Extremal throughputs in free-choice nets

Anne Bouillard, Bruno Gaujal, Jean Mairesse

Extremal throughputs in free-choice nets

Anne Bouillard[1], Bruno Gaujal[1] and Jean Mairesse[2]

[1] Lip (Umr Cnrs, Ens Lyon, Inria, Université Claude Bernard Lyon 1),
École Normale Supérieure de Lyon,
46, allée d'Italie - 69364 Lyon Cedex 07 - France
{abouilla,bgaujal}@ens-lyon.fr
[2] Liafa, Cnrs - Université Paris 7
Case 7014 - 2, place Jussieu - 75251 Paris Cedex 5 - France
mairesse@liafa.jussieu.fr

Abstract. We give a method to compute the throughput in a timed live and bounded free-choice Petri net under a total allocation. We also characterize the conflict-solving policies that achieve the smallest throughput in the special case of a 1-bounded net. It does not correspond to a total allocation, but "almost".

1 Introduction

Consider a live and bounded free-choice Petri net (LBFC). Such Petri nets realize a good compromise between modelling power and mathematical tractability, see [5] for several striking examples of the latter. Assume that the Petri net is timed with a timing specified by a constant real-valued firing time for each transition. To remove the undeterminism of the behavior of the Petri net, a policy for the resolution of all the conflicts needs to be decided. Once it is chosen, all the enabled transitions start to fire as soon as possible, and the time that elapses between the beginning and the completion of the firing of a transition is equal to the firing time. Therefore, the timed evolution of the Petri net is completely determined.

Our goal is to study the global *activity* or *throughput* or *firing rate of the transitions* of the Petri net in a sense to be made precise later on. Of course, the activity depends on the chosen policy for resolving conflicts.

In a free-choice Petri net, one may view a conflict-solving policy as a set of local functions associated with conflict places, and assigning tokens to output transitions. The simplest class of policies consists of the so-called *0-1 policies*: for a conflict place p, allocate all the tokens to a fixed transition. Zero-one policies are called *total allocations* in [5]. The next simplest class of policies is, arguably, the *periodic* ones: for a conflict place p, allocate the tokens to the output transitions according to some fixed periodic pattern. Obviously, 0-1 policies are extremal periodic policies.

In this paper, we address the following natural questions:

– A. Given a periodic policy, is the activity explicitly computable?
– B. Consider the set of all possible, arbitrarily complex, policies for resolving conflicts. Is the infimum, resp. supremum, of the activity over this set attained by a 0-1 or a periodic policy? Can we explicitly determine the policies realizing the infimum, resp. supremum?

For both questions, we are also concerned by the algorithmic complexity of the computations.

Consider first Question A. It is known that the activity is explicitly computable when the timings are rational-valued [2]. The solution relies on the construction of a very large graph G in which a state incorporates three different types of information: the current marking; the remaining time before completion for the currently firing transitions; and the current position of the cursor within the periods for the periodic policy. The timed behavior is ultimately periodic and the period corresponds to a circuit in the graph G. The activity is computed along this circuit.

The method has two major drawbacks. First, it is not efficient from an algorithmic point of view. Indeed, the graph G is in general much larger than the reachability (marking) graph whose size may already be exponential in the one of the Petri net. Second, it does not provide much insight on the structure of the timed behavior. Here, we show that both restrictions can be overcome in the special case of a 0-1 policy: the live part of the Petri net becomes a disjoint union of event graphs. Consequently, the activity can be computed in polynomial (cubic) time in the size of the Petri net, using classical results on the throughput of timed event graphs [1, 3, 4]. Furthermore, the previous restriction on having rational timings is not necessary anymore for 0-1 policies.

Consider now Question B and assume that the timings are rational-valued for simplicity. Using a simplified version of the above graph G, in which the periodic policy is not coded anymore, one can easily prove that the supremum and the infimum of the activity are obtained for periodic policies. The drawbacks are the same as before: the time complexity, and the lack of structural insight. Concerning the latter, the method does not allow to answer the question: is the supremum or infimum attained by 0-1 policies?

Presumably against the intuition, we exhibit an example of a live and 1-bounded Petri net with firing times all equal to one, and for which the infimum is attained only by non-0-1 periodic policies. More generally, we show that for 1-bounded Petri nets with general real-valued timings, the infimum is attained by a periodic policy which may not be 0-1 but is *"almost 0-1"*: it is a 0-1 policy except on an elementary circuit of the Petri net. The same result fails to hold for a k-bounded Petri net, $k \geq 2$, and the general structure of infimum policies is not understood in this case. An example is given of a 2-bounded net with timings all equal to one, for which the supremum is attained by periodic policies which are neither 0-1, nor almost 0-1.

To be complete, let us mention that the general structure of supremum policies is not well understood, even for 1-bounded nets. It is easy to build examples of 1-bounded LBFC with rational timings for which the *supremum* is attained only by non-0-1 Sturmian-like periodic policies, as well as examples with irrational timings for which the supremum is attained only by Sturmian-like non-periodic policies [8].

In order to obtain the above results, we use three different types of building blocks:

- The theory of timed event graphs;
- A structural result stating that the live part of a LBFC with a total allocation is a disjoint union of T-components; and that, given a T-component, there exists a total allocation making this T-component the only live part of the LBFC;
- The notion of Token-Transition invariants. It is a refinement of the classical notion of T-invariants with a dynamical flavor to it, since it "follows" the evolution of a token.

The first point is very classical [1, 3, 4], while the other two may be original and of some interest by their own.

The paper is organized as follows. The known results on Question A appear in Section 3. In Section 4, we study the 0-1 policies in detail. Section 5 introduces the TT-invariants. Section 6 is devoted to Question B. In particular, we characterize the almost 0-1 policies which provide the infimum throughput for a 1-bounded net in Subsection 6.2.

2 Notations and preliminaries

A *net* is a bipartite directed graph $(\mathcal{P}, \mathcal{T}, \mathcal{F})$ with $\mathcal{P} \cup \mathcal{T}$ as the set of nodes ($\mathcal{P} \cap \mathcal{T} = \emptyset$) and $\mathcal{F} \subseteq (\mathcal{P} \times \mathcal{T}) \cup (\mathcal{T} \times \mathcal{P})$ as the set of arcs. A *Petri net* is a quadruple $(\mathcal{P}, \mathcal{T}, \mathcal{F}, M)$, where $(\mathcal{P}, \mathcal{T}, \mathcal{F})$ is a net and M is a map from $\mathcal{P}$ to $\mathbb{N}$. The elements of $\mathcal{P}$ are called *places* and are represented by circles and those of $\mathcal{T}$ are called *transitions* and represented by rectangles. The function M is called the *(initial) marking* of the net and is represented by tokens in places. Let $x \in \mathcal{P} \cup \mathcal{T}$ be a node. We denote by ${}^\bullet x$ the set of its predecessors and by $x^\bullet$ the set of its successors. We also have ${}^\bullet X = \cup_{x \in X} {}^\bullet x$ and $X^\bullet = \cup_{x \in X} x^\bullet$. A transition is *conflicting* if one of its input place has at least two successors. Otherwise, the transition is *non-conflicting*.

The marking evolves according to the *firing rule*. A transition t is *enabled* if $\forall p \in {}^\bullet t,\ M(p) \geq 1$. An enabled transition can *fire*, and then the marking becomes M' with $M'(p) = M(p) - \chi_{{}^\bullet t}(p) + \chi_{t^\bullet}(p)$, where χ_S is the characteristic function of the set S.

If the marking M' is obtained from M by firing the transition t, we write $M \xrightarrow{t} M'$. If M' is obtained by successively firing $\sigma = t_1 t_2 \cdots t_n \in \mathcal{T}^*$ (this sequence of transitions is called an *admissible firing sequence*), we write $M \xrightarrow{\sigma} M'$. Finally, if M' can be reached from M by an admissible firing sequence, we write $M \rightarrow M'$. The set of the *reachable markings* of M is $\mathcal{R}(M) = \{M' \mid M \rightarrow M'\}$.

