sigma_1, \hat{\sigma}_1 \in T^*$such that (1) $\boldsymbol{m}^0 \xrightarrow{\sigma\sigma_1} \boldsymbol{m}^k$under the supervision of $\wp$ in *N*, (2)$\widehat{\boldsymbol{m}}^0 \xrightarrow{\hat{\sigma}\hat{\sigma}_1} \widehat{\boldsymbol{m}}^l$ withoutsupervision in *N*, and (3) $\boldsymbol{x}(\sigma\sigma_1) = \boldsymbol{x}(\hat{\sigma}\hat{\sigma}_1)$. Consequently,$\widehat{\boldsymbol{m}}^l \geq \widehat{\boldsymbol{m}}^k$.

Proof: Let $\hat{T}_u \subseteq T_u$denote the set of uncontrollable transitions that appear more oftenin $\hat{\sigma}$as compared to σ. If $\hat{T}_u = \emptyset$, then$\hat{\sigma} = \sigma$ and the result holds trivially. If $\hat{T}_u \neq \emptyset$,,there is a stringσ_1 such $\boldsymbol{m}^i \xrightarrow{\sigma_1} \boldsymbol{m}^{i+1}$ under the supervision of the LESP $\wp$ suchthat (1) at least one member of $t_u \in \hat{T}_u$ is state-enabled at$\boldsymbol{m}^{i+1}$, and (2) none ofthe members of $\hat{T}_u$ are state-enabled at any marking that results from the firing of aproper prefix of σ_1 at$\boldsymbol{m}^i$. It follows that $\widehat{\boldsymbol{m}}^j \xrightarrow{\sigma_1} \widehat{\boldsymbol{m}}^{j+1}$, without any supervision, in *N*.If this were not the case, there must be a proper prefix of σ_1, of the form $\bar{\sigma}t_m$, suchthat $\widehat{\boldsymbol{m}}^j \xrightarrow{\bar{\sigma}} \overline{\boldsymbol{m}}$ in *N*, but$t_m \notin T_e(N, \overline{\boldsymbol{m}})$. Additionally, $t_m \in \Omega(\bar{t}_u)$for some $\bar{t}_u \in \hat{T}$. Since$N \in \widetilde{\mathcal{H}}$, and $\overline{\boldsymbol{m}} \in \Delta(N)$, it follows that $t_u \in T_e(N, \overline{\boldsymbol{m}})$, which contradicts requirement (2).

Suppose $\boldsymbol{m}^i \xrightarrow{\sigma_1 t_u} \boldsymbol{m}^{j+1}$ under $\wp$ in *N*, and $\widehat{\boldsymbol{m}}^j \xrightarrow{\sigma_1} \widehat{\boldsymbol{m}}^{j+1}$ without supervision in*N*. We let $\boldsymbol{m}^j \leftarrow \boldsymbol{m}^{j+1}, \widehat{\boldsymbol{m}}^j \leftarrow \widehat{\boldsymbol{m}}^{j+1}, \sigma \leftarrow \sigma\sigma_1 t_u$,and$\hat{\sigma} \leftarrow \sigma\sigma_1 t_u$. The result followsby repeating the above construction as often as necessary till $\hat{T}_u = \phi$. ♣

The following theorem notes that $\Delta(N)$ is right-closed if $N \in \widetilde{\mathcal{H}}$.

Theorem 2: $\big((\boldsymbol{m}^0 \in \Delta(N)) \wedge (\widehat{\boldsymbol{m}}^0 \geq \boldsymbol{m}^0)\big) \Rightarrow \big(\widehat{\boldsymbol{m}}^0 \in \Delta(N)\big)$, if $N \in \widetilde{\mathcal{H}}$.

Proof: Since $\boldsymbol{m}^0 \in \Delta(N)$, there is an LESP $\wp$ for $N(\mathbf{m}^0)$. Following reference [5], we define an LESP $\widehat{\wp}$ for $N(\widehat{\boldsymbol{m}}^0)$ as follows (1) $\forall t \in T, \widehat{\wp}(\widehat{\boldsymbol{m}}^0, t) = \wp(\boldsymbol{m}^0, t)$, (2) if $\widehat{\boldsymbol{m}}^0 \xrightarrow{\widehat{\sigma}} \widehat{\boldsymbol{m}}^i$ under $\widehat{\wp}$, then (2a) $\forall t_u \in T_u, \widehat{\wp}(\widehat{\boldsymbol{m}}^i, t_u) = 1$, and (2b) $\forall t_c \in T_c, \big(\widehat{\wp}(\widehat{\boldsymbol{m}}^i, t_c) = 1\big) \Leftrightarrow (\exists \sigma \in T^*$, such that $\boldsymbol{m}^0 \xrightarrow{\sigma} \boldsymbol{m}^k$ under $\wp$, and the number of occurrences of each controllable transition in σ and $\widehat{\sigma} t_c$ are identical).

If $\widehat{\boldsymbol{m}}^0 \xrightarrow{\widehat{\sigma}} \widehat{\boldsymbol{m}}^i$ under $\widehat{\wp}$, then $\exists \sigma \in T^*$ such that $\boldsymbol{m}^0 \xrightarrow{\sigma} \boldsymbol{m}^j$ under $\wp$, and the number of occurrences of each controllable transition in σ and $\widehat{\sigma}$ are identical. Using lemma 1, and the definition of $\widehat{\wp}$, we know $\exists \widehat{\sigma}_1, \sigma_1 \in T^*$ such that $\widehat{\boldsymbol{m}}^0 \xrightarrow{\widehat{\sigma}\widehat{\sigma}_1} \widehat{\boldsymbol{m}}^{i+1}$ under $\widehat{\wp}$, $\boldsymbol{m}^0 \xrightarrow{\sigma\sigma_1} \boldsymbol{m}^{j+1}$ under $\wp$, and $\widehat{\boldsymbol{m}}^{i+1} \geq \boldsymbol{m}^{j+1}$. Consequently, for any $\sigma_2 \in T^*$ such that $\boldsymbol{m}^{j+1} \xrightarrow{\sigma_2} \boldsymbol{m}^{j+2}$ under $\wp$, we have $\widehat{\boldsymbol{m}}^{i+1} \xrightarrow{\sigma_2} \widehat{\boldsymbol{m}}^{i+2}$ under $\widehat{\wp}$ as well. Since $\wp$ is an LESP for $N(\mathbf{m}^0)$, it follows that $\widehat{\wp}$ is an LESP for $N(\widehat{\boldsymbol{m}}^0)$. ♣

Lemma 1 and theorem 2 together imply the following theorem.

