Embedded Product-Form Queueing Networks and the Improvement of Performance Bounds for Petri Net Systems*

Javier Campos and Manuel Silva

Departamento de Ingeniería Eléctrica e Informática, Universidad de Zaragoza

María de Luna 3, 50015 Zaragoza, Spain

This paper addresses the computation of upper bounds for the steady-state throughput of stochastic Petri net systems with immediate and generally distributed timed transitions. It is achieved through the use of a kind of decomposition of the whole net system. Results are obtained deeply bridging stochastic Petri net theory to untimed Petri net and queueing network theories. Previous results are improved by considering some embedded product-form queueing networks (generated by the support of some left annullers of the incidence matrix of the net). The obtained results for the case of live and bounded free choice systems are of special interest. In this case, the subnets generated by the minimal left annullers of the incidence matrix always have a topology of product-form closed monoclass queueing networks.

Keywords: Stochastic Petri net systems, throughput bounds, embedded product-form queueing networks.

1 Introduction

The computation of *performance bounds* is a complementary approach to exact and approximate analysis for the evaluation of *stochastic Petri net systems*. Previous results exist concerning *insensitive* throughput bounds of such systems [4, 5, 6], meaning that the bounds, computed by linear programming techniques, only take into account the mean values of service times associated to the firing of transitions and the routing rates associated with transitions in conflict. Therefore, those bounds are insensitive to the form of the probability distribution functions of service times and to the exact conflict resolution policies.

This paper, a fully revised version of [8], gives an improvement for the *throughput upper bound*, looking at some embedded *product-form queueing networks* [3, 17]. Throughput lower bounds can be improved using Petri nets reduction theory ideas [7].

Stochastic Petri net systems are usually introduced as Petri nets with the addition of a stochastic timing interpretation. General distributions for service times and arbitrary policies for the resolution of conflicts can be considered for these stochastic systems. From a different point of view, stochastic Petri net systems can be seen also as classical *queueing networks with the addition of a general synchronization scheme*. Other proposals can be found in the literature for the inclusion of synchronization primitives in queueing systems, but usually they represent *ad hoc* extensions for the modelling of fork-join's or passive resources (e.g., [11, 22]), and many times they are used only as a description language for simulation techniques [22].

The following analogies can be found between queueing networks and stochastic net systems [5]:

- *Queues* are represented by places.
- Timed transitions represent the *service stations*.
- The *number of servers* at each station is related to the maximum reentrance or maximum self-concurrency which is possible for a timed transition (*enabling bound* concept).

*This work was partially supported by the European DEMON ESPRIT BRA 3148, the Spanish PRONTIC's 358/89 and 354/91, and the Aragonese CONAI P IT-6/91.

- The *unconditional routing* (i.e., from one station to another with probability one) is defined with the arc(s) joining a transition to its output place(s), while the *conditional routing* (decisions) is modelled with conflicts of immediate transitions.
- *Customers* distribution in the queueing network and marking in the net system play analogous roles.

The particular *topology* of classical product-form closed monoclass queueing network can be found in a subclass of Petri net systems called *P-components* (i.e., strongly connected state machines: systems not allowing synchronization of tasks). Therefore, these systems with an adequate stochastic interpretation are amenable to be analysed using product-from solutions even with the addition of *self-loop places* with a given number of tokens (for the modelling of limited number of servers at transitions). Larger classes of product-form stochastic Petri net systems can be found in [16, 18].

One of the advantages of the merging of Petri nets and queueing networks theories is that it brings to a general class of synchronized queueing networks all the knowledge about the logical analysis (e.g., deadlock freeness, boundedness, fairness,...) of Petri nets and the performance analysis techniques of queueing networks.

The computation of insensitive upper bounds for the throughput of transitions of stochastic net systems [4, 5, 6] is achieved by considering the subnets generated by P-semiflows (left annullers of the incidence matrix of the net) and assuming that the transitions in the isolated subnets are delay nodes (infinite-server semantics). In other words, the bounds are computed through a *decomposition* of the system.

The key idea for the improvement presented in this paper is to consider the waiting time in queues due to a limited number of servers present at transitions in the steady state (liveness bound). From the stochastic net system we extract subnets with the topology of queueing networks. Since the enabling of transitions in the whole system is more constrained, the throughput of each of the subnets should be an upper bound for the throughput of the entire system. Restricting to the search of embedded queueing networks we look for upper bounds that can be computed in polynomial time using well-known results of product-form monoclass queueing networks theory, such as *balanced throughput upper bounds* [24], *throughput upper bounds hierarchies* [12], or exact *mean value analysis* [21]. At the same time, a new light to the strong relation between queueing networks and stochastic Petri net systems (or synchronized queueing networks) is given.