A Petri net is *live* if for every transition t and every reachable marking M_1 there exists a marking M_2, reachable from M_1, that enables t. A Petri net is deadlock-free if there exists no reachable marking where no transition is enabled. A Petri net is *k-bounded* if for every reachable marking there is less than k tokens in each place. A net $\mathcal{N}$ is *structurally live* if there exists a marking M such that the Petri net $(\mathcal{N}, M)$ is live. A net $\mathcal{N}$ is *well-formed* if there is a marking M that makes the Petri net $(\mathcal{N}, M)$ live and bounded

An *event graph* is a (Petri) net where: $\forall p \in \mathcal{P},\ |{}^\bullet p| = |p^\bullet| = 1$. A *state machine* is a (Petri) net where: $\forall t \in \mathcal{T},\ |{}^\bullet t| = |t^\bullet| = 1$. A *free-choice (Petri) net* is a (Petri) net where: $\forall (p,t) \in \mathcal{P} \times \mathcal{T},\ (p,t) \in \mathcal{F} \Rightarrow (p^\bullet = \{t\})$ or $({}^\bullet t = \{p\})$. We use the acronym *LBFC* for a live and bounded free-choice Petri net. A *conflict-free (Petri) net* is a (Petri) net where: $\forall p \in \mathcal{P},\ |p^\bullet| = 1$.

The *incidence matrix* of a Petri net is $N \in \mathbb{Z}^{\mathcal{P} \times \mathcal{T}}$ with $N_{p,t} = \chi_{t^\bullet}(p) - \chi_{{}^\bullet t}(p)$. Let $\sigma \in \mathcal{T}^*$ be an admissible firing sequence. The *commutative image* (or *Parikh vector*) of σ is $\overrightarrow{\sigma} = (|\sigma|_t)_{t \in \mathcal{T}}$, the vector of the number of occurrences of each transition t in σ. If $M \xrightarrow{\sigma} M'$, then the equation $M' = M + N\overrightarrow{\sigma}$ is satisfied.

Invariants of Petri nets. A vector $J \in \mathbb{N}^{\mathcal{T}}$ (resp. $I \in \mathbb{N}^{\mathcal{P}}$) is a *T-invariant* (resp. *S-invariant*) if $NJ = 0$ (resp. $I^T N = 0$). A T-invariant (resp. S-invariant) is *minimal* if it is minimal for the component-wise ordering among all the T-invariants (resp. S-invariants). A subnet $\mathcal{N}'$ of the Petri net $\mathcal{N}$ with the set of nodes X is a *T-component* (resp. *S-component*) if for every transition of X, ${}^\bullet t \cup t^\bullet \subseteq X$ (resp. for every place of X, ${}^\bullet p \cup p^\bullet \subseteq X$) and $\mathcal{N}'$ is a strongly connected event graph (resp. state machine). A set of T-components (resp. S-components) forms a *T-cover* (*S-cover*) if every node belongs to one of these components. It is well-known that well-formed free-choice nets are covered by T-components and also by S-components ([5, Theorems 6.6 and 5.18]).

We also need the following result.

Theorem 2.1. *[5, Theorem 5.9] Let p be a place of a live and bounded free-choice Petri net $(\mathcal{N}, M)$. The bound of p is* $\min\{\sum_{p \in \mathcal{P}_1} M(p) \mid p \in \mathcal{P}_1,\ (\mathcal{P}_1, \mathcal{T}_1, \mathcal{F}_1)$ *is a S-component of* $\mathcal{N}\}$.

Clusters. The *cluster* $[x]$ of $x \in \mathcal{P} \cup \mathcal{T}$ is the smallest subset of $\mathcal{P} \cup \mathcal{T}$ such that: (i) $x \in [x]$; (ii) $p \in \mathcal{P}, p \in [x] \Rightarrow p^\bullet \in [x]$; (iii) $t \in \mathcal{T}, t \in [x] \Rightarrow {}^\bullet t \in [x]$. The set of all the clusters of a Petri net defines a partition of the nodes of the net. For free-choice nets, each cluster contains only one place or only one transition.

Blocking marking. Let $(\mathcal{N}, M)$ be a Petri net and t a non-conflicting transition of $\mathcal{N}$. A blocking marking of the transition t is a reachable marking such that the only enabled transition is t.

In [7, Theorem 3.1], it is shown that if b is a non-conflicting transition in a LBFC, there exists a unique reachable marking M_b which enables only transition b. Moreover, M_b is reachable from any other reachable marking without firing b. This result can be generalized to conflicting transitions: we then deal with $M_{[b]}$, the blocking marking of the cluster $[b]$.

Timed and routed nets. A timed Petri net is a Petri net on which timings have been added on places and transitions. With no loss of generality, we only consider timings on the transitions, and not on the places. A timed Petri net is denoted by $(\mathcal{N}, M, \tau)$ with $(\mathcal{N}, M)$ a Petri net and $\tau \in \mathbb{R}_+^{\mathcal{T}}$ the vector of the timings. The timed semantics is the following one. Consider a transition t with timing τ_t which gets enabled at instant d. It the transition t is fired, the firing occurs as follows:

- At time d the firing begins. A token is frozen in each input place of t and cannot enable another transition.
- At time $d+\tau_t$ the firing ends. The frozen tokens are removed from the input places of t and one token is added in each output place of t.

The resulting evolution is called *as soon as possible* (*asap*), since a transition begins to fire as soon as it is enabled.

Any conflict-solving policy may be viewed as a set of local *routing functions* at each conflicting place. The global routing function is a vector $u = (u_p)_{p \in \mathcal{P}}$ where u_p is a function from $\mathbb{N}^*$ to $p^\bullet$. The k-th token arriving in the place p (we consider the tokens in place p in the initial marking as the first arriving tokens) can only enable the transition $u_p(k)$. So the notion of enabled transition is modified by the routing function. A transition can be fired if all its input places contain a token which is routed to that transition. We denote by $(\mathcal{N}, M, u)$ a routed Petri net with routing u and by $(\mathcal{N}, M, \tau, u)$ a timed and routed Petri net.

A marking is *reachable* for a routed Petri net $(\mathcal{N}, M, \tau, u)$ if it is reachable for $(\mathcal{N}, M)$ via a firing sequence compatible with u. The notions of boundedness and liveness of $(\mathcal{N}, M, u)$ are defined accordingly.

If the Petri net $(\mathcal{N}, M)$ is bounded, so is the routed Petri net $(\mathcal{N}, M, u)$. If the Petri net $(\mathcal{N}, M)$ is live, $(\mathcal{N}, M, u)$ is not necessarily live, neither deadlock-free. However, if $(\mathcal{N}, M)$ is a live and bounded free-choice net, the routed net $(\mathcal{N}, M, u)$ cannot have any deadlock, because choices and synchronizations are separated. Hence, if a routed free-choice net is not live, we can always define its non-empty live part.

A routing $u = (u_p)_{p \in \mathcal{P}}$ is periodic if u_p is a periodic function for every p. A routing u is 0-1 if: $\forall p \in \mathcal{P}$, u_p is a constant function. A 0-1 routing is called a *total allocation* in [5].

3 Throughput in routed free-choice Petri nets

3.1 Throughput in free-choice Petri nets

Consider a timed Petri net and let $\sigma = \sigma(1)\sigma(2)\cdots \in \mathcal{T}^{\mathbb{N}}$ be an infinite admissible firing sequence. Set $\sigma_n = \sigma(1)\cdots\sigma(n)$. Consider the timed evolution starting at instant 0 and associated with σ. The *activity* $A(\sigma)$ of σ is the asymptotic average number of firings per unit of time:

$$A(\sigma) = \liminf_{n \to \infty} \frac{n}{d(n)}$$

where $d(n)$ is the first instant of completion of all the firings from σ_n.

To make this definition more general and more flexible, it is possible to "weight" the activity of each transition.