Theorem 3: $\Delta(N)$ is right-closed if $N \in \widetilde{\mathcal{H}}$.

The above condition is not necessary for the right-closure of $\Delta(N)$. For instance, $\Delta(N_1)$ is right-closed for the general PN N_1 shown in figure 1(a), but $N_1 \notin \widetilde{\mathcal{H}}$. Specifically, $\Delta(N_1)$ is identified by the inequality $(1\ 1\ 1\ 1\ 1)\mathbf{m} \geq 1$, and $\mathbf{m} = (1\ 0\ 0 0 0)^T \in \Delta(N_1)$, $t_2 \in T_u, t_3 \in T_e(N_1, \boldsymbol{m})$, but $t_2 \notin T_e(N_1, \boldsymbol{m})$.

There is an LESP for the PN $N(\mathbf{m}^0)$ if and only if $\boldsymbol{m}^0 \in \Delta(N)$, and the existence of an LESP is undecidable for a general PN (cf. corollary 5.2, [11]). This would mean that the set $\Delta(N)$ cannot be computed for an arbitrary PN structure *N*. To overcome this limitation, we modify the requirement of equation 1 as

$$\forall \boldsymbol{m} \in \mathbb{N}^n, \forall t_u \in T_u, \forall t \in \Omega(t_u), \big(t \in T_e(N, \boldsymbol{m})\big) \Rightarrow \big(t_u \in T_e(N, \boldsymbol{m})\big) \qquad (2)$$

This requirement defines a class of PNs, which we denote as $\mathcal{H} (\subseteq \widetilde{\mathcal{H}})$, and from theorem 3, we conclude $\Delta(N)$ is right-closed for any $N \in \mathcal{H}$.

Theorem 4: A PN structure $N = (\Pi, T, \Phi, \Gamma)$ belongs to the class $\mathcal{H}$ if and only if $\forall p \in \Pi, \forall t_u \in p^\bullet \cap T_u$,

$$\left(\Gamma(p, t_u) = \min_{t \in p^\bullet} \Gamma(p, t)\right) \wedge (\forall t \in \Omega(t_u), {}^\bullet t_u \subseteq {}^\bullet t).$$

Proof: (*If*) Suppose, $t \in T_e(N, \boldsymbol{m})$ for $\boldsymbol{m} \in \mathbb{N}^n$, and $\exists t_u \in \Omega(t) \cap T_u (\Rightarrow t \in \Omega(t_u))$. Since ${}^\bullet t_u \subseteq {}^\bullet t$ and $\forall p \in {}^\bullet t_u, \Gamma(p, t_u) = min_{t \in p^\bullet} \Gamma(p, t)$, it follows that $t_u \in T_e(N, \boldsymbol{m})$.

(*Only If*) We will show that the violation of requirement in the statement of the theorem for a PN structure *N* would imply that $N \notin \mathcal{H}$. Suppose $\exists p \in \Pi, \exists t_u \in p^\bullet \cap T_u$ such that either

1. $\Gamma(p, t_u) > min_{t \in p^\bullet} \Gamma(p, t)$, or

2. $\exists t \in \Omega(t_u),\ {}^{\bullet}t_u - {}^{\bullet}t \neq \emptyset$

In each of these cases we construct a marking $\boldsymbol{m} \in \mathbb{N}^n$ such that $\exists t \in \Omega(t_u) \cap T_e(N, \boldsymbol{m})$ and $t_u \notin T_e(N, \boldsymbol{m})$, which leads to the conclusion that $N \notin \mathcal{H}$.

For the first-case, the marking **m** places exactly $(min_{t \in p^{\bullet}} \Gamma(p, t))$-many tokensin p, and sufficient tokens in the input places of any transition $\hat{t} \in \Omega(t_u)$ such that $\Gamma(p, \hat{t}) = min_{t \in p^{\bullet}} \Gamma(p, t)$ that will result in $\hat{t} \in T_e(N, \boldsymbol{m})$ and $t_u \notin T_e(N, \boldsymbol{m})$.

Similarly, for the second-case, the marking **m**places sufficient tokens in the inputplaces of t such that $t \in T_e(N, \boldsymbol{m})$, while ensuring that the places in $({}^{\bullet}t_u - {}^{\bullet}t)$ remain empty. Consequently, $t \in T_e(N, \boldsymbol{m})$ and $t_u \notin T_e(N, \boldsymbol{m})$.
♣

There is an $O(n^2m^2)$ algorithm that decides if an arbitrary PN structure belongs to the class $\mathcal{H}$, where $n = card(\Pi)$ and $m = card(T)$. The right-closure of $\Delta(N)$ for any $N \in \mathcal{H}$, along with the results in reference [5] implies that the existenceof an LESP for $N(\mathbf{m}^0)$ is decidable. Furthermore, the software package described inreference [7] can be used to compute the minimally restrictive LESP for $N(\mathbf{m}^0)$, whenit exists. As noted in the introduction section, each decidable class of PN structuresidentified in references [4–6] are strictly contained in the class $\mathcal{H}$. As an illustration, the PN structure N_2 shown in figure 1(b) is a member of $\mathcal{H}$ as it meets the structuralrequirements of theorem 4, and it does not belong to any of the classes of structuresidentified in references [4–6]. Additionally,

$$\min(\Delta(N_2)) = \{(0\ 0\ 0\ 1\ 0)^T, (1\ 0\ 1\ 0\ 2)^T, (0\ 0\ 2\ 0\ 2)^T, (2\ 0\ 0\ 0\ 2)^T, (1\ 1\ 1\ 0\ 1)^T, (0\ 1\ 2\ 0\ 1)^T, (2\ 1\ 0\ 0\ 1)^T, (0\ 2\ 2\ 0\ 0)^T, (1\ 2\ 1\ 0\ 0)^T, (2\ 2\ 0\ 0\ 0)^T\}$$

There is an LESP for $N_2(\boldsymbol{m}_2^0)$ if and only if $\boldsymbol{m}_2^0 \in \Delta(N_2)$.

