Information Processing Letters 41 (1992) 313–319
North-Holland

17 April 1992

A solution to the covering problem for 1-bounded conflict-free Petri nets using Linear Programming *

Javier Esparza
Institut für Informatik, Universität Hildesheim, Samelsonplatz 1, W-3200 Hildesheim, Germany

Communicated by T. Lengauer
Received 21 August 1991
Revised 30 December 1991

Abstract

Esparza, J., A solution to the covering problem for 1-bounded conflict-free Petri nets using Linear Programming, Information Processing Letters 41 (1992) 313–319.

Given a marking μ of a Petri net, the covering problem consists of determining if there exists a reachable marking $\mu' \geqslant \mu$. We show that the covering problem for 1-bounded conflict-free Petri nets is polynomially reducible to a Linear Programming problem. This proves that the covering problem is in PTIME for this class of Petri nets, which generalises a result of Yen.

Keywords: Concurrency, conflict-free Petri nets, Linear Programming

1. Introduction

Conflict-free Petri nets have been extensively studied from a computational point of view. Some interesting problems have been proved to be tractable. In particular, Howell and Rosier showed in [1] that the reachability problem for bounded conflict-free Petri nets is in PTIME (whereas, if the restriction to bounded nets is removed, the problem is NP-complete). In [6], Yen gave a polynomial time algorithm to determine if two transitions in a 1-bounded conflict-free Petri net can become simultaneously enabled (called in [6] the concurrency problem), a problem with applications to the verification of self-timed circuits [5].

In this paper, we show that the *covering problem* for 1-bounded conflict-free Petri nets is polynomially reducible to a Linear Programming problem. Since Linear Programming is known to be in PTIME, it follows that this problem is in PTIME as well. Both the concurrency problem and the reachability problem in the 1-bounded case are particular instances of the covering problem.

Given a marking μ of a Petri net, the covering problem consists of determining if there exists a reachable marking $\mu' \geqslant \mu$. The concurrency problem for any set of transitions U (i.e. the problem of determining if all the transitions in U can become simultaneously enabled) can be reduced to the covering problem by taking $\mu(p) = 1$ iff p is an input place of some transition in U. We are

Correspondence to: Dr. J. Esparza, Institut für Informatik, Universität Hildesheim, Samelsonplatz 1, W-3200 Hildesheim, Germany. Email: esparza@infhil.uucp.

* This work was partially supported by the Esprit Basic Research Action 3148 DEMON.

also able to compute the smallest reachable marking $\mu' \geqslant \mu$; this solves the reachability problem because $\mu' = \mu$ if and only if μ is reachable.

The reduction makes use of a result of [6], and of a characterisation of the Parikh vectors corresponding to the computations of 1-bounded conflict-free Petri nets. This characterisation can be used as a basis for the solution of other problems by means of Linear Programming.

2. Definitions

As in [6], a Petri net is a 4-tuple (P, T, ϕ, μ_0), where P is a finite set of *places*, T is a finite set of *transitions*, ϕ is a *flow function*

$$\phi:(P\times T)\cup(T\times P)\to\{0, 1\}$$

and μ_0 is the *initial marking*, $\mu_0 : P \to \mathbb{N}$. *Markings*, *firing of transitions*, *firing sequences*—also called here *computations*—and *reachable markings* are defined as usual.

Given a place p, we denote

$$p^\bullet = \{t \in T \mid \phi(p, t) = 1\}$$

and

$$^\bullet p = \{t \in T \mid \phi(t, p) = 1\}.$$

Let $\mathscr{P} = (P, T, \phi, \mu_0)$ be a Petri net with $P = \{p_1, \ldots, p_n\}$ and $T = \{t_1, \ldots, t_m\}$.

$\mathscr{P}$ is a *marked graph* iff for every $p \in P$, $|^\bullet p| \leqslant 1$ and $|p^\bullet| \leqslant 1$. $\mathscr{P}$ is *conflict-free* iff for every place $p \in P$ either

(1) $|p^\bullet| \leqslant 1$, or

(2) $\forall t \in p^\bullet$: $t \in {}^\bullet p$.

Given $p \in P$, we denote $T_p^+ = {}^\bullet p \setminus p^\bullet$ and $T_p^- = p^\bullet \setminus {}^\bullet p$. T_p^+ (T_p^-) is the set of transitions whose occurrence increases (decreases) the marking of p. Notice that if a net is conflict-free then for every place p: $|T_p^-| \leqslant 1$.