A weight $\alpha_t \in \mathbb{R}_+$ is associated to each transition t and we set $\alpha = (\alpha_t)_{t\in\mathcal{T}}$. The *throughput* $D(\sigma)$ of σ (for the weight α) is defined by:

$$D(\sigma) = \liminf_{n\to\infty} \frac{\alpha^T \cdot \overrightarrow{\sigma_n}}{d(n)} .$$

If all the weights are equal to one, the throughput is equal to the activity. On the other hand, if $\alpha_t = 1, \alpha_{t'} = 0, t' \neq t$, then the throughput measures the firing rate of transition t.

The above notion of throughput allows one to modify a Petri net without changing its throughput, for example, by replacing a transition of timing n and weight α by n transitions of timing 1 and weight α/n.

Periodic routings. Consider a timed and routed LBFC $(\mathcal{N}, M, \tau, u)$ with a *periodic* routing and *integer* firing times (rational firing times can be treated in a similar way).

The state of the Petri net at time t is a triple (M_t, R_t, U_t) where M_t is the marking at time t, R_t is the remaining firing time of all the enabled transitions at time t and U_t is the current routing decision in all the routing places. Observe that the number of states is finite and bounded by $(k+1)^{|\mathcal{P}|} \times F^{|\mathcal{T}|} \times L^{|\mathcal{P}|}$, where k is a bound on the number of tokens per place, F is a bound on the firing times of all the transitions and L is a bound on the period of the routing at each place.

Since the behavior of the net is deterministic, the net jumps from one state to its unique successor at each time-step.

The state space being finite, there exists a state which is visited twice for the first time, and the whole behavior becomes periodic from that point on. This shows that the throughput exists and can be computed along the periodic behavior of the net. However, this computation may have a very high complexity (in time and in space) because the state space is potentially huge.

A construction similar to the above one is proposed in [2].

A particular case: event graphs. In a live and bounded event graph, there is no routing place, hence no routing. In that case, it is useless to sweep the whole state space to compute the throughput. It is well-known that the firing rate is the same for all transitions and the throughput can computed in polynomial time (cubic time) by:

$$D = \frac{\sum_{t\in\mathcal{T}} \alpha_t}{\rho(\mathcal{G})}, \quad \text{where} \quad \rho(\mathcal{G}) = \max_{c \text{ circuit of } \mathcal{G}} \frac{\sum_{t\in c} \tau_t}{\sum_{p\in c} M(p)}.$$

The constant $\rho(\mathcal{G})$ is usually called the *cycle time* of $\mathcal{G}$ (see [1, 3, 4] for details).

4 Zero-one policies

In this section, we consider 0-1 routing policies instead of arbitrary periodic routings. We show that all the combinatorial difficulties of periodic routings can be overcome for 0-1 routings.

4.1 Total allocations and 0-1 policies

An *allocation* is a function u from a set of clusters C to $\mathcal{T}$ such that $\forall c \in C,\ a(c) \in c$. A transition is *allocated* if it belongs to the image of a. An allocation is *total* if it is defined on all clusters. An allocation *points to* C if for every place p in $\mathcal{N}$ there exists a path π from p to a place of C such that every transition along the path π is allocated.

A firing sequence σ agrees with an allocation $u : C \to \mathcal{T}$ if it does not contain any transition t such that $[t] \in C$ and $t \neq u(c),\ \forall c \in C$.

Lemma 4.1. *[5, Lemma 6.5] Let C be a set of clusters of a strongly connected free-choice Petri net $\mathcal{N}$ and let $\bar{C}$ be the complementary set of C in the clusters of $\mathcal{N}$. Then there exists an allocation u defined on $\bar{C}$ that points to C, and if M is a bounded marking and $M \xrightarrow{\sigma}$ is an infinite sequence that agrees with u, then some transition of C is fired an infinite number of times in σ.*

In this paper, we see total allocations as 0-1 routing policies in the places: each place routes all its tokens to its unique allocated output transition.

When u is a 0-1 routing, $(\mathcal{N}, u)$ is a free-choice Petri net where all transitions which are not allocated can be removed. Therefore, exactly one output transition remains for each place. We obtain a conflict-free Petri net. We first study these nets before giving a characterization of the live part of a free-choice Petri net with a 0-1 routing.

4.2 Conflict-free nets

A *siphon* is a set of places R such that ${}^\bullet R \subseteq R^\bullet$. A *trap* is a set of places R such that $R^\bullet \subseteq {}^\bullet R$.

Lemma 4.2. *A strongly connected conflict-free net is structurally live.*

Proof. Let $\mathcal{N}$ be a strongly connected conflict-free net. Every place has only one output transition. Then, every circuit is a trap. But a siphon of $\mathcal{N}$ contains at least an elementary circuit. If we define the initial marking to be $M : \forall p \in \mathcal{P},\ M(p) = 1$, then every siphon contains an initially marked trap. By Commoner's Theorem, this Petri net is live, so $\mathcal{N}$ is structurally live.

Lemma 4.3. *Consider a strongly connected and live conflict-free Petri net. It is an event graph if and only if it is bounded.*

It follows from the above lemma, that a connected and live conflict-free Petri net is either unbounded, or bounded in which case it is a strongly connected event graph.

Proof. Let $(\mathcal{N}, M)$ be the conflict-free Petri net. Assume that it is bounded. We are in the domain of application of the Blocking Marking Theorem of [7], recalled in Section 2. We want to show that every place has exactly one input transition. Suppose that there exists a place p with at least two input transitions, t_1 and t_2. Let t be the output transition of p. Let M_t be the blocking marking of t. Recall that M_t is unique and reachable from M, see [7]. From the marking M_t, let us fire the transition t once and block again the transition t and let the net evolve. The transitions that are fired are well-defined as the net is conflict-free. We end up in the marking M_t again. Clearly, only one of the transitions t_1 and t_2 has been fired. Otherwise there would be one additional token in p. Assume it is t_1 which was not fired. Even if we unblock transition t, transition t_1 will never be fired again. Therefore, the Petri net is not live, which is a contradiction. We conclude that every place has exactly one input transition, meaning that the Petri net is an event graph.

Conversely, assume that $(\mathcal{N}, M)$ is an event graph. Any strongly connected event graph is bounded, which completes the proof.

4.3 Live part of a LBFC with a 0-1 routing

Consider a LBFC $\mathcal{N}$ with a 0-1 routing. Let $\widetilde{\mathcal{N}}$ be the Petri net obtained by removing the arcs and transitions which are not chosen by the 0-1 routing. The Petri net $\widetilde{\mathcal{N}}$ is conflict-free, bounded, and composed of non-trivial strongly connected components (scc). There are two kinds of such components: the final components and the non-final components.

Lemma 4.4. *The final components are T-components of $\mathcal{N}$. The non-final components are event graphs, but not T-components of $\mathcal{N}$. The final components may or may not be live in $\widetilde{\mathcal{N}}$, the non-final components are not live in $\widetilde{\mathcal{N}}$.*

Proof. The non-trivial scc's of $\widetilde{\mathcal{N}}$ are strongly connected conflict-free nets. Since the original net is live and bounded, every scc is bounded. Then, from Lemma 4.3, they are event graphs. Let C be a final scc. We modify the allocation as follows: the allocation in C is not modified; outside of C, we choose an allocation that points to C. In the resulting conflict-free Petri net, there is only one final component which is C. From [5, Lemma 8.9], for a total allocation, applied from a marking that marks every place, there exists a T-component which is live and where every transition is allocated. This can only be C, which is consequently a T-component.

A non-final scc C is not a T-component. Indeed, since it is non-final and conflict-free, there must exist a transition $t \in C$ with an output place $p \notin C$. Moreover, if C was to be live, then the place p could receive an infinite number of tokens, implying that the original Petri net $\mathcal{N}$ is unbounded. So C is not live.