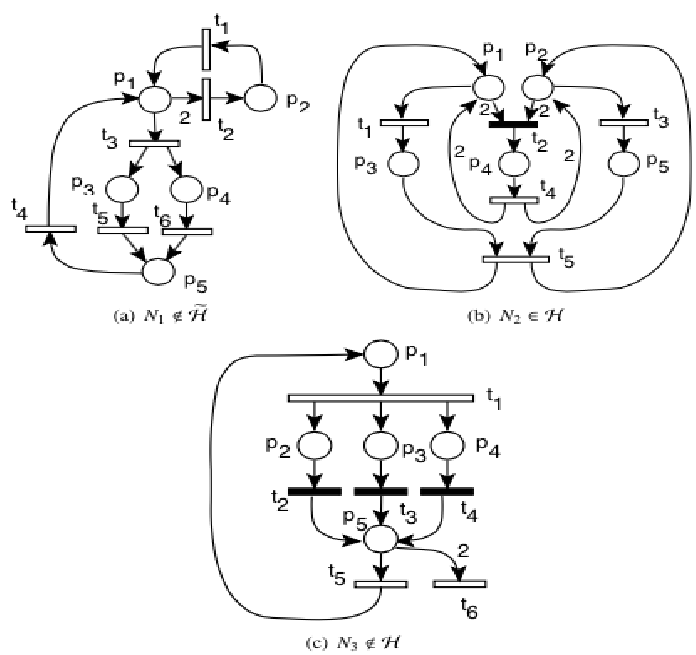

Figure 1. (a) The PN structure N_1 is not a member of the class $\widetilde{\mathcal{H}}$. However, $\Delta(N_1)$ is right-closed (cf. figure1, [4]). (b) The PN structure N_2 is a member of $\mathcal{H}$ as it meets the structural requirements of theorem 4(cf. figure 2a, [6]). (c) The PN structure N_3 is not a member of the class $\mathcal{H}$, and $\Delta(N_3) = \{\boldsymbol{m} \in \mathbb{N}^5 | (\boldsymbol{m}(p_1) + \boldsymbol{m}(p_2) + \boldsymbol{m}(p_3) + \boldsymbol{m}(p_4) \geq 1) \vee (\boldsymbol{m}(p_5)_{mod2} = 1)\}$ is not right-closed.

A transition$t \in T$ is said to be a *choice-transition* (resp. *non-choice transition*) if $({}^{\bullet}t)^{\bullet} \neq \{t\}$ (resp. $({}^{\bullet}t)^{\bullet} = \{t\}$). In reference [26] it is shown that the minimally restrictive LESP for a class of ordinary PNs called *Free-choice* PNs does not control-disable a non-choice (controllable) transition. The following result shows that a similar observation holds for any minimally restrictive LESP for $N(\mathbf{m}^0)$ where $N \in \mathcal{H}$.

Theorem 5: [26] Suppose $\boldsymbol{m}^0 \in \Delta(N)$ for a PN $N(\mathbf{m}^0)$, where $N \in \mathcal{H}$, then theminimally restrictive LESP $\wp^*$ for $N(\mathbf{m}^0)$ does not disable any controllable transition$t_c \in T_c$ that satisfies the requirement $({}^{\bullet}t_c)^{\bullet} = \{t_c\}$.

Proof: (*Sketch*) Suppose $\exists \boldsymbol{m}^1 \in \Re(N, \boldsymbol{m}^0, \wp^*)(\subseteq \Delta(N)), \exists \boldsymbol{m}^2 \in \Re(N, \boldsymbol{m}^0)$, such that$\boldsymbol{m}^1 \xrightarrow{t_c} \boldsymbol{m}^2$ in N for some $t_c \in T_C$ that satisfies the requirement $({}^{\bullet}t_c)^{\bullet} = \{t_c\}$. We willshow that

1. $\exists \widetilde{\omega} \in T^*$such that $\boldsymbol{m}^2 \xrightarrow{\widetilde{\omega}} \widetilde{\boldsymbol{m}}$ in N, where $\widetilde{\boldsymbol{m}} \in \Delta(N)$.
2. Additionally, if $\widetilde{\omega} = \widetilde{\omega}_1\widetilde{\omega}_2, \boldsymbol{m}^2 \xrightarrow{\widetilde{\omega}_1} \widehat{\boldsymbol{m}}^1 \xrightarrow{\widetilde{\omega}_2} \widetilde{\boldsymbol{m}}$, and$\widehat{\boldsymbol{m}}^1 \xrightarrow{t_u} \widehat{\boldsymbol{m}}^2$ in N for some$t_u \in T_u$, then $\exists \widehat{\omega} \in T^*, \exists \widehat{\boldsymbol{m}}^3 \in \Re(N, \widehat{\boldsymbol{m}}^2)$, such that$\widehat{\boldsymbol{m}}^1 \xrightarrow{t_u} \widehat{\boldsymbol{m}}^2 \xrightarrow{\widehat{\omega}} \widehat{\boldsymbol{m}}^3and\widehat{\boldsymbol{m}}^3 \in \Delta(N)$.

Following the repeated application of the above observation, we conclude that$\boldsymbol{m}^2 \in \Delta(N)$.