The $n \times m$ matrix C given by $C(i, j) = \phi(t_i, p_j) - \phi(p_j, t_i)$ is the *incidence matrix* of $\mathscr{P}$.

A nonnegative integer vector of dimension $|T|$ is called a *Parikh vector*. Given a firing sequence σ, $\#_\sigma : T \to \mathbb{N}$ is the mapping defined by $\#_\sigma(t) =$ number of occurrences of t in σ. The representation of this mapping in vector form, according to the total order $t_1 < t_2 < \quad < t_m$, is the Parikh vector associated to σ. A Parikh vector X is *executable* iff $X = \#_\sigma$ for some computation σ. We denote by $S(X)$ the set of transitions with positive components in X, i.e. $S(X) = \{t \in T \mid X(t) > 0\}$.

3. Characterisation of executable Parikh vectors

We assume within this paper, unless otherwise stated, that $\mathscr{P} = (P, T, \phi, \mu_0)$ is a 1-bounded conflict-free Petri net in which every transition occurs in some computation. Transitions occurring in no computation can be removed in polynomial time without changing the behaviour of the net, as the next lemma shows. Therefore, we can transform an arbitrary 1-bounded conflict-free Petri net into an equivalent one satisfying the assumption in polynomial time.

Lemma 3.1 [2]. *Given a conflict-free Petri net $\mathscr{P} = (P, T, \phi, \mu_0)$, we can construct in polynomial time a computation τ enabled in μ_0 in which no transition in τ is used more than once, such that if some transition t is not used in τ, then there is no computation (emanating from μ_0) in which t is used.* □

In the sequel, the symbol τ is reserved for this particular computation. Notice that, in the nets satisfying our assumption, $\#_\tau(t) = 1$ for every transition t.

There exists a simple characterisation of the set of executable Parikh vectors for the class of marked graphs, when every transition is known to occur in some computation [4]. Namely, X is executable iff

$$\mu_0 + C \cdot X \geqslant 0, \tag{1}$$

where C is the incidence matrix of the marked graph. This is no longer true for 1-bounded conflict-free systems. The Petri net of Fig. 1 is an example: $X = (0, 1, 1)$ satisfies (1), but it corresponds to no computation. This is due to the fact that (1) does not impose any constraint on the

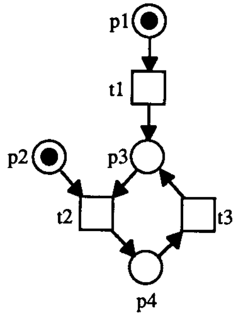

Fig. 1. A 1-bounded conflict-free system.

order in which transitions have to occur, and in this case t_1 must occur before t_2 or t_3.

We show in this section that by adding some information about the causal relationships between transitions these solutions corresponding to no computation can be eliminated.

We recall, first, a fundamental lemma.

Given two vectors $X=(x_1,\dots,x_n)$ and $Y=(y_1,\dots,y_n)$ of integers, define

$$\max(X, Y)=(\max(x_1, y_1),\dots,\max(x_n, y_n)).$$

Lemma 3.2 [3]. *Let σ_1, σ_2 be computations of $\mathscr{P}$. There is a computation σ such that*

$$\#_\sigma=\max(\#_{\sigma_1}, \#_{\sigma_2}).$$

Moreover, if $\mu_0 \xrightarrow{\sigma_1} \mu_1$, $\mu_0 \xrightarrow{\sigma_2} \mu_2$ and $\mu_0 \xrightarrow{\sigma} \mu$, then μ is reachable from μ_1 and from μ_2. □

In fact, this lemma is proved in [3] for all persistent Petri nets, of which conflict-free nets are a subclass.

Definition 3.3. Let $\mathscr{P}=(P, T, \phi, \mu_0)$ be a Petri net. The *causal relation* $\mathscr{C}\subseteq T\times T$ is defined by:

$$(t, u)\in\mathscr{C} \Leftrightarrow \forall \text{ computations } \sigma u\colon \#_\sigma(t)>0.$$

In words, in every computation containing u, t precedes u.

It is easy to see that, as would be expected, the causal relation is a partial order.