Embedded queueing networks can be extracted from the entire model looking at the subnets generated by minimal (support) P-semiflows having state machine topology. To be able to freely use the product-form solution theorems [3, 17], we need to impose an additional restriction: the choice between any two transitions in conflict in the subnet should be *free* in the whole net (i.e., both transitions have the same precondition). This fact guarantees that the conflicts among transitions are solved according to a marking independent discrete probability distribution.

For the particular case of free choice nets [15], it is well-known that if they allow a live and bounded marking (the net is said to be structurally live and structurally bounded) they can be fully decomposed into strongly connected subnets with state machine topology (P-components), while the choices in the whole net are obviously free. Some attention will be paid to this subclass of net systems, whose liveness and boundedness can be computed in polynomial time [5, 13].

Basics of Petri nets notation: Let us recall some notation about Petri nets (we refer the reader to [19] for a nice survey). $\mathcal{N} = \langle P, T, Pre, Post \rangle$ is a net with $n = |P|$ places and $m = |T|$ transitions. If the Pre and $Post$ incidence functions take values in $\{0, 1\}$, $\mathcal{N}$ is said ordinary. PRE, $POST$, and $C = POST - PRE$ are $n \times m$ matrices representing the Pre, $Post$, and global incidence functions. Vectors $Y \geq 0$, $Y^T \cdot C = 0$ ($X \geq 0$, $C \cdot X = 0$) represent P-semiflows, also called conservative components (T-semiflows, also called consistent components). The *support* of a P-semiflow (T-semiflow) is defined as $||Y|| = \{p \in P | Y(p) \neq 0\}$ ($||X|| = \{t \in T | X(t) \neq 0\}$). A (P- or T-) semiflow is called *minimal* if it has minimal support. M (M_0) is a marking (initial marking). $\langle \mathcal{N}, M_0 \rangle$ is a net system (or marked net), with $\mathcal{N}$ as underlying net. If $\mathcal{N}$ is an ordinary net such that $\forall t \in T : |{}^\bullet t| = |t^\bullet| = 1$, it is called a *state machine*. A *P-component* is a strongly connected state machine. P-components define minimal P-semiflows, but the reverse is not true in general. If $\mathcal{N}$ is ordinary and $\forall p \in P : |{}^\bullet p| = |p^\bullet| = 1$, the net is a *marked graph*. If $\mathcal{N}$ is ordinary and $\forall p \in P : |p^\bullet| > 1 \Rightarrow {}^\bullet(p^\bullet) = \{p\}$, then $\langle \mathcal{N}, M_0 \rangle$ is a free choice system. Finally, σ represents a firable sequence, while $\vec{\sigma}$ is the firing count vector associated to σ. If M is reachable from M_0 (i.e., $\exists \sigma$ such that $M_0[\sigma\rangle M$), then $M = M_0 + C \cdot \vec{\sigma} \geq 0$ and $\vec{\sigma} \geq 0$.

Assumptions on the stochastic interpretation: In this paper we consider stochastic net systems with general distributions for the service times of transitions. Only *mean values* of these variables are used, denoted s_i for each transition t_i of the net. We assume that a transition t enabled K times in a marking M (i.e., $K = \max\{k | M \geq k PRE[t]\}$) works at speed K times that it would work in the case it was enabled only once (*infinite-server semantics* or *delay node*, with queueing networks terminology). Of course, an infinite-server transition can always be constrained to a "k–server" behaviour by adding one place that is both input and output (self-loop with multiplicity one) for that transition and marking it with k tokens. Other kinds of marking or time dependency of service times are forbidden. We assume that timed transitions may never be in conflict. For the modelling of conflicts we use immediate transitions with the addition of (marking and time independent) routing rates $\mathcal{R}$. In other words, for the subset of immediate transitions $\{t_1, \ldots, t_k\} \subset T$ being in conflict at each reachable marking, we suppose that the constants $r_1, \ldots, r_k \in \mathbb{R}^+$ are explicitly defined in the system interpretation in such a way that when $t_1, \ldots, t_k$ are simultaneously enabled, transition t_i ($i = 1, \ldots, k$) fires with relative rate $r_i / (\sum_{j=1}^{k} r_j)$. In this way, routing is completely decoupled from duration of activities. The only restriction that this decoupling imposes to the system is that *preemption* cannot be modelled with two timed transitions (in conflict) competing for the tokens (i.e., according to the terminology in [1], *race policy* cannot be modelled; our constraint is equivalent to the use of a *preselection policy* for the resolution of conflicts among timed transitions).