Since $\wp^*$ is an LESP, $\exists \omega_1 \in (T - \{t_c\})^*$, and$\boldsymbol{m}^1 \xrightarrow{\omega_1} \boldsymbol{m}^3 \xrightarrow{t_c} \boldsymbol{m}^4$ in N underthe supervision of$\wp^*$. Since$({}^{\bullet}t_c)^{\bullet} = \{t_c\}$, it follows that $\boldsymbol{m}^2 \xrightarrow{\omega_1} \boldsymbol{m}^4$in N, and$\{\boldsymbol{m}^1, \boldsymbol{m}^3, \boldsymbol{m}^4\} \subseteq \Delta(N)$(cf. figure 2(a)). Suppose$\omega_1 = \omega_2\omega_3, \boldsymbol{m}^2 \xrightarrow{\omega_2} \boldsymbol{m}^9 \xrightarrow{\omega_3} \boldsymbol{m}^4$,and $t_u \in T_u$ such that$\boldsymbol{m}^9 \xrightarrow{t_u} \boldsymbol{m}^{10}$. Also,$\exists \boldsymbol{m}^2 (\in \Delta(N))$ such that$\boldsymbol{m}^1 \xrightarrow{\omega_2} \boldsymbol{m}^5$. There aretwo cases to consider – (i) $t_u \in T_e(N, \boldsymbol{m}^5)$, and (ii) $t_u \notin T_e(N, \boldsymbol{m}^5)$.

In the first case, $\exists \boldsymbol{m}^6 (\in \Delta(N))$ such that $\boldsymbol{m}^5 \xrightarrow{t_u} \boldsymbol{m}^6$ (cf. figure 2(b)). Since $\wp^*$ isan LESP, $\exists \omega_4 \in (T - \{t_c\})^*, \exists \boldsymbol{m}^7, \boldsymbol{m}^8 \in \Delta(N)$,such that $\boldsymbol{m}^6 \xrightarrow{\omega_4} \boldsymbol{m}^7 \xrightarrow{t_c} \boldsymbol{m}^8$. Since $({}^{\bullet}t_c)^{\bullet} = \{t_c\}$, we have $\boldsymbol{m}^{10} \xrightarrow{\omega_4} \boldsymbol{m}^4$, where$\boldsymbol{m}^4 \in \Delta(N)$.

For the second case where $t_u \notin T_e(N, \boldsymbol{m}^5)$, since $t_u \in T_e(N, \boldsymbol{m}^9)$, it follows that$\exists p \in \Pi$ such that$\{(t_c, p), (p, t_u)\} \subseteq \Phi$, and the prior-firing of t_c is necessary to place sufficient tokens in $p \in \Pi$, for t_u to be state-enabled at $\boldsymbol{m}^5$ (cf. figure 2(c)). Since$N \in \mathcal{H}$, it follows that none of the transitions in $\Omega(t_u)$ can fire at any marking thatis reachable in the segment identified by $\boldsymbol{m}^1 \xrightarrow{\omega_1} \boldsymbol{m}^3$. Consequently, $t_u \in T_e(N, \boldsymbol{m}^4)$,and $\boldsymbol{m}^4 \xrightarrow{t_u} \boldsymbol{m}^{11}$ under the supervision of $\wp^*$, where $\boldsymbol{m}^{11} \in \Delta(N)$. Consequently,$\boldsymbol{m}^{10} \xrightarrow{\omega_4} \boldsymbol{m}^{11}$. ♣

Theorem 5 does not hold for general PN structures. The PN structure N_3 shownin figure 1(c) does not belong to the class $\mathcal{H}$. This is becausemin$_{t \in p_5^{\bullet}} \Gamma(p_5, t) = 1$,while $\Gamma(p_5, t_6) = 2$, and$t_6 \in p_5^{\bullet} \cap T_u$. dditionally, $\Delta(N_3) = \{\boldsymbol{m} \in \mathbb{N}^5 | (\boldsymbol{m}(p_1) + \boldsymbol{m}(p_2) + \boldsymbol{m}(p_3) + \boldsymbol{m}(p_4) \geq 1) \vee (\boldsymbol{m}(p_5)_{mod2} = 1)\}$, which is not right-closed. Theminimally restrictive LESP for$N_3()$for any $\boldsymbol{m}_3^0 \in \Delta(N_3)$ control-disables a controllabletransition at a marking in $\Delta(N_3)$ only if its firing results in a new marking thatis not in $\Delta(N_3)$. The minimally restrictive LESP would control-disable the non-choicetransition $t_2 \in T_c$ at the marking$(0\ 1\ 0\ 0\ 1)^T \in \Delta(N_3)$.

As a consequence of theorem 5, without loss of generality, we can assume all non-choice transitions are uncontrollable, even when they are not, for any instance ofthe class of PN structures $\mathcal{H}$. This is formally stated in the following theorem.

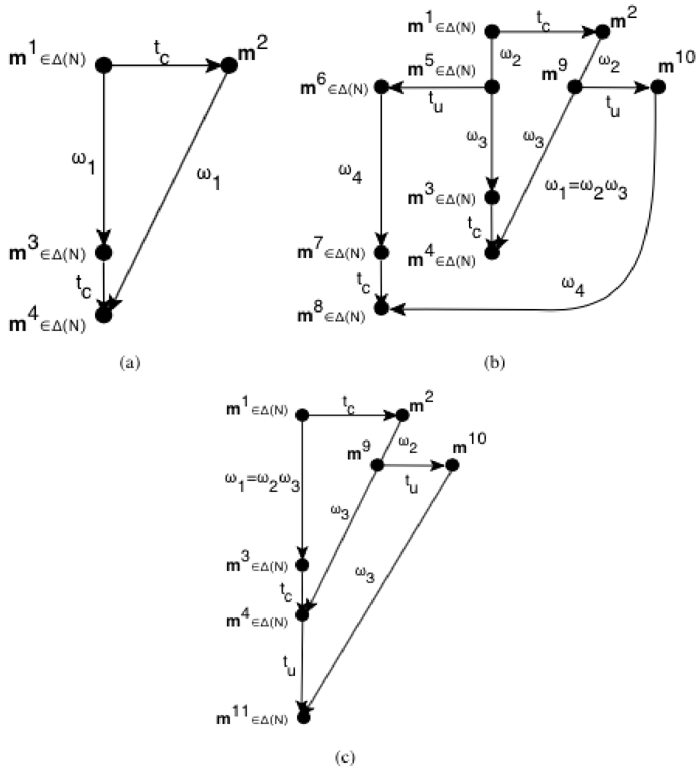

Figure 2. A graphical illustration used in the proof of theorem 5.