Theorem 4.1. *The live part of $\widetilde{\mathcal{N}}$ is a non-empty disjoint union of T-components of $\mathcal{N}$. For every T-component $\mathcal{N}'$ of the LBFC $\mathcal{N}$, there exists a 0-1 routing that makes $\mathcal{N}'$ the live part of $\widetilde{\mathcal{N}}$. Moreover, the resulting marking of a circuit of $\mathcal{N}'$ does not depend on the 0-1 routing such that $\mathcal{N}'$ is the live part of $\widetilde{\mathcal{N}}$.*

Proof. The first part of the result follows from Lemma 4.4. Now fix a T-component $\mathcal{N}'$. Let C be the set of clusters of $\mathcal{N}'$. Define an allocation that points to C, see Lemma 4.1. For the routing places of $\mathcal{N}$ which belong to $\mathcal{N}'$, route the tokens within $\mathcal{N}'$. Globally this defines a 0-1 routing such that the Petri net $\widetilde{\mathcal{N}}$ has only one final scc which is $\mathcal{N}'$. Since the non-final scc are not live, and since the live part of $\widetilde{\mathcal{N}}$ is non-empty (otherwise, it would contradict the liveness of $\mathcal{N}$), we conclude that the live part of $\widetilde{\mathcal{N}}$ is precisely $\mathcal{N}'$.

By uniqueness of the blocking marking of a transition of a T-component, we get the uniqueness of the total number of tokens on a circuit.

4.4 On the circuits of free-choice nets

Lemma 4.5. *In a LBFC, every elementary circuit is included in a T-component of the net.*

Proof. Let c be an elementary circuit of $\mathcal{N}$, a LBFC, and u a 0-1 routing such that on c, every transition is chosen by the routing, and such that for every other place of choice, the chosen transition is a transition which is on a shortest path to c. The conflict-free Petri net corresponding to that 0-1 routing has only one final scc, so it is the live part of the net. Moreover, this scc contains c. The live part of the net is a T-component, so c is included in that T-component.

Corollary 4.1. *Let $\mathcal{N}$ be a LBFC and let c be an elementary circuit of $\mathcal{N}$. For every T-invariant J of $\mathcal{N}$ such that $J \geq \chi_c$, there exists a minimal T-invariant J' of $\mathcal{N}$ such that $J \geq J' \geq \chi_c$.*

Proof. The vector J is a T-invariant, so it can be decomposed into a sum of minimal T-invariants $J = \sum_{i=1}^{k} J_i$, where J_i is a minimal T-invariant for all i. Now consider the net $\mathcal{N}'$ generated by the transitions that appear in J and their output and input places. The circuit c is a circuit of $\mathcal{N}'$, and from Theorem 4.5, it belongs to a T-component of $\mathcal{N}'$, which is also a T-component of $\mathcal{N}$. Then there exists J' a minimal T-invariant (corresponding to the T-component) such that $J' \leq J$ and $J' \geq \chi_c$.

4.5 Throughput of 1-bounded free-choice nets with 0-1 routings

Lemma 4.6. *Let $(\mathcal{G}, M)$ be a live and 1-bounded event graph. Then*
- *if one token is removed, the graph is not live anymore,*
- *if one token is added, the graph is not 1-bounded anymore.*

Proof. Let p be a place such that $M(p) = 1$ and M' be the marking obtained from M by removing the token of p. If $(\mathcal{N}, M')$ were live, then there would exist an admissible firing sequence σ such that $M' \xrightarrow{\sigma} M''$ with $M''(p) = 1$. But this sequence would also be admissible for $(\mathcal{N}, M)$, and there would be 2 tokens in p. This is a contradiction with the 1-boundedness of $(\mathcal{N}, M)$. Therefore, $(\mathcal{N}, M')$ is not live.

If a token is added in $(\mathcal{N}, M)$, then the net is still live. If that token is removed, then the net is still live (this is the original net). Then, when a token is added, the net is not 1-bounded anymore.

Theorem 4.2. *When the live part of a net with a 0-1 routing is a single T-component, $\mathcal{G}$, corresponding to the T-invariant $J_{\mathcal{G}}$, the throughput is*

$$D_{\mathcal{G}} = \frac{\alpha^T J_{\mathcal{G}}}{\rho(\mathcal{G})}.$$

If the live part of a live 1-bounded free-choice net with a 0-1 routing is a disjoint union of T-components, $\mathcal{G}_1, \cdots, \mathcal{G}_k$, the throughput of the net is

$$D = \sum_{i=1}^{k} D_{\mathcal{G}_i} = \sum_{i=1}^{k} \frac{\alpha^T.J_{\mathcal{G}_i}}{\rho(\mathcal{G}_i)}.$$

Proof. From Theorem 4.1, the marking on circuits does not depend on the routings that makes $\mathcal{G}$ be the live part of the net. So, the same holds for the cycle time, that depends only on the timings and the markings of circuits.

Now, we show that if a T-component is live, then the marking is the same as if it was the only live T-component. If there were a circuit of $\mathcal{G}_i$ with more tokens than if it were the only live T-component, then the net wold not be 1-bounded, and if there were a circuit with fewer tokens, the net would not be live, from Lemma 4.6.

5 Token-Transition-Invariants

Definition 5.1 (compatible firing sequence). *Let $\pi = p_1 t_1 p_2 t_2 \cdots p_\ell t_\ell$, $p_i \in \mathcal{P}$, $t_i \in \mathcal{T}$ be a path of $\mathcal{N}$ and M be a marking that marks p_1. Let σ be an admissible occurrence sequence of $(\mathcal{N}, M)$. The sequence σ is compatible with π if the first sub-word of σ in $[t_1][t_2] \cdots [t_\ell]$ is $t_1 t_2 \cdots t_\ell$.*

In other words, a firing sequence is compatible with a path in a Petri net if all transitions along that path are fired in that order. It means that the token which was initially in place p_1 successively enters places $p_2 \cdots p_\ell$ when σ is fired.

Definition 5.2 (Token-Transition-Invariant). *Let $c = p_1 t_1 p_2 t_2 \cdots p_\ell t_\ell$ be a circuit and M be a marking that marks p_1. The vector J is a Token-Transition-invariant (or TT-invariant), generated by c and the marking M if it is a T-invariant and if it is the commutative image of an admissible firing sequence compatible with c.*

A TT-invariant J generated by c and the marking M is minimal if for every other TT-invariant J' generated by c and the marking M, $J \leq J'$.

A TT-invariant generated by c is minimal if it is minimal for the coordinate-wise ordering among all the TT-invariants generated by c and a reachable marking.

Summarizing the discussion started before Definition 5.2, a TT-invariant is a T-invariant such that one token has moved along a circuit and is back to its original place when the corresponding sequence is fired (hence the name).

In spite of what the definition suggests, TT-invariants generated by c do not depend on the initial marking: if the commutative image of $\sigma_1 \cdot \sigma_2$ is a TT-invariant for c, so is $\sigma_2 \cdot \sigma_1$, as soon as it is admissible from the marking M' such that $M \xrightarrow{\sigma_1} M'$.

However, unlike general T-invariants, TT-invariants depend on the set of reachable markings. We will see in the following that it actually depends on the maximal number of tokens in circuit c.

Theorem 5.1. *Let c be an elementary circuit of a live and 1-bounded event graph containing n tokens. The minimal TT-invariant generated by c is $n\mathbf{1}$.*

Proof. Recall that the T-invariants of the event graph are of the form $x\mathbf{1}, x \in \mathbb{N} \setminus \{0\}$, see [5, Prop. 3.16]. Now, let J be a minimal TT-invariant associated with $c = p_1 t_1 \cdots$, and let σ be a corresponding firing sequence. Since the Petri net is 1-bounded, tokens along the circuit c cannot overcome each other. Hence, when the token initially in p_1 is back to p_1, after the firing of σ, we know that transition t_1 must have fired n times.

Example 5.1. In Figure 1 is shown the evolution of an event graph containing 2 tokens in circuit c. We look at the minimal TT-invariant generated by c. Using Theorem 5.1, the minimal TT-invariant is $(2, 2, 2, 2, 2, 2, 2, 2, 2, 2)$. The white token (as well as the black one) is back to its original place (Figure 1(c)).

The minimal T-invariant is $(1, 1, 1, 1, 1, 1, 1, 1, 1, 1)$. Note that, after a single firing of every transition (Figure 1(b)), the marking is unchanged , but the white token has switched its position with the black one. After firing every transition again (Figure 1(c)), the white token is back in the right place.