The paper is organized as follows. In section 2 we recall the insensitive throughput upper bounds for stochastic net systems derived in previous works. An interpretation in terms of some subnets is given. In section 3, an improvement of the previous bounds is proposed by computing the throughput of queueing networks generated by P-semiflows with limited-server semantics for transitions. Algorithmic aspects of this improvement are studied in section 4. Special attention is devoted to live and bounded free choice systems. Finally, the addition of *implicit places* [10] can have an additional advantage to improve the bounds, since new closed queueing networks embedded in the whole net can be created. Conclusions are summarized in section 5.

2 Insensitive throughput upper bounds and their interpretation

In this section we consider the steady-state behaviour of stochastic net systems (ordinary or not) under *weak ergodicity* assumption for the firing and the marking processes [4]:

$$\overline{M} \stackrel{\text{def}}{=} \lim_{\tau \to \infty} \frac{1}{\tau} \int_0^\tau M_s ds < \vec{\infty}; \qquad \vec{\sigma}^* \stackrel{\text{def}}{=} \lim_{\tau \to \infty} \frac{\vec{\sigma}_\tau}{\tau} < \vec{\infty} \tag{1}$$

where τ represents the time, $\overline{M}_\tau$ and $\vec{\sigma}_\tau$ are the marking at time τ and the firing count vector until this time, respectively, $\overline{M}$ and $\vec{\sigma}^*$ are constants, and *almost everywhere convergence* is assumed (in other words, a set of sample paths with probability one give the same estimation of average values). $\overline{M}$ and $\vec{\sigma}^*$ are called the limit average marking and the limit throughput vector, respectively. Additionally we assume that the *residence time* of a token at each place (time spent by the token within the place) is bounded; therefore,

$$\lim_{n\to\infty} \frac{R_n(p_i)}{n} = 0 \tag{2}$$

where $R_n(p_i)$ is the residence time at p_i of the nth token that arrives to this place. This condition is assured for live and bounded net systems if a *locally fair* consumption of tokens at each place is assumed (for instance, FIFO discipline or random order for the selection of the tokens assure condition (2) while LIFO discipline can lead to an infinite waiting time of a token at a place).

Three of the most significant performance measures for a closed region of a network in the analysis of queueing systems are related by Little's formula: the average number of customers, the output rate (throughput), and the average time spent by a customer within the region. If weak ergodicity of firing and marking processes (1) and condition (2) over the residence times at every place are verified then Little's result can be applied to each place p_i of the net (conditions (ii), (iv), and (ix) of Theorem 2 in [14] hold) as follows:

$$\overline{M}(p_i) = (PRE[p_i] \cdot \vec{\sigma}^*)R(p_i) \tag{3}$$

where $PRE[p_i]$ is the i^{th} row of the pre-incidence matrix of the underlying Petri net, thus $PRE[p_i] \cdot \vec{\sigma}^*$ is the output rate of place p_i, and $R(p_i) = \lim_{n\to\infty} n^{-1} \sum_{k=1}^{n} R_k(p_i)$ is the average residence time at p_i.

In the study of computer systems, Little's law is frequently used when two of the related quantities are known and the third one is needed. This is not exactly the case here. In this case, $R(p_i)$ and $\overline{M}(p_i)$ are unknown, while information about $\vec{\sigma}^*$ can be easily computed only for some (interesting) net subclasses. Let us define the relative firing frequency vector or *vector of visit ratios to transitions* as $\vec{v}^{(i)} \stackrel{\text{def}}{=} \Gamma^{(i)}\vec{\sigma}^*$, where $\Gamma^{(i)} = 1/\sigma^*(t_i)$ represents the *mean interfiring time* of transition t_i (i.e., the inverse of its throughput). Here we consider stochastic net systems whose vector of visit ratios to transitions can be computed in polynomial time from the net structure $\mathcal{N}$ and from the relative frequency of conflict resolutions $\mathcal{R}$ (i.e., the routing rates associated with decisions). As an example, let us consider the net system depicted in figure 1. For this net, the vector of visit ratios for transitions can be computed by solving the following linear system of equations:

$$\begin{array}{l} C \cdot \vec{v}^{(1)} = 0 \ \wedge \ \vec{v}^{(1)} \geq 0; \\ r_1 v^{(1)}(t_2) = r_2 v^{(1)}(t_1); \\ r_3 v^{(1)}(t_4) = r_4 v^{(1)}(t_3); \\ v^{(1)}(t_1) = 1 \end{array} \tag{4}$$

where r_1 and r_2 (r_3 and r_4) are the routing rates used for the resolution of the conflict between t_1 and t_2 (respectively, t_3 and t_4). The first set of equations (implying that $\vec{v}^{(1)}$ is a T-semiflow) are the *flow balance* equations written for each place (input and output flows of tokens are equal, provided that the places are bounded). The second (third) equation is directly derived from the fact that conflict between t_1 and t_2 (respectively, t_3 and t_4) is free and rates r_1 and r_2 (respectively, r_3 and r_4) are fixed. The fourth equation is the normalization for transition t_1.