Theorem 6: Let $N = (\Pi, T, \Phi, \Gamma)$ be a PN structure from the family$\mathcal{H}$, where the set of transitions is partitioned into the set of uncontrollable transitions T_u, and controllabletransitions T_c (i.e. $T = T_c \cup T_u$and$T_u \cap T_c = \emptyset$). Suppose $\widehat{N}$is another memberof the family $\mathcal{H}$ that is structurally identical to *N*, but the set of transitions in $\widehat{N}$ arepartitioned into a different set of uncontrollable- and controllable-transitions, where

$$\hat{T}_u = T_u \cup \{t \in T | \left({}^{\bullet}t \right)^{\bullet} = \{t\}\},$$

and$\hat{T}_c = T - \hat{T}_u$. Then $\Delta(N) = \Delta(\widehat{N})$.

Proof: Since$\hat{T}_c \subseteq T_c$, it follows that $\Delta(\widehat{N}) \subseteq \Delta(N)$. The reverse inclusion is shown bycontradiction. Suppose,$\Delta(\widehat{N}) \subset \Delta(N)$, then$\Delta(N)$ is not control invariant with respectto $\widehat{N}$. That is, $\exists \boldsymbol{m}^1 \in \Delta(N), \exists \hat{t}_u \in \hat{T}_u$such that $\boldsymbol{m}^1 \overset{t_u}{\to} \boldsymbol{m}^2$ and $\boldsymbol{m}^2 \notin \Delta(N)$. Since $\subseteq \Delta(N)$is control invariant with respect to *N*, it must be that$\hat{t} \in \{t \in T | \left({}^{\bullet}t \right)^{\bullet} = \{t\}\}$. But,from theorem 5, we know that$\boldsymbol{m}^2 \in \Delta(N)$, which establishes the result. ♣

As an illustration, the non-choice, controllable transition $t_2$in the PN structure $N_2$of figure 1(b) can be considered to be uncontrollable, which effectively results in aPN structure with no controllable transitions. There is an LESP for a PN $N(\mathbf{m}^0)$withno controllable transitions if and only if the PN is live. This leads to the observationthat the PN$N_2(\boldsymbol{m}_2^0)$is live for any$\boldsymbol{m}_2^0 \in \Delta(N_2)$.

The observation that we can assume all non-choice transitions are uncontrollable,even when they are not for any $N \in \mathcal{H}$, is critical to the speeding-up the execution ofthe software package described in reference [7]. This is illustrated in subsequent text.

The PN structures N_4 and N_5 shown in figure 3(a) and 3(b) are FCPN structures,and consequently they belong to the class $\mathcal{H}$. The only difference between them isthat the non-choice transition t_5 is controllable (resp. uncontrollable) in N_4 (resp. N_5).

As a consequence of theorem 6, the sets $\Delta(N_4)$ and $\Delta(N_5)$ are identical, and areidentified by the twenty-four minimal elements shown in figure 4, which shows theoutput generated by the above mentioned software for N_5. The algorithm in references[5,7], finds a series of outer-approximations Ψ_i for $\Delta(N)$ for an appropriate PNstructure *N*, that are control invariant with respect to *N*.

The iteration starts with Ψ_0, the largest controllable, right-closed subset of the setof initial markings for which there is an LESP for the fully-controlled version of *N*.In the context of this example, eight minimal elements identify the right-closed ofinitial markings for which there is an LESP for the fully-controllable version of N_5shown in figure 4. The second and third among this list of eight minimal elements arenot control invariant as$t_5, t_7 \in T_u$ and $(0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0)^T \overset{t_5}{\rightarrow} (0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0)^T \overset{t_7}{\rightarrow} (0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0)^T$.

The largest controllable subset of this right-closed set is Ψ_0,which is identified by the six minimal elements shown immediately afterwards in thesame figure.

For any right-closed set of markings Ψ that is control invariant with respect toa PN structure *N*, we can envisage a supervisory policy $\wp_\Psi$ that disables the firingof a controllable transition at a marking if its firing would result in a new marking that is not in Ψ. It is possible to construct the coverability graph for the PN $N(\mathbf{m}^0)$,under the influence of this supervisory policy. The loop-test procedure of reference[7] checks if there is a closed-path identified by $\sigma \in T^*$ in this coverability graph,where (1) every transition in *T* appears at least once in σ, and (2) $\mathbf{Cx}(\sigma) \geq \mathbf{0}$, that is,the net token-load change after the execution of the string σ is non-negative.

With reference to the examples at hand, the *loop-test* checks if the above mentioned path-conditionis satisfied in the coverability graph that is generated by each minimal element of$\Psi_i$under the influence of the supervisory policy $\wp_\Psi$. If a minimal element fails this test,it is elevated by a set of unit-vectors, which defines a right-closed proper subset ofΨ_i. The largest controllable subset of this right-closed set is Ψ_{i+1}, which used in lieu of Ψ_i in the next iteration.

As shown in figure 4, four minimal elements,

$$(1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0)^T, (0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0)^T, (0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0)^T, \text{and} (0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1)^T$$

that define Ψ_0 for N_5, fail this test. The loop-test will fail for the first minimal element$(1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0)^T \in \min(\Psi_0)$, as $T_e(N_5, (1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0)^T) = \{t_1, t_3\} (\subseteq T_c)$. But,

$$(1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0)^T \overset{t_1}{\rightarrow} (0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0)^T \text{and}$$
$$(1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0)^T \overset{t_3}{\rightarrow} (0\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ 0)^T.$$

Since, $(0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0)^T, (0\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ 0)^T \notin \Psi_0$, the supervisorypolicy $\wp_{\Psi_0}$would disable these transitions at the marking $(1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0)^T$,which effectively creates a policy-induced deadlock state. The test fails for second,third and fourth minimal elements

$$(0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0)^T, (0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0)^T, (0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1)^T \in \min(\Psi_0),$$

as the marking $(1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0)^T$ is inevitably reached after the firing of an appropriate set of transitions. Specifically,

$$(0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0)^T \xrightarrow{t_9 t_{11}} (1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0)^T,$$
$$(0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0)^T \xrightarrow{t_{10} t_{11}} (1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0)^T,$$
$$\text{and } (0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1)^T \xrightarrow{t_{11}} (1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0)^T.$$

Since the marking $(1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0)^T$ failed the loop-test,it follows that these three marking would fail the test, as well.