Lemma 3.4. *Let p be a place.*

(1) *If $t\in T_p^+$ and $u\in({}^\bullet p\cap p^\bullet)$, then $(t, u)\in\mathscr{C}$.*

(2) *If $t,u\in T_p^+$ and there exists a computation $\sigma_1 t$ with $\#_{\sigma_1}(u)=0$, then $(t, u)\in\mathscr{C}$.*

Proof. (1) Since $u\in({}^\bullet p\cap p^\bullet)$ and the net is conflict-free, $T_p^-=\emptyset$. Hence, the marking of p cannot decrease along a computation. Then $T_p^+=\{t\}$, because otherwise the computation τ contains at least two occurrences of transitions in T_p^+, against the 1-boundedness of $\mathscr{P}$. Moreover, since τ contains t, $\mu_0(p)=0$.

Let σu be a computation. Since u is enabled after the occurrence of σ, and $\mu_0(p)=0$, t must occur in σ. So $(t, u)\in\mathscr{C}$.

(2) Let $\sigma_2 u$ be a computation and $\mu_0 \xrightarrow{\sigma_2} \mu$. By Lemma 3.2, there exists a computation σ such that

$$\#_\sigma=\max\{\#_{\sigma_1 t}, \#_{\sigma_2 u}\}.$$

Let $\mu_0 \xrightarrow{\sigma_1 t} \mu_1$, $\mu_0 \xrightarrow{\sigma_2 u} \mu_2$, $\mu_0 \xrightarrow{\sigma} \mu$. We show that $\mu(p)>1$, against the 1-boundedness of $\mathscr{P}$. Consider two cases:

Case 1: $T_p^-=\emptyset$. Then $\mu(p)>1$ because σ contains at least one occurrence of t and one of u.

Case 2: $T_p^-\neq\emptyset$. Then, by the conflict-free property, $T_p^-=\{v\}$. We have:

$$1\geqslant\mu_1(p)\geqslant\mu_0(p)+\#_{\sigma_1 t}(t)-\#_{\sigma_1 t}(v), \tag{2}$$

$$1\geqslant\mu_2(p)\geqslant\mu_0(p)+\#_{\sigma_2 u}(u)-\#_{\sigma_2 u}(v), \tag{3}$$

$$\mu(p)\geqslant\mu_0(p)+\#_{\sigma_1 t}(t)+\#_{\sigma_2 u}(u) - \max\{\#_{\sigma_1 t}(v), \#_{\sigma_2 u}(v)\}. \tag{4}$$

If $\max\{\#_{\sigma_1 t}(v), \#_{\sigma_2 u}(v)\}=\#_{\sigma_1 t}(v)$, we get from (2) and (4)

$$\mu(p)\geqslant\#_{\sigma_2 u}(u)+1. \tag{5}$$

If $\max\{\#_{\sigma_1 t}(v), \#_{\sigma_2 u}(v)\}=\#_{\sigma_2 t}(v)$, we get from (3) and (4)

$$\mu(p)\geqslant\#_{\sigma_1 t}(t)+1. \tag{6}$$

Since $\#_{\sigma_2 u}(u)\geqslant 1$ and $\#_{\sigma_1 t}(t)\geqslant 1$, we have in both cases $\mu(p)>1$. □

We can now formulate and prove our characterisation of the executable Parikh vectors. Consider a Parikh vector X and $(t, u) \in \mathscr{C}$. If X is executable and $X(u) > 0$, then, since every computation containing u contains also t, we have $X(t) > 0$. This gives a necessary condition for a Parikh vector to be executable, which turns out to be sufficient as well. Moreover, it is not necessary to verify this condition for every pair of transitions in $\mathscr{C}$; it suffices to do it for the pairs (t, u) where both t and u are input transitions of the same place.

Theorem 3.5. *A Parikh vector X is executable iff*

(1) $\mu_0 + C \cdot X \geqslant 0$ *and*

(2) $\forall p \in P, \forall (t, u) \in \mathscr{C} \cap (T_p^+ \times {}^\bullet p)$: $(X(t) = 0 \Rightarrow X(u) = 0)$.

Proof. ($\Rightarrow$) (1) is well known. (2) follows from the definition of the causal relation.

($\Leftarrow$) We show that if X is not executable and (1) holds, then (2) does not hold.

Let $X' \leqslant X$ be a maximal executable Parikh vector (i.e. no vector X'' with $X' < X'' \leqslant X$ is executable). Let $\mu = \mu_0 + C \cdot X$, $\mu' = \mu_0 + C \cdot X'$ and $Y = X - X' \neq 0$. We have:

(a) $\mu' + C \cdot Y \geqslant 0$. Follows from $\mu' + C \cdot Y = \mu_0 + C \cdot X \geqslant 0$.