Equations like (4) have been shown to characterize the vector of visit ratios for important net subclasses such as, for instance, live and bounded *mono-T-semiflow systems* [4] and live and bounded *free choice systems* [5]. Unfortunately, for other subclasses like *simple net systems*, the relative firing frequency vector also depends on the initial marking M_0 and on the service times of transitions [5].

Timed transitions can never be in conflict, so that either all output transitions of a place p_i are immediate or p_i has a unique output transition, say t_i, and t_i is timed. Then, in the later case $\overline{M}(p_i) = (PRE[p_i] \cdot \vec{\sigma}^*)R(p_i) = PRE[p_i, t_i]\sigma^*(t_i)R(p_i) \geq PRE[p_i, t_i]\sigma^*(t_i)s_i = \sum_{j=1}^{m} PRE[p_i, t_j]\sigma^*(t_j)s_j$ (the inequality follows from the fact that the residence time $R(p_i)$ of a token at place p_i with only one output transition is greater than or equal to the service time s_i of that transition). So that $\Gamma^{(i)}\overline{M}(p_i) \geq \sum_{j=1}^{m} PRE[p_i, t_j]\Gamma^{(i)}\sigma^*(t_j)s_j = \sum_{j=1}^{m} PRE[p_i, t_j]v^{(i)}(t_j)s_j$, hence:

$$\Gamma^{(i)}\overline{M} \geq PRE \cdot \vec{D}^{(i)} \tag{5}$$

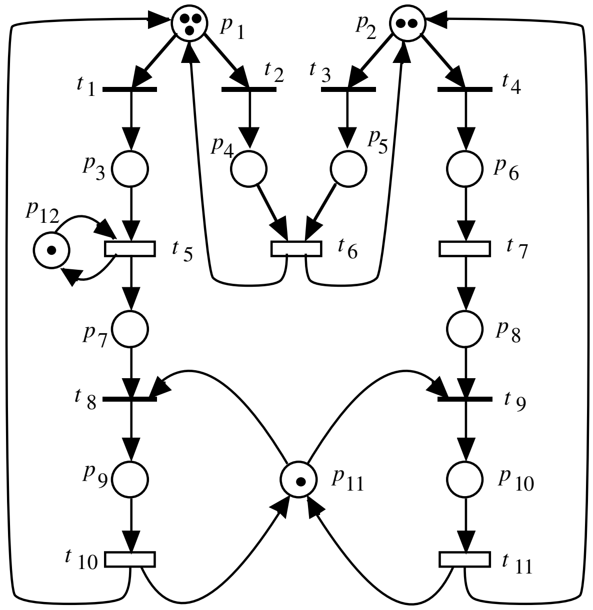


Figure 1: A live and bounded stochastic Petri net system.

where $\vec{D}^{(i)}$ is the vector of *average service demands* of transitions, with components:

$$D^{(i)}(t_j) \stackrel{\text{def}}{=} s_j v^{(i)}(t_j) \tag{6}$$

If all output transitions of place p_i are immediate, then $\overline{M}(p_i) = PRE[p_i]\vec{D}^{(i)} = 0$, thus inequality (5) holds for all place p_i.

P-semiflows Y are non-negative left annullers of the incidence matrix C (i.e., $Y^T \cdot C = 0$), thus $\forall M_0 : Y^T \cdot M = Y^T \cdot M_0$ for all reachable marking M. Therefore, $Y^T \cdot \overline{M} = Y^T \cdot M_0$. Now, premultiplying by Y the relation (5), the following lower bound for the mean interfiring time of a given transition t_i can be derived:

$$\Gamma^{(i)} \geq \max_{Y \in \{P-semiflow\}} \frac{Y^T \cdot PRE \cdot \vec{D}^{(i)}}{Y^T \cdot M_0} \tag{7}$$

The previous lower bound has been formulated in [4] in terms of a fractional programming problem and later, after some considerations, transformed into a linear programming problem [20]:

Property 2.1 [4] *For any live and bounded system, a lower bound for the mean interfiring time $\Gamma^{(i)}$ of transition t_i can be computed by the following linear programming problem:*

$$\begin{array}{rll} \Gamma^{(i)} \geq & \textit{maximum} & Y^T \cdot PRE \cdot \vec{D}^{(i)} \\ & \textit{subject to} & Y^T \cdot C = 0 \\ & & Y^T \cdot M_0 = 1 \\ & & Y \geq 0 \end{array} \tag{LPP1}$$

If the solution of the above problem is unbounded and since it is a lower bound for the mean interfiring time of transition t_i, the non-liveness can be assured (infinite interfiring time). If the visit ratios of all transitions are non-null (i.e., $\vec{v}^{(i)} > 0$), the unboundedness of the above problem implies that a total deadlock is reached by the net system. Anyhow, the unboundedness of (LPP1) means that there exists an unmarked P-semiflow, and obviously the net system is non-live: if $Y_i^T \cdot C = 0$ and $Y_i^T \cdot M_0 = 0$, then $\forall M\ \forall p \in ||Y_i||$: $M(p) = 0$, and the input and output transitions of p are never firable.