The four minimal elements, that failed the loop-test, are elevated by nine unitvectors, and the largest controllable, right-closed set of this newly constructed set isidentified by the twenty-four minimal elements shown in figure 4, which identifiesthe next iterate Ψ_1. Each of these twenty-four minimal elements pass the loop-testreferred to earlier, implying that $\Delta(N_5) = \Psi_1$. From theorem 6, we infer $\Delta(N_4) = \Psi_1$,as well.

We turn our attention to the iteration scheme for N_4 where t_5 is left as a controllabletransition. The right-closed set of initial markings for which there is an LESPfor the fully-controlled version of N_4 is identified by the same set of eight minimalelements shown in the initial part of the output of figure 4. The largest controllablesubset of this set (Ψ_0) is identified by the six minimal elements of figure 4 along withthe vector $(0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0)^T$. This extra minimal element is due to the fact thatt_5 is controllable in N_4, which fails the loop-test along with the four that failed thetest in figure 4. After the elevation by unit-vectors as described above, the next iterateΨ_1 has the twenty-four minimal elements shown in figure 4 together with eight newelements

$$(1\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0)^T, (0\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0)^T, (0\ 0\ 2\ 0\ 0\ 0\ 0\ 0\ 0)^T, (0\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ 0)^T,$$
$$(0\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 0)^T, (0\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 0)^T, (0\ 0\ 1\ 0\ 0\ 0\ 0\ 1\ 0)^T, (0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 1)^T$$

That is, the minimal element $(0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0)^T$ of $\Psi_0$is replaced by these eightelements, which defines Ψ_1. These eight elements fail the loop-test, and are replacedwith additional elevated vectors, and so on. The right-closed set that is defined by thisiteration scheme, in the limit, is the set $\Delta(N_5)$ described earlier. *But, as a computationscheme this procedure will not terminate*. This issue is mitigated by ensuringthat all controllable, non-choice transitions are interpreted as being a part of the setof uncontrollable transitions. This has theoretical sanction as per theorems 5 and 6.There are other examples where the improvement in runtime is not as dramatic as thisillustrative example.

The method *isNetFreeChoice()* within the class *PetriNet* of the previous version of the code (cf. reference [7]),is replaced by *isNetHClass()* instead. The program terminates if the PN structureunder consideration does not belong to the class H. If the the PN structure belongsto the class $\mathcal{H}$, the method *relabelNonChoiceTransitions()* is used to relabel all controllablenon-choice transitions in the PN structure as uncontrollable transitions. Following this, the procedures outlined in reference [7] are executed to synthesize theminimally restrictive LESP for the PN structure at hand.

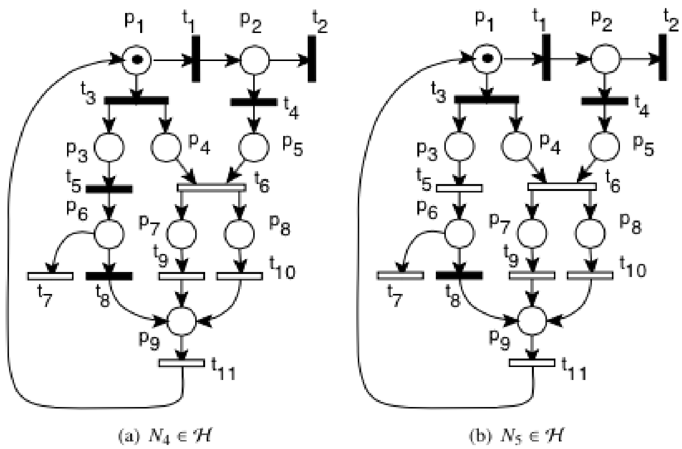

Figure 3. (a) The PN structure N_4 is a member of $\mathcal{H}$ as it is an FCPN structure. (b) The PN structure $N_5$is also a member of $\mathcal{H}$. The non-choice transition t_5 is controllable (resp. uncontrollable) in N_4 (resp. N_5). Fromtheorem 6, we infer that $\Delta(N_4) = \Delta(N_5)$.

4. Concluding Remarks

If some process in a manufacturing system enters into a state of suspended animationfor perpetuity, while other events proceed towards completion with no impediment,we say the system is in a livelocked-state. Procedures that can synthesize supervisorypolicies that can prevent the manufacturing system from entering into livelockedstatesare highly desirable.

In this paper we identified two enhancements to the software tool [7] that synthesizesthe minimally restrictive, *liveness enforcing supervisory policy* (LESP) for aclass of manufacturing systems modeled using a class of weighted *Petri nets* (PNs).