(b) No transition in $S(Y)$ is enabled at μ'. Follows from the maximality of X'.

By (b), every transition in $S(Y)$ has at least one unmarked input place at μ'. Denote by Q the set of these unmarked places. Then:

(c) Every place in Q has at least one input transition in $S(Y)$. By the definition of Q, every place $p \in Q$ has at least one output transition in $S(Y)$. Assume p has no input transition in $S(Y)$. Then, since $\mu'(p) = 0$, we have $(\mu' + C \cdot Y) < 0$, contradicting (a).

We construct a preorder $\trianglelefteq$ on $Q \cup S(Y)$ in the following way:

$x \trianglelefteq x'$ iff there exists a path $(x, \ldots, x')$ in $\mathscr{P}$ containing only elements of $Q \cup S(Y)$.

Since $Q \cup S(Y)$ is finite, there exists at least one minimal element x_0 of $\trianglelefteq$ (i.e. $x_0 \trianglelefteq x$ for every $x \in Q \cup S(Y)$). Let X_0 be the equivalence class of x_0 induced by $\trianglelefteq \cap \trianglerighteq$, and $N_0 = (P_0, T_0, \phi)$ the subnet given by:

- $P_0 = P \cap X_0$, $T_0 = T \cap X_0$,
- $\phi_0 = \phi \cap (P_0 \times T_0 \cup T_0 \times P_0)$.

Notice that, by definition of $\trianglelefteq$, N_0 is strongly connected.

We have:

(d) $P_0 \neq \emptyset$ and $T_0 \neq \emptyset$. Follows easily from (b), (c) and the minimality of x_0.

(e) $P_0^\bullet \subset {}^\bullet P_0$ (P_0 is a trap). In conflict-free nets, this is true of the set of places of any strongly connected subnet. Since N_0 is strongly connected, (e) holds.

(f) $\Sigma_{p \in P_0} \mu'(p) = 0$ and $\Sigma_{p \in P_0} \mu_0(p) = 0$. $\Sigma_{p \in P_0} \mu'(p) = 0$ follows from $P_0 \subseteq Q$ and the definition of Q as set of unmarked places at μ'.

By (e), for every μ_1, μ_2:

$$\left(\sum_{p \in P_0} \mu_2(p) = 0 \wedge \mu_1 \xrightarrow{\sigma} \mu_2 \right) \Rightarrow \sum_{p \in P_0} \mu_1(p) = 0.$$

(if a trap becomes marked, it remains marked). Taking $\mu_1 = \mu_0$ and $\mu_2 = \mu'$, the second part of (f) follows.

(g) $({}^\bullet P_0 \setminus P_0^\bullet) \cap S(Y) = \emptyset$. Let $t \in {}^\bullet P_0 \setminus P_0^\bullet$. Then there exists $p \in t^\bullet \cap P_0$. Assume $t \in S(Y)$ as well. We derive a contradiction: Since $t \in S(Y)$, we have $t \trianglelefteq p$. Then $t \in X_0$ by the minimality of p w.r.t. $\trianglelefteq$. By (b), there exists a place $p' \in {}^\bullet t \cap Q$. By definition of $\trianglelefteq$, $p' \trianglelefteq t$. This implies $p' \in P_0$, against our hypothesis $t \in {}^\bullet P_0 \setminus P_0^\bullet$.

Consider the computation τ.

By (f), the places of P_0 are initially unmarked. Hence, no transition of $P_0^\bullet$ can occur in τ before some transition ${}^\bullet P_0 \setminus P_0^\bullet$ has occurred. So a transition $t \in {}^\bullet P_0 \setminus P_0^\bullet$ appears in τ before any transition of $P_0^\bullet$.

By (g), $t \notin S(Y)$.

Since N_0 is strongly connected, there exists $u \in T_0$ such that $t^\bullet \cap u^\bullet \neq \emptyset$. Since t occurs before u in τ, we have $(t, u) \in \mathscr{C}$ by Lemma 3.4.

We show that $X(u) > 0$ and $X(t) = 0$, which implies that (2) does not hold.

- $X(u) > 0$: Since $u \in T_0$ and $T_0 \subseteq S(Y)$ by definition of T_0, we have $u \in S(Y)$. So $Y(u) > 0$. Since $Y = X - X'$, $X(u) > 0$.