The basic advantage of property 2.1 lies in the fact that the *simplex method* for the solution of a linear programming problem has almost linear complexity in practice, even if it has exponential worst case complexity. In any case, algorithms of polynomial worst case complexity can be found in [20].

In order to interpret property 2.1, let us consider again the net system of figure 1. Assuming, for instance, that all routing rates associated with output transitions at conflicts in p_1 and p_2 are equal to one, then the system (4) gives $\vec{v}^{(1)} = \vec{1}$ ($\vec{1}$ is a vector with all entries equal to one). Therefore, according to (6), the vector of average service demands for transitions normalized for t_1 is $\vec{D}^{(1)} = (0,0,0,0,s_5,s_6,s_7,0,0,s_{10},s_{11})^T$, because transitions t_1, t_2, t_3, t_4, t_8, and t_9 are assumed to be immediate.

The *minimal* P-semiflows (minimal support solutions of $Y^T \cdot C = 0, Y \geq 0$) of this net are:

$$\begin{array}{l} Y_1 = (1,0,1,1,0,0,1,0,1,0,0,0)^T \\ Y_2 = (0,1,0,0,1,1,0,1,0,1,0,0)^T \\ Y_3 = (0,0,0,0,0,0,0,0,1,1,1,0)^T \\ Y_4 = (0,0,0,0,0,0,0,0,0,0,0,1)^T \end{array} \tag{8}$$

and the application of (LPP1) gives:

$$\Gamma^{(1)} \geq \max\{ \begin{array}{l} (s_5 + s_6 + s_{10})/3, \\ (s_6 + s_7 + s_{11})/2, \\ s_{10} + s_{11}, \\ s_5 \ \} \end{array} \tag{9}$$

Now, let us consider the P-semiflow decomposed view of the net: the four *subnets generated by* Y_1, Y_2, Y_3, and Y_4 are depicted in isolation in figure 2. Formally speaking, if Y_i is a minimal P-semiflow of a net $\mathcal{N} = \langle P, T, Pre, Post\rangle$, the subnet generated by Y_i is $\mathcal{N}_i = \langle P_i, T_i, Pre_i, Post_i\rangle$ where $P_i = ||Y_i||$ (the support of the P-semiflow), $T_i = {}^\bullet P_i \cup P_i^\bullet$ (i.e., the subset of input or output transitions of places belonging to P_i), and Pre_i, $Post_i$ are the functions Pre, $Post$ restricted to $P_i \times T_i$.

The quantities under the max operator in (9) represent, for this particular case, the mean interfiring time of a transition of each of the four subnets (embedded queueing networks) assuming that all the nodes are delay stations (infinite-server semantics). Therefore, the lower bound for the mean interfiring time of t_1 in the original net system given by (9) is computed *looking at the "slowest subsystem" generated by the P-semiflows*, considered in *isolation* (with delay nodes).

We remark that in this case, since $\vec{v}^{(1)} = \vec{1}$, the throughput of all transitions is equal and it is not necessary to weight the mean interfiring time of transitions computed in isolated subnets in order to get a bound for transition t_1.

In the next section, we improve the previous bound by taking into account that the maximum number of servers that can be available at transitions of the embedded queueing networks can be limited by the number and distribution of tokens in the other subnets.

3 Improvement derived from embedded product-form queueing networks

As stated earlier, the mean interfiring time of transitions of isolated subnets generated by P-semiflows is computed in (LPP1) assuming infinite-server semantics for the involved transitions (i.e., as if they were delay nodes). A more "realistic" computation of the mean interfiring time of transitions of these subnets than that obtained from the analysis in complete isolation is considered now, with finite-server semantics for transitions.