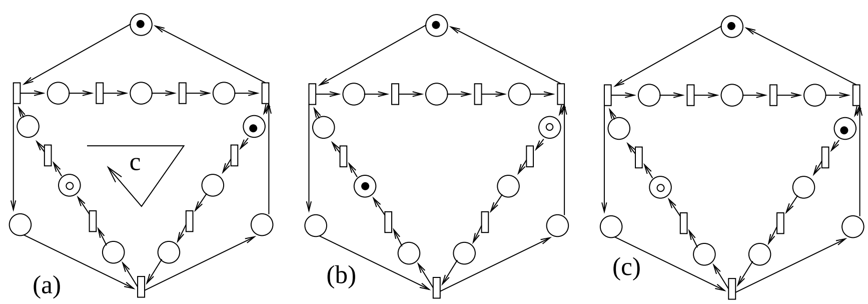

Fig. 1. TT-invariant in an event graph

Consider an elementary circuit c of a live and 1-bounded free-choice net that can contain (for a reachable marking) at least two tokens. As the net is 1-bounded, each place of c belongs to an S-component containing only one token

(Theorem 2.1), and therefore to an elementary circuit containing at most one token.

We call a *section* of c the non-empty intersection of c with an elementary circuit containing at most one token. From what was said above, the set of sections of c is a cover of c. We now consider a cover of c by the sections $S_1, \cdots, S_m$ such that $\forall i, j,\ S_i \nsubseteq S_j$. The section S_i begins with the transition t_i and ends with the transition t'_i. We denote $\{t_1, \cdots, t_m\}$ by $\mathcal{T}_c$.

Lemma 5.1. *Consider a path $\pi = (p_1, \cdots, p_k)$ with $p_1 \in S_i \setminus c$ and $p_k \in S_i$. If σ is a firing sequence compatible with π, then $|\sigma|_{t_i} \geq 1$.*

Proof. Suppose that such a path exists and that there is a compatible sequence σ with π such that $|\sigma|_{t_i} = 0$. We also suppose that p_1 is a conflict place. Let u be an allocation pointing to p_1 such that every transition of π and of c is allocated. The live part the net with that allocation is strongly connected, by Lemma 4.1, and for an infinite occurrence sequence, a transition of $[p_1]$ is fired infinitely often, so it contains p_1, c and π. The live part of the net is by construction live, bounded and free-choice. Then it is covered by T-components. Let G be the T-component containing c, and p_l be the first node on π not in G. It is necessary a place by definition of G. Now, p_j is necessary in c and not in S_i: otherwise, the number of tokens would have changed in the circuits of G, which could lead to a dead-lock. As a consequence, to add a token in S_i, t_i must be fired.

Lemma 5.2. *Consider a path $\pi = (p_1, \cdots, p_k)$ with $p_1 \in S_i$ and $p_k \in S_i \setminus c$. If σ is a firing sequence compatible with π, then $|\sigma|_{t'_i} \geq 1$.*

Proof. Well-formed free-choice nets are stable under the operation that consists in replacing transitions by places and vice versa, and inverting the direction of every arc. With this operation, we get the result from the previous lemma: t_i, t'_i, p_1, p_k are respectively replaced by p_1, p_k, t_i, t'_i.

Theorem 5.2. *Let $\mathcal{N}$ be a live and 1-bounded free-choice net, c be an elementary circuit of $\mathcal{N}$ and n be the maximal number of tokens in c. Then the minimal TT-invariants generated by c are of the form $J = \sum_{i=1}^{n} J_i$ where, for each i, J_i is a minimal T-invariant containing the set $\mathcal{T}_c$ defined above.*

Proof. If the circuit c contains at most one token, then a T-invariant containing c is a minimal TT-invariant generated by c. Indeed, it suffices to apply Theorem 5.1 to the routed net with a 0-1 routing that makes a T-component containing c be the live part of the net.

Now, we consider that the maximum number of tokens in c is $n \geq 2$. To fire the transition t_i, there must be no token in S_i. If there is one, then t'_i must be fired before, but then t_{i+1}, as the sections form a cover of c. As a consequence, every transition beginning a section must fire n times.

Note that if a T-invariant contains a transition $t_i \in \mathcal{T}_c$, it contains also the other transitions $t_j \in \mathcal{T}_c$.

Example 5.2. Figure 2 illustrates Theorem 5.2: consider the circuit $c = 1, 2, 3, 5, 6, 7, 9, 10, 11$. It is covered by three sections : $(1, a, 2, b, 3, c, 5)$, $(5, e, 6, f, 7, g, 9)$ and $(9, i, 10, j, 11, k, 1)$. The minimal TT-invariants of c are the sum of the T-invariant containing c and of a T-invariant containing the transition $1, 5, 9$. For example, this last T-invariant can be associated with c or with $\gamma = 1, 4, 5, 8, 9, 12$.

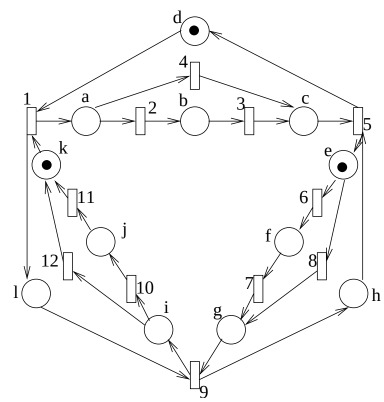

Fig. 2. Example of a free-choice net.

Lemma 5.3. *Let $\mathcal{C}$ be a circuit in a live and 1-bounded free-choice net and J be a minimal TT-invariant of $\mathcal{C}$. The circuit is composed of the elementary circuits $c_1, \cdots, c_k$. Then there exists k minimal TT-invariants, $J_1, \cdots, J_k$ respectively generated by $c_1, \cdots, c_k$ such that $J \geq \sum_{i=1}^{k} J_i$.*

Proof. The result is proved by induction. We assume that the proposition $(H(n))$ is true for a circuit that is composed of n elementary circuits.
First, note that $H(1)$ is true: this is Proposition 5.2.
Now, let $\mathcal{C}$ be a circuit composed of $n+1$ elementary circuits. Let J be a minimal TT-invariant generated by $\mathcal{C}$. One elementary circuit can be removed from $\mathcal{C}$ such that the remaining is still connected. We denote by c_{n+1} the removed circuit, and $\mathcal{C}'$ the circuit obtained by removing c_{n+1} from $\mathcal{C}$. It is composed of n circuit, $c_1, \cdots, c_n$, and that we can apply the recurrence hypothesis $H(n)$: there exists $J' \leq J$ (fewer transitions to fire) a minimal TT-invariant for $\mathcal{C}'$ such that $J' \geq \sum_{i=1}^{n} J_i$, where J_i is a minimal TT-invariant generated by c_i. If c_{n+1} is added to $\mathcal{C}'$, we get $\mathcal{C}$ again.

If there are at most k tokens in c_{n+1}, then $J' + \chi_{c_{n+1}} + (k-1)\chi_{t_1 \cdots t_\ell} \leq J$, where $t_1, \cdots, t_\ell$ are the transition beginning the sections of a cover of c_{n+1}. As $J - J'$ is a semi-positive T-invariant, greater than $\chi_{c_{n+1}} + (k-1)\chi_{t_1 \cdots t_\ell}$, $J - J'$ is greater that a minimal TT-invariant generated by c_{n+1}, from Theorem 5.2. Then $H(n+1)$ is verified.

6 Extremal Throughputs

As stated in the introduction, we now consider all possible routing policies and try to address the following questions: can one compute the routing policy which yields the best (or worst) throughput? Are the best and worst policies periodic? Are the best and worst policies 0-1?

6.1 Dominance of periodic policies

In this section we show that the best and worst throughputs in LBFC with rational firing times are achieved by periodic policies, and we provide an algorithm to construct them.