– $X(t)=0$. Assume $X(t)>0$. Since $t \notin S(Y)$, we have $Y(t)=0$. Therefore, since $Y=X-X'$, $X'>0$ follows. Since $t \in {}^{\bullet}P_0$, some marking μ_1 is reached along the execution of X' such that $\Sigma_{p \in P_0}\mu_1(p)>0$. By (e), this implies $\Sigma_{p \in P_0}\mu_0(p)>0$. This contradicts (f). So $X(t)=0$. □

4. The covering problem is reducible to a Linear Programming problem

We would like to add additional inequations to the system (1) in Theorem 3.5 which guarantee that the solutions of the augmented system are exactly the executable Parikh vectors. This requires to express condition (2) as an inequation. However, it is easy to see that this inequation cannot be linear, which forbids the use of Linear Programming. Fortunately, things change if we are interested only in the executable Parikh vectors $X \leqslant (k,\dots,k)$ for some number k. Then, condition (2) can be expressed as:

$$k \cdot X(t) \geqslant X(u) \tag{7}$$

which is linear. We show that for every reachable marking there exists an executable Parikh vector X leading to it such that $X \leqslant (|T|,\dots,|T|)$. This result derives from Lemma 3.2 of [6], which will allow us to make use of (7) to reduce the covering problem to a Linear Programming problem.

We have to reformulate this lemma to adapt them to our context.

Lemma 4.1 [6]. *If $\mu_0 \xrightarrow{\sigma} \mu$ is one of the shortest computations leading to μ, then σ can be rearranged into $\sigma_1\sigma_2\dots\sigma_k$ such that*

(1) $\mu_0 \xrightarrow{\sigma_1\sigma_2\dots\sigma_k} \mu$,
(2) $\forall 1 \leqslant i \leqslant k, \forall t \in T$: $\#_{\sigma_i}(t) \leqslant 1$,
(3) $\forall 1 \leqslant i \leqslant k-1$, $S(\sigma_{i+1}) \subset S(\sigma_i)$, *and*
(4) $k \leqslant |T|$ *and* $|\sigma| \leqslant |T|^2$. □

This result is proved in [6] for computations enabling two particular transitions u and v, but this assumption is not used in the proof. We have now the following:

Corollary 4.2. *If $\mu_0 \xrightarrow{\sigma} \mu$ is one of the shortest computations leading to μ, then $\#_\sigma \leqslant (|T|,\dots,|T|)$.*

Proof. Rearrange σ into $\sigma_1\sigma_2\dots\sigma_k$ as in Lemma 4.1. Then:

$$\#_\sigma = \sum_{i=1}^{k} \#_{\sigma_i} \tag{8}$$

$$\leqslant \sum_{i=1}^{k} (1,\dots,1) \qquad \text{(by Lemma 4.1(2))} \tag{9}$$

$$\leqslant (|T|,\dots,|T|) \qquad \text{(by Lemma 4.1(4))}. \tag{10}$$

□

Theorem 4.3. *Let μ be a marking of $\mathscr{P}$, and $I(\mu)$ be the following system of linear inequalities:*

$$\mu_0 + C \cdot X \geqslant \mu,$$
$$X \leqslant (|T|,\dots,|T|),$$
$$\forall p \in P, \forall (t,u) \in \mathscr{C} \cap (T_p^+ \times {}^{\bullet}p):$$
$$|T| \cdot X(t) \geqslant X(u).$$

There exists a reachable marking $\mu' \geqslant \mu$ iff $I(\mu)$ has an integer solution.

Proof. Use Theorem 3.5, Corollary 4.2 and the observation leading to equation (7). □

Deciding if a system of linear inequalities has an *integer* solution is known to be NP-complete. Hence, even if we can write $I(\mu)$ in polynomial time (which will be shown later), Theorem 4.3 does not directly provide a polynomial algorithm. Fortunately, we can do better.

Given a real number x, $\lceil x \rceil$ denotes the smallest integer greater than or equal to x and $\lfloor x \rfloor$ the largest integer less than or equal to x.

Theorem 4.4. *Let $LP(\mu)$ be the following Linear Programming problem:*

$$\textit{maximise} \quad \sum_{t \in T} X(t)$$
$$\textit{subject to} \quad I(\mu)$$

There exists a reachable marking $\mu' \geqslant \mu$ iff $LP(\mu)$ has an optimal solution X_{op} (i.e. a solution such that for any other solution X, $\Sigma_{t \in T} X(t) \leqslant \Sigma_{t \in T} X_{op}(t)$). Moreover, X_{op} is an executable Parikh vector leading to μ'.