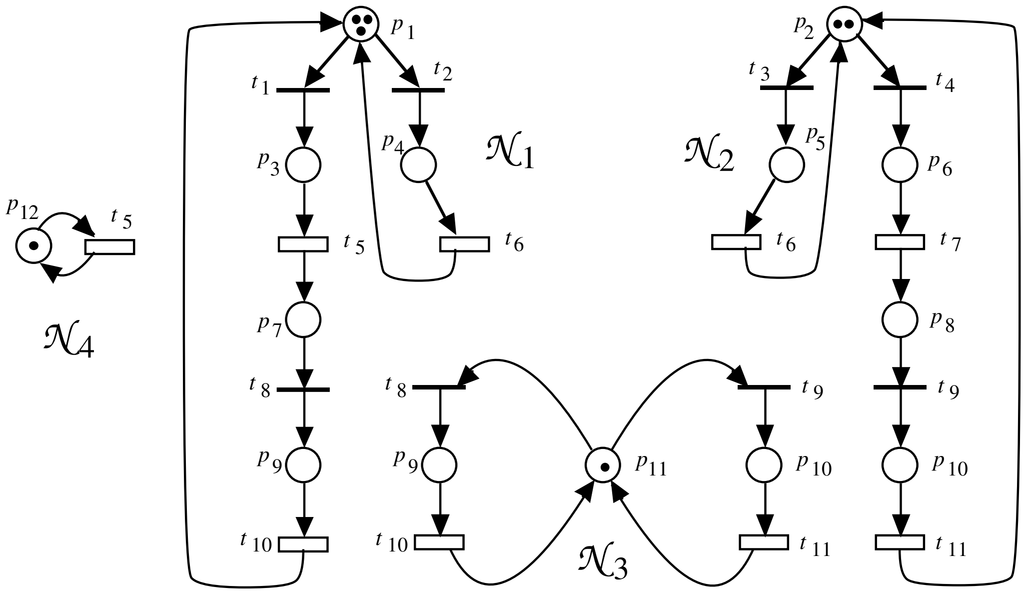


Figure 2: Embedded queueing networks of the net in figure 1 generated by minimal P-semiflows.

The technique we are going to use is based on a decomposition of the original model in subsystems. If we look for embedded product-form closed monoclass queueing networks consisting of P-components eventually with self-loop places, we gain the well-known efficient algorithms that exist for the computation of their throughput. For instance, *mean value analysis algorithm* [21] has $O(A^2B)$ worst case time complexity, where A is the number of customers at the subnet (i.e., the number of tokens at the P-component, $A = Y^T \cdot M_0$, where Y is the P-semiflow that generates the P-component) and B is the number of involved stations (i.e., of transitions, $B = Y^T \cdot PRE \cdot \vec{1}$).

We also remark that other techniques for the computation of throughput upper bounds (instead of exact values) of closed product-form monoclass queueing networks can be used, such as, for instance, *balanced throughput upper bounds* [24] or *throughput upper bounds hierarchies* [12]. Hierarchies of bounds guarantee different levels of accuracy (including the exact solution), by investing the necessary computational effort. This fact immediately provides a *hierarchy of bounds for the mean interfiring time of transitions of Petri net systems.*

Therefore, let us concentrate in the search of such subsystems. How are they structurally characterized? From a topological point of view, they are P-components (i.e., strongly connected state machines and, in particular, ordinary). Timing of transitions can be done with generally distributed services and limited-server semantics. Conditional routing is modelled with decisions among immediate transitions, corresponding to generalized free conflicts in the whole system. In other words, if t_1 and t_2 are in conflict in the considered P-component, they should be in generalized free conflict in the original net: $PRE[t_1] = PRE[t_2]$. The reason for this constraint is that since we are going to consider P-components as product-form closed monoclass queueing networks with limited number of servers at stations (transitions), the throughput of these systems is *sensitive to the conflict resolution policy*, even if the relative firing rates are preserved. Therefore, conflicts in the P-component must be solved with exactly the same *marking independent discrete probability distributions* (defined by means of the routing rates) as

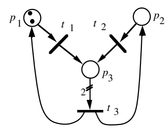


Figure 3: A net system with enabling bound greater than liveness bound for transition t_1.

in the whole net, in order to obtain an optimistic bound for the throughput of the original net system. A counterexample showing that this constraint cannot be relaxed will be presented at the end of this section.

Definition 3.1 *Let $\mathcal{N}$ be a net and $\mathcal{N}_i$ a P-component of $\mathcal{N}$. $\mathcal{N}_i$ is a routing preserving P-component, RP-component, iff for any pair of transitions, t_j and t_k, in conflict in $\mathcal{N}_i$, they are in generalized free (equal) conflict in the whole net, $\mathcal{N}$: $PRE[t_j] = PRE[t_k]$.*

We remark that checking if a subnet generated by a P-semiflow is an RP-component has obviously linear time complexity on the number of transitions of the net. For the example of figure 1, $\mathcal{N}_1$, $\mathcal{N}_2$, and $\mathcal{N}_4$ are RP-components, while $\mathcal{N}_3$ is not ($PRE[t_8] \neq PRE[t_9]$).