Marking graph of a free-choice net. Suppose that a timed Petri net $(\mathcal{N}, M, \tau)$ has all its timings equal to 1: $\tau_t = 1\ \forall t \in \mathcal{T}$. All the *asap* behaviors of the net can then be read on the asap *marking graph* defined as follows.

$Q \leftarrow \{M\}$; $\tilde{Q} \leftarrow \{M\}$;
while $\tilde{Q} \neq \emptyset$ **do**
  Pick $M' \in \tilde{Q}$
  for all maximal multi-set T of transitions that can be fire simultaneously from M' **do**
    $M'' \leftarrow M' + N.T$;
    if $M'' \notin Q$ **then**
      $Q \leftarrow Q \cup \{M''\}$; $\tilde{Q} \leftarrow \tilde{Q} \cup \{M''\}$;
    Add the arc $M' \stackrel{T|\sum_{t\in T}\alpha_t}{\longrightarrow} M''$;
  $\tilde{Q} \leftarrow \tilde{Q} \setminus \{M'\}$

This construction stops because the net is bounded. This way, we can model the behavior of all timed nets with rational timings: to be back to the case of timings 1, it suffices to set the unit of time as the lowest common denominator of the timings and duplicate a transition of timing $\tau \in \mathbb{N}$ and weight α into τ transitions of timing 1 and weight α/τ.

Theorem 6.1. *Let $(\mathcal{N}, M, \tau)$ be a timed LBFC with rational timings. The minimal and maximal throughputs are obtained for periodic routings.*

The proof is not essential and is postponed to the appendix because of lack of space (for reviewing purposes).

Example 6.1. We consider the net of Figure 2 as example throughout the article. Every transition has timing 1 and weight 1. The *asap* marking graph is represented in Figure 3. The minimal average weight of a cycle is 15/9, given by the cycle $\{(dek), (afl), (cgl), (chi), (dej), (dgk), (agl), (bhi), (chk)\}$. This gives the minimal throughput and the routing to reach it. The maximal average weight is 2. That weight is reached for the cycle $\{(del), (agl), (chi)\}$. So the maximal throughput is 2.

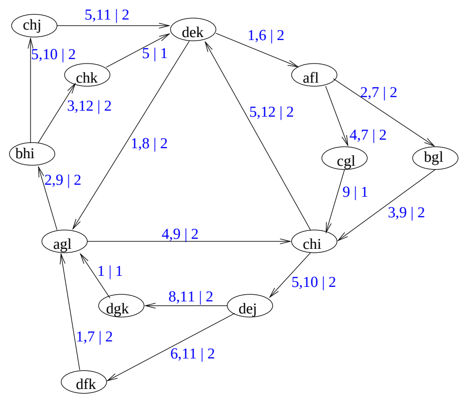

Fig. 3. (max,+) automaton corresponding to Petri net of figure 2.

When the graph is built, computing the throughput can be made in cubic time in the number of nodes of the *asap* marking graph. But this graph can have an exponential size in the size of the original Petri net. One reason is that the number of marking of the net can be exponential in the size of the net. The other reason is that transitions are duplicated to build the Petri net with timings 1 starting from a Petri net with rational timings. Moreover, this method gives no information about the structure of the corresponding extremal policies.

For general irrational timings, the maximal throughput is not always periodic, as shown in the following example.

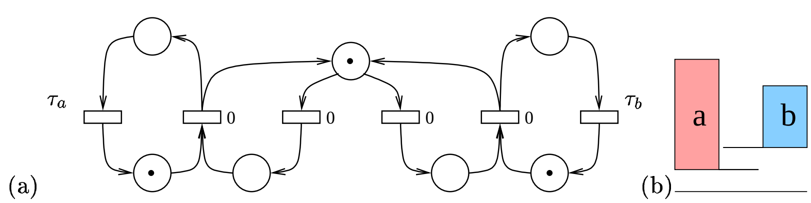

Fig. 4. Net for which the optimal routing is not periodical.

Example 6.2. Look at figure 4. The left T-component is denoted by a and the right one by b. We have $\tau_a/\tau_b \notin \mathbb{Q}$. The firing of the transitions of a can be represented by the piece a and that of b by a piece b, as shown on figure 4(b). We are in the case studied in [8]. The optimal routing is Sturmian aperiodic. The best routing consists in choosing the transition of the T-component a if a token appear in p_a before p_b, and else choosing the transition of the T-component b. If t_a has fired n_a times and t_b n_b times, it suffices to compare $n_a\tau_a$ and $n_b\tau_b$. The non-periodicity comes from the irrationality of the ratio of the timings.

6.2 Minimal throughput in 1-bounded free-choice nets

The Petri nets considered here are all live and 1-bounded timed free-choice nets. In this part, we show that the minimal throughput is obtained for a periodical routing even for general timings. Furthermore, we give a precise insight on the structure of the periodic routing reaching the minimal throughput. Roughly, this periodic routing is "almost 0-1".

From Theorem 4.2 we can easily deduce the following lemma:

Lemma 6.1. *The worst 0-1-routing is among those that make only one T-component live.*

Critical circuit. Suppose again that the timings are 1. The worst routing can be read on the *asap* marking graph and we can choose if to consist in following an elementary circuit, $c = (M_0, \cdots, M_{T-1})$, of minimal average weight. The length of c is T, the period of the evolution.

Let t_0 be a live transition of the net appearing in the label between states M_0 and M_1 of the circuit c. We build a path in the net with final extremity t_0 in the following recursive way.

If the path $t_i, \cdots, t_1, t_0$ is built, we choose the transition t_{i+1} in the label of the arc between $M_{T-i-1 \mod T}$ and $M_{T-i \mod T}$ and $t_{i+1} \in {}^{\bullet\bullet}t_i$.

We stop the construction when we find $n \in \mathbb{N}^*$ and a transition t_i such that $t_i = t_{i-nT}$. Indeed, the routing is periodic, so the path is periodic with period nT. The "critical circuit" is then $t_i, t_{i-1}, \cdots, t_{i-nT}$, denoted by $\mathcal{C}_c$. The length of this circuit is nT by construction.

The commutative image of $t_{i-1} \cdots t_{i-nT}$ is a TT-invariant: the net is back to the same marking, and by construction, one token is back to its original place.

Lemma 6.2. *The critical circuit of $(\mathcal{N}, M)$ is an elementary circuit of $\mathcal{N}$.*

Proof. Let J be the commutative image of $t_{i-1} \cdots t_{i-nT}$. The throughput is the ratio between the weight of transitions fired during a period and the length of this period. We have:

$$D = \frac{\alpha^T J}{nT}.$$

As J is a TT-invariant for $\mathcal{C}_c$, there exists J_c a minimal TT-invariant for $\mathcal{C}_c$ such that $J \geq J_c$. The critical circuit composed of k elementary circuits $c_1, \cdots, c_k$ of length $\ell_1, \cdots, \ell_k$, with $\sum_{i=1}^k \ell_i = nT$. So, J_c can be decomposed in k minimal cyclic T-invariants of $c_1, \cdots, c_k$, $J_1, \cdots, J_k$. Then,

$$D = \frac{\alpha^T J}{nT} \geq \frac{\alpha^T J_c}{nT} = \frac{\alpha^T (\sum_{i=1}^k J_i)}{\sum_{i=1}^k \ell_i} = \frac{\sum_{i=1}^k \alpha^T J_i}{\sum_{i=1}^k \ell_i} \geq \min_{i=1}^{k} \frac{\alpha^T J_i}{\ell_i}.$$

Hence, $\mathcal{C}_c$ is an elementary circuit.

Theorem 6.2. *The minimal throughput is obtained for a periodical routing, where the period is less than the number of tokens in the net. The routing is 0-1 everywhere except on an elementary circuit.*

Proof. We first consider $(\mathcal{N}, M, \tau)$ a live and 1-bounded free-choice net with rational timings. We show the result $\mathcal{N}$ with the *asap* marking graph obtained by duplicating the transitions so the timings are 1 everywhere. Then, the length of a circuit c, denoted by $|c|$ is the sum of the timings of the circuit.