Proof. Since the solutions of $LP(\mu)$ are bounded by $(|T|, \ldots, |T|)$, $LP(\mu)$ has an optimal solution iff it has a solution at all. By Theorem 4.3, all we have to do is prove that the optimal solutions of $LP(\mu)$ are integer.

Given a vector $X = (x_1, \ldots, x_n)$, define $\lceil X \rceil = (\lceil x_1 \rceil, \ldots, \lceil x_n \rceil)$.

Assume X_{op} is a noninteger optimal solution. We show that $\lceil X_{op} \rceil$ is solution of $LP(\mu)$, against the optimality of X_{op}:

(1) $(\mu_0 + C \cdot \lceil X_{op} \rceil) \geqslant \mu$. Let p be a place. By the conflict-free property, $|T_p^-| \leqslant 1$. Taking into account that $(\mu_0 + C \cdot X_{op})(p) \geqslant \mu(p)$ and $\mu(p)$ is integer, some elementary arithmetic leads to:

$$(\mu_0 + C \cdot \lceil X_{op} \rceil)(p) \geqslant \lfloor (\mu_0 + C \cdot X_{op})(p) \rfloor \geqslant \mu(p).$$

(2) $\lceil X_{op} \rceil \leqslant (|T|, \ldots, |T|)$. Follows from $X_{op} \leqslant (|T|, \ldots, |T|)$.

(3) $\forall (t, u) \in \mathscr{C} \cap (T_p^+ \times {}^{\bullet}p)$: $|T| \cdot \lceil X(t) \rceil \geqslant \lceil X(u) \rceil$. Follows from the corresponding property for X_{op}. $\square$

Solving $LP(\mu)$ takes polynomial time in the size of the system of inequalities. We prove now that writing $LP(\mu)$ takes polynomial time in the size of the net, which implies that the covering problem is polynomial in the size of the net.

By inspection of $LP(\mu)$, it is easy to see that the only problem consists of finding for each place p the pairs $(t, u) \in \mathscr{C} \cap (T_p^+ \times T_p^+)$.

Proposition 4.5. *Let p be a place. $\mathscr{C} \cap (T_p^+ \times T_p^+)$ can be computed in polynomial time.*

Proof. If $|T_p^+| = 1$, we are done. Assume $|T_p^+| > 1$ and let $(t, u) \in (T_p^+ \times T_p^+)$. By our initial assumption, every transition occurs in some computation. Therefore, the sequence τ contains all transitions. In particular, it contains t and u. By Lemma 3.4(2), if t appears before u in τ, then $(t, u) \in \mathscr{C}$; otherwise, $(u, t) \in \mathscr{C}$.

Therefore, once the sequence τ is known, we can easily compute $\mathscr{C} \cap (T_p^+ \times T_p^+)$. Since τ can be computed in polynomial time by Lemma 3.1, $\mathscr{C} \cap (T_p^+ \times T_p^+)$ can be computed in polynomial time. $\square$

Howell and Rosier gave in [1] a polynomial time algorithm to solve the reachability problem for bounded conflict-free Petri nets. We show now that the problem $LP(\mu)$ can also be used to decide the reachability of μ, albeit only in the 1-bounded case. The advantage of the derived algorithm is that it is easy to implement once a Linear Programming tool is available, and joins the already large set of Linear Programming algorithms for the verification of properties in Petri nets.

Theorem 4.6. *μ is reachable iff $LP(\mu)$ has optimal solutions and every optimal solution X_{op} satisfies $\mu = \mu_0 + C \cdot X_{op}$. Moreover, X_{op} is an executable Parikh vector leading to μ.*

Proof. ($\Rightarrow$) Since μ is reachable, there exists an executable Parikh vector X_μ leading to μ. By Theorem 4.4, X_μ is solution of $LP(\mu)$. Since the basic solutions of $LP(\mu)$ are bounded by $(|T|, \ldots, |T|)$, the problem has optimal solutions.

Let X_{op} be an optimal solution, and let $\mu_{op} = \mu_0 + C \cdot X_{op}$. By the definition of $LP(\mu)$, we have $\mu_{op} \geqslant \mu$.