The performance of a net with infinite-server semantics of transitions depends on the maximum number of servers at stations: the maximum degree of enabling of the transitions, the *enabling bound.*

Definition 3.2 [4] *Let $\langle \mathcal{N}, M_0 \rangle$ be a net system. The enabling bound of a given transition t of $\mathcal{N}$ is $E(t) \stackrel{\text{def}}{=} \max\{k \mid \exists M \in R(\mathcal{N}, M_0) : M \geq kPRE[t]\}$.*

In particular, the steady-state performance does depend on the maximum degree of enabling of transitions in steady-state, which can be different from the maximum degree of enabling of a transition during all its evolution from the initial marking. Therefore, we also recall the concept of *liveness bound*, which allows to generalize the classical concept of liveness of a transition:

Definition 3.3 [4] *Let $\langle \mathcal{N}, M_0 \rangle$ be a net system. The liveness bound of a given transition t of $\mathcal{N}$ is $L(t) \stackrel{\text{def}}{=} \max\{k \mid \forall M' \in R(\mathcal{N}, M_0), \exists M \in R(\mathcal{N}, M') : M \geq kPRE[t]\}$.*

An example of a live and bounded net system with enabling bound of a transition (t_1) greater than its liveness bound is depicted in figure 3: $E(t_1) = 2 > 1 = L(t_1)$.

The definitions above refer to behavioural properties. Since we are looking for computational techniques at the structural level, we also recall the structural counterpart of the first concept.

Definition 3.4 [4] *Let $\langle \mathcal{N}, M_0 \rangle$ be a net system. The structural enabling bound of a given transition t of $\mathcal{N}$ is:*

$$SE(t) \stackrel{\text{def}}{=} \begin{array}{ll} \textit{maximum} & k \\ \textit{subject to} & M_0 + C \cdot \vec{\sigma} \geq kPRE[t] \\ & \vec{\sigma} \geq 0 \end{array} \tag{LPP2}$$

Note that the definition of structural enabling bound reduces to the formulation of a linear programming problem. The following result related to the above concepts has been obtained in [4]:

Property 3.1 [4] *Let $\langle \mathcal{N}, M_0 \rangle$ be a net system, then for all transition t of $\mathcal{N}$, $SE(t) \geq E(t) \geq L(t)$.*

The interest of the above property lies in the fact that for those net systems whose exact liveness bounds of transitions cannot be efficiently computed, upper bounds (i.e., optimistic values) can be always obtained by solving the linear programming problems (LPP2), i.e., by computing the structural enabling bounds.

Going back to the semantics of transitions, the number of servers at each transition t of a given net system in steady state is limited to its corresponding liveness bound $L(t)$ (or to its structural enabling bound which can always be computed in an efficient manner), because this bound is the *maximum reentrance* (or maximum self-concurrency) that the net structure and the marking allow for the transition.

The next property states that the mean interfiring time of a transition t_i of an isolated RP-component with $L(t)$–server semantics for each transition t is a lower bound for the mean interfiring time of the same transition t_i computed in the whole net system.

Property 3.2 *Let Y be a minimal P-semiflow of a Petri net system that generates an RP-component. Let $\Gamma^{(i)}$ be the exact mean interfiring time of t_i in the whole net system and $\Gamma^{(i)}_{Y_L}$ be the exact mean interfiring time of t_i in the isolated RP-component generated by Y, with $L(t)$–server semantics for each involved transition t. Then, $\Gamma^{(i)} \geq \Gamma^{(i)}_{Y_L}$, provided*

(i) the transitions of the RP-component are FIFO in the sense that the nth service completion of each transition correspond to the nth service start of that transition; or

(ii) only one of the transitions of the RP-component generated by Y, say t, synchronizes with the rest of the net system in the sense that there exists an input place to t which does not belong to the RP-component and is not a self-loop place of t.

Proof: Assume that the transitions of the RP-component are FIFO. In [2, Theorem 7.2], it is shown that for every live FIFO stochastic net system whose conflicts are free (this is true, in particular for an RP-component), the mean interfiring time of each transition does not decrease if the service times of transitions increase (or they are preserved for some transitions). Then, let us consider the RP-component generated by Y, with $L(t)$–server semantics for each involved transition t, embedded in the whole net. The effect that the rest of the net system has on the behaviour of the RP-component can be seen as an increases of the service time of the transitions that synchronize the RP-component with the rest of the system. Therefore, according to [2, Theorem 7.2], the mean interfiring time of a transition if the RP-component is considered in isolation, with $L(t)$–server semantics for each involved transition t, is less than or equal to the mean interfiring time of the same transition if the RP-component is synchronized with the rest of the net system.