For the first enhancement, we identified a class of PN structures, $\mathcal{H}$, that includesall known classes of PN structures where the existence of a LESP for an instanceinitialized at a marking is sufficient to conclude that there is an LESP when the sameinstance is initialized with a larger marking. This broadens the scope of the softwaretool of reference [7]. For the second enhancement, we showed that the minimallyrestrictive LESP does not control-disable non-choice transitions in the PN model ofthe manufacturing system. This observation plays a crucial role in improving thespeed of convergence of the iterative scheme used in the software described above,which was illustrated by an example.

```
pn5.res        Tue Nov 26 04:30:44 2013        1
Input File = "pn5"

Incidence Matrix :

   T   1   2   3   4   5   6   7   8   9  10  11
P
   1  -1   .  -1   .   .   .   .   .   .   .   1
   2   1  -1   .  -1   .   .   .   .   .   .   .
   3   .   .   1   .  -1   .   .   .   .   .   .
   4   .   .   1   .   .  -1   .   .   .   .   .
   5   .   .   .   1   .  -1   .   .   .   .   .
   6   .   .   .   .   1   .  -1  -1   .   .   .
   7   .   .   .   .   .   1   .   .  -1   .   .
   8   .   .   .   .   .   1   .   .   .  -1   .
   9   .   .   .   .   .   .   .   1   1   1  -1

Initial Marking : ( 1 0 0 0 0 0 0 0 0 )

There is an LESP for this (fully controlled) PN

________________________________________________


Minimal Elements of the fully controlled Net
============================================

 1: ( 1 0 0 0 0 0 0 0 0 )
 2: ( 0 0 1 0 0 0 0 0 0 )
 3: ( 0 0 0 0 0 1 0 0 0 )
 4: ( 0 0 0 0 0 0 1 0 0 )
 5: ( 0 0 0 0 0 0 0 1 0 )
 6: ( 0 0 0 0 0 0 0 0 1 )
 7: ( 0 0 0 1 1 0 0 0 0 )
 8: ( 0 1 0 1 0 0 0 0 0 )


List of Controllable Transitions
--------------------------------
t1 t2 t3 t4 t8

(Final) Minimal Elements of the control-invariant set
-----------------------------------------------------

 1: ( 1 0 0 0 0 0 0 0 0 )
 2: ( 0 0 0 0 0 0 1 0 0 )
 3: ( 0 0 0 0 0 0 0 1 0 )
 4: ( 0 0 0 0 0 0 0 0 1 )
 5: ( 0 0 0 1 1 0 0 0 0 )
 6: ( 0 1 0 1 0 0 0 0 0 )

The loop-test failed for the minimal_element: ( 1 0 0 0 0 0 0 0 0 )

The loop-test failed for the minimal_element: ( 0 0 0 0 0 0 1 0 0 )

The loop-test failed for the minimal_element: ( 0 0 0 0 0 0 0 1 0 )

The loop-test failed for the minimal_element: ( 0 0 0 0 0 0 0 0 1 )

(Final) Minimal Elements of the control-invariant set
-----------------------------------------------------

 1: ( 0 0 0 1 1 0 0 0 0 )
 2: ( 0 1 0 1 0 0 0 0 0 )
 3: ( 2 0 0 0 0 0 0 0 0 )
 4: ( 1 1 0 0 0 0 0 0 0 )
 5: ( 1 0 1 0 0 0 0 0 0 )
 6: ( 1 0 0 1 0 0 0 0 0 )
 7: ( 1 0 0 0 1 0 0 0 0 )
 8: ( 1 0 0 0 0 1 0 0 0 )
 9: ( 1 0 0 0 0 0 1 0 0 )
10: ( 1 0 0 0 0 0 0 1 0 )
11: ( 1 0 0 0 0 0 0 0 1 )
12: ( 1 0 0 0 0 0 1 0 0 )
13: ( 0 1 0 0 0 0 1 0 0 )
14: ( 0 0 1 0 0 0 1 0 0 )
15: ( 0 0 0 1 0 0 1 0 0 )
16: ( 0 0 0 0 1 0 1 0 0 )
17: ( 0 0 0 0 0 1 1 0 0 )
18: ( 0 0 0 0 0 0 2 0 0 )
19: ( 0 0 0 0 0 0 1 1 0 )
20: ( 0 0 0 0 0 0 1 0 1 )
21: ( 1 0 0 0 0 0 0 1 0 )
22: ( 0 1 0 0 0 0 0 1 0 )
23: ( 0 0 1 0 0 0 0 1 0 )
24: ( 0 0 0 1 0 0 0 1 0 )
25: ( 0 0 0 0 1 0 0 1 0 )
26: ( 0 0 0 0 0 1 0 1 0 )
27: ( 0 0 0 0 0 0 1 1 0 )
28: ( 0 0 0 0 0 0 0 2 0 )
29: ( 0 0 0 0 0 0 0 1 1 )
30: ( 1 0 0 0 0 0 0 0 1 )
31: ( 0 1 0 0 0 0 0 0 1 )
32: ( 0 0 1 0 0 0 0 0 1 )
33: ( 0 0 0 1 0 0 0 0 1 )
34: ( 0 0 0 0 1 0 0 0 1 )
35: ( 0 0 0 0 0 1 0 0 1 )
36: ( 0 0 0 0 0 0 1 0 1 )
37: ( 0 0 0 0 0 0 0 1 1 )
38: ( 0 0 0 0 0 0 0 0 2 )

(Final) Minimal Elements of the control-invariant set
=====================================================

 1: ( 0 0 0 1 1 0 0 0 0 )
 2: ( 0 1 0 1 0 0 0 0 0 )
 3: ( 2 0 0 0 0 0 0 0 0 )
 4: ( 1 1 0 0 0 0 0 0 0 )
 5: ( 1 0 0 1 0 0 0 0 0 )
 6: ( 1 0 0 0 1 0 0 0 0 )
 7: ( 1 0 0 0 0 0 1 0 0 )
 8: ( 1 0 0 0 0 0 0 1 0 )
 9: ( 1 0 0 0 0 0 0 0 1 )
10: ( 0 1 0 0 0 0 1 0 0 )
11: ( 0 0 0 1 0 0 1 0 0 )
12: ( 0 0 0 0 1 0 1 0 0 )
13: ( 0 0 0 0 0 0 2 0 0 )
14: ( 0 0 0 0 0 0 1 1 0 )
15: ( 0 0 0 0 0 0 1 0 1 )
16: ( 0 1 0 0 0 0 0 1 0 )
17: ( 0 0 0 1 0 0 0 1 0 )
18: ( 0 0 0 0 1 0 0 1 0 )
19: ( 0 0 0 0 0 0 0 2 0 )
20: ( 0 0 0 0 0 0 0 1 1 )
21: ( 0 1 0 0 0 0 0 0 1 )
22: ( 0 0 0 1 0 0 0 0 1 )
23: ( 0 0 0 0 1 0 0 0 1 )
24: ( 0 0 0 0 0 0 0 0 2 )

This is An LESP
```

Figure 4. The output file generated by the software described in reference [7] for the FCPN structure shownin figure 3(b).