We show that $\mu = \mu_{op}$, which proves the result.

By Theorem 4.4, X_{op} is executable. Then, by Lemma 3.2, $Y = \max\{X_{op}, X_\mu\}$ is also executable.

Claim. $Y = X_{op}$.

Proof of the claim. Since $Y \geqslant X_{op}$ and X_{op} is optimal, it suffices to show that Y is a basic solution of $LP(\mu)$. It is easy to see from its definition that Y satisfies the second and third constraints of $LP(\mu)$. It remains to prove $\mu_0 + C \cdot Y \geqslant \mu$.

Let $p \in P$. Consider two cases:

(1) $\mu(p)=0$. Since Y is executable, we have $\mu_0+C\cdot Y\geqslant 0$. So $(\mu_0+C\cdot Y)(p)\geqslant\mu(p)$.

(2) $\mu(p)=1$. Then, since $\mu=\mu_{op}$, we also have $\mu_{op}(p)=1$. Moreover:

$$\begin{aligned}(\mu_0+C\cdot Y)(p)&\\ &=\mu(p)+\sum_{t\in T_p^+}Y(t)-\sum_{t\in T_p^-}Y(t)\\ &=\mu_0(p)+\sum_{t\in T_p^+}\max\{X_{op}(t),X_\mu(t)\}\\ &\quad-\sum_{t\in T_p^-}\max\{X_{op}(t),X_\mu(t)\}.\end{aligned}$$

If $T_p^-=\emptyset$, then $(\mu_0+C\cdot Y)(p)\geqslant\mu_{op}(p)$ and $(\mu_0+C\cdot Y)(p)\geqslant\mu(p)$, and we are done. If $T_p^-=\{u\}$, then we have:

$$\begin{aligned}&(\mu_0+C\cdot Y)(p)\geqslant\mu_{op}(p)\\ &\quad\text{if }\max\{X_{op}(u),X_\mu(u)\}=X_{op}(u),\\ &(\mu_0+C\cdot Y)(p)\geqslant\mu(p)\\ &\quad\text{if }\max\{X_{op}(u),X_\mu(u)\}=X_\mu(u).\end{aligned}$$

In all cases, $(\mu_0+C\cdot Y)(p)=1$.

By Lemma 3.2, μ_{op} is reachable from μ and, since X_{op} is solution of $LP(\mu)$, we have $\mu_{op}\geqslant\mu$. If $\mu_{op}\neq\mu$, then the Petri net is not 1-bounded (even not bounded), against our hypothesis. So $\mu_{op}=\mu$.

($\Leftarrow$) Use Theorem 4.4. $\square$

5. Conclusions

We have shown that the covering problem for 1-bounded conflict-free Petri nets is polynomially reducible to a Linear Programming problem. As a consequence, it is in PTIME. The concurrency of an arbitrary set of transitions (a generalisation of the problem considered in [6]) and the reachability of a marking in the 1-bounded case can be decided solving particular instances of this problem. In order to obtain the reducibility result, we have proved a characterisation of the set of executable Parikh vectors, generalising known results for marked graphs (see [4] for a survey). A lemma of [6] was also very useful.

Acknowledgment

I am very indebted to two anonymous referees for pointing out several mistakes in an earlier version of this paper and for very helpful suggestions.

References

[1] R. Howell and L. Rosier, On questions of fairness and temporal logic for conflict-free Petri nets, in: G. Rozenberg, ed., *Advances in Petri Nets 1988*, Lecture Notes in Computer Science **340** (Springer, Berlin, 1988) 200–226.

[2] R. Howell, L. Rosier and H. Yen, An $O(n^{1.5})$ algorithm to decide boundedness for conflict-free vector replacement systems, *Inform. Process. Lett.* **25** (1987) 27–33.

[3] L. Landweber and E. Robertson, Properties of conflict-free and persistent Petri nets. *J. ACM* **25** (3) (1978) 352–364.

[4] T. Murata, Petri nets: Properties, analysis and applications, *Proc. IEEE* **77** (4) (1989) 541–580.

[5] M. Tiusanen, Some unsolved problems in modeling self-timed circuits using Petri nets, *EATCS Bulletin* **36** (1988) 152–160.

[6] H. Yen, A polynomial time algorithm to decide pairwise concurrency of transitions for 1-bounded conflict-free Petri nets, *Inform. Process. Lett.* **38** (1991) 71–76.