The critical circuit c of the net is elementary. The minimal TT-invariants generated by c is of the form

$$J = \sum_{i=1}^{n} J_i \qquad J_i \text{ minimal T-invariant},$$

with $J_1 \geq \chi c$, and $J_i \geq \chi_{\mathcal{T}_c}$, and n is the maximal number of tokens in c.

Let J_a be a T-invariant of minimal weight such that $J_a \geq \chi_c$ ($\alpha^T J_a = \min\{\alpha^T J | J \geq \chi_c\}$) and J_b be a T-invariant of minimal weight such that $J_b \geq \chi_c$ ($\alpha^T J_b = \min\{\alpha^T J | J \geq \chi_c\}$).

1. If $J_a = J_b$, then the TT-invariant generated by c of minimal weight is nJ_a. The cycle time of the T-component corresponding to J_a is $|c|/n$ (else c would not be critical). The minimal throughput is $\frac{\alpha^T J_a}{|c|}$ and is obtained for a 0-1 routing.
2. Else we necessary have $\alpha^T J_b \leq \alpha^T J_a$. If the cycle time of the T-component corresponding to J_b were greater than $|c|/n$, then c would not be the critical circuit. So, the minimal TT-invariant of minimal weight is $J_a + (n-1)J_b$ and the minimal throughput is $\frac{\alpha^T(J_a+(n-1)J_b)}{|c|}$.

Moreover, remark that the circuits in J_a and J_b have a cycle time less than $|c|/n$. As a consequence, the routings are 0-1 except on circuit c, where the routings have period n.

This theorem also holds for irrational timings. Due to lack of space, the proof is postoned in appendix (for reviewing purposes).

Example 6.3. Take again the example of Figure 2. The critical circuit is $(1, 2, 3, 5, 6, 7, 9, 10, 11)$. The minimal T-invariant containing this circuit is the T-invariant containing exactly those transitions. This circuit contains two tokens, so the minimal TT-invariant generated by this circuit is $I + J$ where I is the minimal T-invariant containing the transitions $(1, 2, 3, 5, 6, 7, 9, 10, 11)$ and J is a minimal T-invariant containing the transitions $1, 5, 9$. The minimal T-invariant of minimal weight containing those transitions is $(1, 4, 5, 8, 9, 12)$. The minimal throughput is then $(9 + 6)/9$ as announced before. As for the worst routing policy, it can be obtained directly from the minimal TT-invariant: $u_a = (2, 4)^\infty, u_e = (8, 6)^\infty$ and $u_i = (1, 10)^\infty$.

6.3 Algorithm to compute a routing that minimizes the throughput

Let T be a subset of $\mathcal{T}$. Let **Minimal-T-invariant**(T) be the function that computes a T-invariant of minimal weight that contains the transitions of T. It is the solution of the following linear programming problem:

Input: $T \subseteq \mathcal{T}$
Minimize $\alpha^T.I$
With constraints $N.I = 0$; $I \geq \chi_T$.

This function is in polynomial time because it is linear programming over $\mathbb{Q}$. This function is called when T is a singleton or represents the transitions of a circuit. Then the solution will always be integer: in the case of one transition, the minimal T-invariant containing T and minimizing $\alpha^T I$ is the solution. In the other case, the solution is a minimal T-invariant that contains the circuit represented by the transitions. In both cases, the solution is a minimal T-invariant, and the coordinates are in $\{0, 1\}$.

Before giving the algorithm, let define the functions **Blocking-Marking**(t), **Cycle-time**(I), **Compute-Tc**(c), **Length**(c) and **Marking**(c, M).

- **Blocking-Marking**(t) computes the blocking marking of transition t (or of the cluster containing it). This marking can be computed in time $O(|\mathcal{T}|^3)$ ([7]).
- **Cycle-time**(I, M) computes the cycle time of the event graph corresponding to the minimal T-invariant I with marking M. The time complexity is $O(|\mathcal{T}|^3)$.
- **Length**(c) computes the length of circuit c. The time complexity is linear.
- **Marking**(c, M) computes the number of tokens in circuit c under the marking M. The time complexity is linear.
- **Compute-Tc**(c, I) finds a transition in $\mathcal{T}_c$. It has a time complexity $O(|\mathcal{T}|^3)$. It is given by Algorithm 1 where **Dfs-red**(t', I, col) is a deep-first-search algorithm of the graph associated to I from t', but no red arc or red node can be visited. The algorithm stops when a blue transition is visited, and that transition is returned. Else, the value FAILS is returned.

Algorithm 1 Compute-Tc(c, I)

Input: A circuit c and a T-invariant I
$G \leftarrow$ net generated by I;
for all $t \in c \cup \mathcal{T}$ **do**
  $col[t] \leftarrow$ red;
for all $f \in c \cup \mathcal{F}$ **do**
  $col[f] \leftarrow$ red;
Let $t \in c \cap \mathcal{T}$;
Let $t' = {}^{\bullet\bullet}t \cap c$;
$col[t] \leftarrow$ blue;
while Dfs-red(t', t, I, col)= FAILS **do**
  $col[t'] \leftarrow$ blue;
  $t' \leftarrow {}^{\bullet\bullet}t' \cap c$;
RETURN Dfs-red(t', I, col);

This function finds a transition of $\mathcal{T}_c$. Indeed, (u, t') defines a section, as there is a circuit containing u and t' whose intersection with c is the path from u to t' in c, by construction. Moreover, this section is maximal: this is a maximal section containing the place p that follows transition t in c. We make at most $|c|$ deep-first-search, whose complexity is $O(|\mathcal{T}|^2)$. Thus, the complexity is cubic.

Algorithm 2 finds the minimal throughput of a 1-bounded free-choice net.

Algorithm 2 Worst-routing

Input: $(\mathcal{N}, M, \tau)$ a timed, live and bounded free-choice net.
Throughput $\leftarrow +\infty$;
Tmin $\leftarrow \mathbf{0}$;
for all elementary circuit c of $\mathcal{N}$ **do**
  $I \leftarrow$ **Minimal-T-invariant**(c);
  Let t be such that $I_t = 1$. $M \leftarrow$ **Blocking-Marking**(t);
  $\rho \leftarrow$ **Cycle-time**(I, M);
  $l \leftarrow$ **Length**(c); $n \leftarrow$**Marking**(c, M);
  if $\rho = l/n$ **then**
    if $n = 1$ **then**
      if Throughput $\geq \frac{\alpha^T I}{\rho}$ **then**
        Throughput $\leftarrow \frac{\alpha^T I}{\rho}$;
        Tmin $\leftarrow I$;
  else
    $T_c \leftarrow$**Compute-Tc**(c, I);
    $J \leftarrow$ **Minimal-T-inv**(T_c);
    if **Cycle-time**$(J) \leq \rho$ **then**
      if Throughput $\geq \frac{\alpha^T (I+(n-1)J)}{l}$ **then**
        Throughput $\leftarrow \frac{\alpha^T (I+(n-1)J)}{l}$;
        Tmin $\leftarrow I + (n-1)J$;

At each step, the time complexity is polynomial (more precisely cubic) and there are as many steps as elementary circuits in the net. Therefore, the total time complexity is $O(|\mathcal{C}||\mathcal{T}|^3)$, where $|\mathcal{C}|$ is the number of circuits. Since the number of circuits can be exponential in the number of tranistions ($O(2^{|\mathcal{T}|})$), the time complexity is exponential in the worst case. As for the space complexity, it remains linear in the size of the Petri net. On the other hand, the general method of computation given in part 6.1 is in exponential time: the automaton that is computed is exponential in the size of the Petri net (this is the *asap* marking graph). More precisely, the complexity in time is $O((2^{|\mathcal{P}|})^3)$, that is $O(8^{|\mathcal{P}|})$. The space complexity is also exponential (at least $O(2^{|\mathcal{P}|})$), which makes it less efficient than the algorithm presented above (with an exponential decrease factor in time complexity and a linear space complexity instead of an exponential one). We do not know of any algorithm in the general case which is less than exp-space.