REFERENCES

[1] P. Valckenaers and H. Van-Brussel, "Deadlock avoidance in flexible flow shops with loops," Journalof Intelligent Manufacturing, vol. 14, pp. 137–144, 2003.

[2] L. Castillo, J. Fdez-Olivares, and A. Gonzalez, "A three-level knowledge-based system for the generationof live and safe petri nets for manufacturing systems," Journal of Intelligent Manufacturing,vol. 11, pp. 559–572, 2000.

[3] L. Ferrarini, L. Piroddi, and S. Allegri, "Modeling and logic controller specification of flexible manufacturingsystems using behavioral traces and petri net building blocks," Journal of Intelligent Manufacturing,vol. 15, pp. 351–371, 2004.

[4] N. Somnath and R. Sreenivas, "On Deciding the Existence of a Liveness Enforcing Supervisory Policyin a Class of Partially-Controlled General Free-Choice Petri Nets," IEEE Transactions on AutomationScience and Engineering, vol. 10, pp. 1157–1160, October 2013.

[5] R. Sreenivas, "On the existence of supervisory policies that enforce liveness in partially controlled free-choice petri nets," IEEE Transactions on Automatic Control, vol. 57, no. 2, pp. 435–449, February2012.

[6] R. Sreenivas, "On a decidable class of partially controlled petri nets with liveness enforcing supervisory policies," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 43, no. 5, pp. 1256–1261, August 2013.

[7] S. Chandrasekaran, N. Somnath, and R. Sreenivas, “A Software Tool for the Automatic Synthesis of Minimally Restrictive Liveness Enforcing Supervisory Policies for a class of General Petri Nets,”Journal of Intelligent Manufacturing, 2014, to appear (DOI 10.1007/s10845-014-0888-5).

[8] R. Valk and M. Jantzen, “The residue of vector sets with applications to decidability problems in Petri nets,” ActaInformatica, vol. 21, pp. 643–674, 1985.

[9] P. Ramadge and W. Wonham, “Modular feedback logic for discrete event systems,” SIAM J. Controland Optimization, vol. 25, no. 5, pp. 1202–1218, September 1987.

[10] R. Sreenivas, “On a weaker notion of controllability of a language K with respect to a language L,” IEEE Transactions on Automatic Control, vol. 438, no. 9, pp. 1446–1447, September 1993.

[11] R. Sreenivas, “On the existence of supervisory policies that enforce liveness in discrete-event dynamic systems modeled by controlled Petri nets,” IEEE Transactions on Automatic Control, vol. 42, no. 7, pp.928–945, July 1997.

[12] A. Giua, “Petri nets as discrete event models for supervisory control,” Ph.D. dissertation, ECSE Dept.,Rensselaer Polytechnic Institute, Troy, NY., 1992.

[13] J. Moody and P. Antsaklis, Supervisory Control of Discrete Event Systems using Petri Nets. MA: Kluwer Academic Publishers, 1998.

[14] M. Iordache and P. Ansaklis, Supervisory control of Concurrent Systems: A Petri net Structural Approach. MA: Kulwer Academic Publishers, 2006.

[15] F. Basile, L. Recalde, P. Chiacchio, and M. Silva, “Closed-loop Live Marked Graphs under GeneralizedMutual Exclusion Constraint Enforcement,” Discrete Event Dynamic Systems, vol. 19, no. 1, pp.1–30, 2009.

[16] J. Lopez-Grao and J. Colom, “A petri net perspective on the resource allocation problem in software engineering,” in Transactions on Petri Nets and Other Models of Concurrency, LNCS Vol. 6900,Springer, 2012, pp. 181–200.

[17] F. Tricas, “Analysis, prevention and avoidance of deadlocks in sequential resource allocation systems,” PhD Thesis, Departamento de Ingenier´ıaEl´ectrica e Inform´atica, Universidad de Zaragoza, 2003.

[18] S. Reveliotis, E. Roszkowska, and J. Choi, “Generalized algebraic deadlock avoidance policies for sequential Resource Allocation Systems,” IEEE Transactions on Automatic Control, vol. 53, no. 7,pp. 2345–2350, December 2007.

[19] A. Ghaffari, N. Rezg, and X. Xie, “Design of a live and maximally permissive Petri net controllerusing the theory of regions,” IEEE transactions on robotics and Automation, vol. 19, no. 1, pp. 137–142, January 2003.

[20] O. Marchetti and A. Munier-Kordon, “A sufficient condition for the liveness of weighted event graphs,” European Journal of Operations Research, vol. 197, pp. 532–540, 2009.

[21] L. Ferrarini, L. Piroddi, and S. Allegri, “A comparative performance analysis of deadlock avoidance control algorithms for FMS,” Journal of Intelligent Manufacturing, vol. 10, pp. 569–585, 1999.

[22] Y. Chen, Z. Li, and K. Barkaoui, “Maximally permissive liveness-enforcing supervisor with lowest implementation cost for flexible manufacturing systems,” Information Sciences, vol. 256, pp. 74–90,2014.

[23] H. Hu and Z. Li, “Synthesis of liveness enforcing supervisor for automated manufacturing systemsusing insufficiently marked siphons,” Journal of Intelligent Manufacturing, vol. 21, no. 4, pp. 555–567, 2010.

[24] H. Hu and Y. Liu, “Supervisor simplification for ams based on petri nets and inequality analysis,” IEEE Transactions on Automation Science and Engineering, vol. 11, no. 1, pp. 66–77, January 2014.

[25] S.-Y. Li, A.-M. An, Y. Wang, G. Wang, C. Hou, and Y. Cai, “Design of liveness-enforcing supervisors with simpler structures for deadlock-free operations in flexible manufacturing systems usingnecessary siphons,” Journal of Intelligent Manufacturing, vol. 24, pp. 1157–1173, 2013.

[26] R. Sreenivas, “Some observations on supervisory policies that enforce liveness in partially controlled Free Choice Petri nets,” Mathematics and Computers in Simulation, vol. 70, pp. 266–274, 2006.