6.4 Bounded nets

If one considers a live and k-bounded free-choice net with $k \geq 2$, then the previous constructions for 1-bounded nets do not work anymore. Moreover, the worst routing is not 0-1 nor "almost 0-1" as shown by the following example.

Example 6.4. Figure 5(a) represents a 2-bounded free-choice net where period of the worst routing is greater that the number of tokens in any circuits, and the critical circuit for this routing is not elementary.

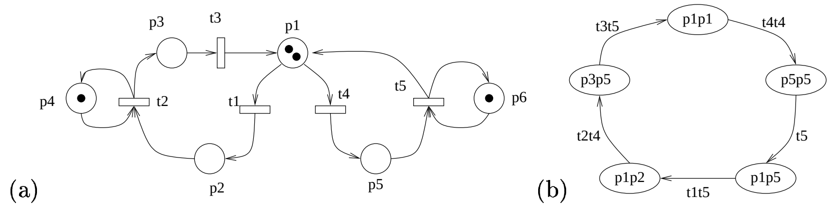

Fig. 5. free-choice net where the worst routing is not 0-1 and its worst evolution.

All the timings are equal to 1, as well as the weights. The routing which gives the minimal throughput is, for p_1, $u = (t1t4t4t4)^*$, and the periodical evolution is given in Figure 5(b). The throughput of this evolution is 9/5, whereas if only the left (or right) event graph is live the throughput is 2. The period of the routing function of place p_1 of the example is greater than the number of tokens in the circuits containing this place (2 tokens), as opposed to what happens in the case of 1-bounded nets. Even in the case of 2-bounded nets, bounding the period of the worst routing seems difficult.

References

1. F. Baccelli, G. Cohen, G.J. Olsder, and J.P. Quadrat. *Synchronization and Linearity.* John Wiley & Sons, New York, 1992.
2. J. Carlier and P. Chretienne. Timed Petri net schedules. In *Advances in Petri Nets*, number 340 in LNCS, pages 62–84. Springer Verlag, 1988.
3. P. Chretienne. *Les Réseaux de Petri Temporisés.* PhD thesis, Université Paris VI, Paris, 1983.
4. G. Cohen, D. Dubois, J.P. Quadrat, and M. Viot. A linear system-theoretic view of discrete-event processes and its use for performance evaluation in manufacturing. *IEEE Trans. Automatic Control*, 30:210–220, 1985.
5. J. Desel and J. Esparza. *Free Choice Petri Nets*, volume 40 of *Cambridge Tracts in Theoretical Comp. Sc.* Cambridge Univ. Press, 1995.
6. S. Gaubert and J. Mairesse. Modeling and analysis of timed Petri nets using heaps of pieces. *IEEE Trans. Aut. Cont.*, 44(4):683–698, 1999.
7. B. Gaujal, S. Haar and J. Mairesse. Blocking a transition in a free-choice net and what it tells about its throughput. *Journal of Computer and System Sciences*, 66(3):515-548, 2003.
8. J. Mairesse and L. Vuillon. Optimal sequences in a heap model with two pieces. Liafa research report 98/09, Université Paris 7, 1998.

Appendix: Proof of theorem 6.1

Consider the *asap* marking graph of $\mathcal{N}$ with duplicated transition so that every transition has timing 1. The number of reachable marking is finite. Then, for every infinite evolution of $\mathcal{N}$, there exists a state (marking) that is visited infinitely often on the graph. Let c be a circuit of the graph of maximum average weight. The maximal throughput can be reached by following this circuit: if s is a state that is visited infinitely often during an evolution, and that several circuits are visited, then the throughput is an average of the average weights of the circuits that are visited. Then it cannot be greater than the evolution consisting in following a circuit of maximal average weight.

With the same reasoning, we show that the minimal throughput can be reached by following the circuit of minimal average weight. This shows that the routings are periodic: the routing can be read on those circuits, and then have a period less than the length of the circuits.

Appendix: Proof of theorem 6.2 for irrational timings

We first show that the minimal throughput is a right-continuous function of the timings.

Lemma .3. *Let* $(\mathcal{N}, u, \tau)$ *a live and 1-bounded routed and timed free-choice net. Let* α *be a weight on the transitions. Suppose that the throughput is defined for this routing. Let* t_0 *a transition of the net. Then the throughput in function of* τ_{t_0} *is continuous.*

Proof. We use here the representation of 1-bounded free-choice nets by heaps of pieces (see [6]). As the net is routed, the sequence of the firings is exactly defined (up to commutation between concurrent transitions), and independently of the timings. When the pieces are piled one by one to model the behavior of the net, we denote by H_n the height of the heap after piling n-th piece, and A_n the weight of H_n. The throughput is then the limit when n tends to infinity of A_n/H_n.

We change the timing of t_0 to $\sigma_{t_0} + \varepsilon$, $\varepsilon > 0$ (instead of σ_{t_0}). The weight of the heap is still A_n, and the height is then $H'_n \leq H_n + K_n.\varepsilon$ where K_n is the number of occurrences of t_0. The new throughput is the limit when n tends to infinity of A_n/H'_n, and we have $A_n/H_n \geq A_n/H'_n \geq A_n/(H_n + K_n.\varepsilon)$. So,

$$\begin{aligned} \frac{A_n}{H_n} - \frac{A_n}{H'_n} &\leq \frac{A_n}{H_n} - \frac{A_n}{H_n + K_n.\varepsilon} \\ &\leq \frac{A_n}{H_n}.\frac{K_n\varepsilon}{H_n + K_n\varepsilon} \\ &\leq \frac{A_n}{H_n}.\frac{K_n\varepsilon}{H_n} \end{aligned}$$

As $H_n \geq nt_{max}/C$ where t_{max} is the greater firing time and C the number of columns of the heap and as $K_n \leq n/\sigma_{t_0}$, we have:

$$\frac{A_n}{H_n} - \frac{A_n}{H'_n} \leq \frac{A_n}{H_n} . \frac{n/\sigma_{t_0}\varepsilon}{nt_{max}/C} = \frac{A_n}{H_n} . \frac{C}{t_{max}\sigma_{t_0}} .\varepsilon.$$

We conclude that the throughput of the net with the timing $\sigma_{t_0} + \varepsilon$ tends to the throughput of the initial timings when ε tends to zero.

Proof of Theorem 6.2 (irrational case). The same can be done with all the transitions. So, for a given routing function, the throughput functions continuous in the timings of the transitions. This function is also decreasing (the height of the heap increases when increasing the timings).

The function giving the minimal throughput is an infimum on all the throughput functions. This function is then lower semi-continuous and, as it is decreasing, the function is right-continuous.

We denote by $d : \mathbb{R}_+^n \to \mathbb{R}_+^n$ the throughput function, which is function of the timings (there are n transitions). This function is right-continuous. Let $x = (x_1, \cdots, x_n) \in \mathbb{R}^n$. We approach x with a sequence of rational n-tuples $(x'_k)_{k\in\mathbb{N}}$ satisfying $\forall k \in \mathbb{N}$, $x'^{(k)} \geq x$.

$$d(x_1, x_2 \cdots, x_n) = \lim_{k\to\infty} d(x_1'^{(k)}, x_2'^{(k)}, \cdots, x_n'^{(k)}). \tag{1}$$

Let m be the maximal number of tokens in a circuit of the free-choice net $\mathcal{N}$. When changing timings, m remains the same. For rational timings, we know that the minimal throughput is reached for a periodic routing whose period is at most m. There are only a finite number of these routings. So there exists a sub-sequence $(x'^{(\phi(k))})$ of timings where the minimal throughput is reached for the same routing. So,

$$d(x_1, x_2 \cdots, x_n) = \lim_{k\to\infty} d(x_1'^{(\phi(k))}, x_2'^{\phi((k))}, \cdots, x_n'^{\phi((k))}).$$

Then we can approach the minimal throughput $d(x_1, x_2 \cdots, x_n)$ by a sequence of rationals that have the same worst routing. This routing will also be the worst routing, that reaches $d(x_1, x_2 \cdots, x_n)$.