Assume now that only one of the transitions of the RP-component generated by Y synchronizes with the rest of the net system. The isolated RP-component, with $L(t)$–server semantics for each involved transition t, can be seen as a product-form monoclass queueing network whose service rates are non-decreasing functions in the load. Then, according to [23, Corollary 3.1], if the service time of one of the transitions increases (due to the effect of synchronization with the rest of the system), its mean interfiring time does not decrease. ∎

We remark that condition (i) holds, in particular, if each transition in the RP-component has a single server (i.e., liveness bound equal to one).

Conjecture. *Even though we are not able to present a formal proof of the previous property without the assumptions (i) or (ii), we conjecture that it holds in general.*

In the next property, the mean interfiring times of a transition in an RP-component with limited and with infinite number of servers at transitions are related.

Property 3.3 *Let Y be a minimal P-semiflow (feasible solution of the problem (LPP1)) of a Petri net system that generates an RP-component. Let $\Gamma^{(i)}_{Y_L}$ be the exact mean interfiring time of t_i in the isolated*

RP-component generated by Y, with $L(t)$–server semantics for each involved transition t, and $\Gamma^{(i)}_{Y_\infty}$ be the value of the objective function of (LPP1) corresponding to Y. Then, $\Gamma^{(i)}_{Y_L} \geq \Gamma^{(i)}_{Y_\infty}$.

Proof: The isolated RP-component generated by Y, with $L(t)$–server semantics for each involved transition t, is a product-form monoclass queueing network whose service rates are non-decreasing functions in the load. If all transitions are now substituted by infinite-server nodes, the RP-component is still a product-form monoclass queueing network whose service rates are non-decreasing functions in the load. Moreover, the service rate functions have been increased for some values of the load. Therefore, the result follows from [23, Corollary 3.1]. ■

Properties 3.2 and 3.3 state that the knowledge of the liveness bound of transitions for a given net can allow to improve the throughput upper bound computed in property 2.1 by investing an additional computational effort. We summarize now an interpretation of this improvement from a queueing theory point of view:

Interpretation. *Both the bound presented in section 2 and the presented in this section are based on the computation of the mean interfiring time of transitions of subnets generated by P-semiflows considered in isolation. In the first case, since infinite-server semantics is considered for the isolated subnet, the real (unknown) residence time at places is lowerly bounded by the service time of transitions, but waiting time due to synchronizations is not considered at all. Now, the bound for the residence time at places is improved taking into account not only the service time but also a part of the queueing time due to synchronizations: the time in queue when $L(t)$ servers is the maximum available at each transition t. Anyhow, only bounds (i.e., not exact results) are computed in general because synchronizations among components are only partially represented (the number of servers has been reduced, but they are always available for working: their "setup" times after service are neglected).*

As an example, let us consider once more the net system depicted in figure 1. The structural enabling bounds of all transitions can be computed in polynomial time by solving the corresponding problems (LPP2). In fact, we only require to compute these bounds for the timed (non-immediate) transitions. In this case $L(t_i) = SE(t_i), i = 1, \ldots, 11$. They are $L(t_6) = L(t_7) = 2$ and $L(t_5) = L(t_{10}) = L(t_{11}) = 1$. Then, the embedded queueing network generated by the P-semiflow Y_1, considered with $L(t)$–server semantics for each of the timed transitions t, is the one depicted in figure 4 (compare with $\mathcal{N}_1$ in figure 2). Assume that exponentially distributed service times are associated to timed transitions, with means $s_5 = 3$, $s_6 = 4$, $s_7 = 1$, $s_{10} = 2$, and $s_{11} = 1$, while transitions t_1, t_2, t_3, t_4, t_8, and t_9 are immediate. The exact mean interfiring time of transitions of the net system in figure 4 can be efficiently computed using, for instance, mean value analysis algorithm [21]. The exact mean interfiring time of t_1 in the whole net system, and the upper bounds obtained from properties 2.1 and 3.2 are the following:

$$\Gamma^{(1)} = 4.89; \quad \Gamma^{(1)}_{Y_L} = 4.19; \quad \Gamma^{(1)}_{Y_\infty} = 3.00 \tag{10}$$

The reader is noticed that the conjecture $\Gamma^{(1)} \geq \Gamma^{(1)}_{Y_L}$ holds even if conditions (i) and (ii) of property 3.2 are not true. In this example, the relative error of the bound ($\Gamma^{(1)}_{Y_\infty}$) has been reduced from 38% to 14% (with the bound $\Gamma^{(1)}_{Y_L}$).

As stated before, in order to be applicable, property 3.2 demands the conflicts present in the considered P-component to be free in the original net. A counterexample showing that this constraint cannot be relaxed is depicted in figure 5. A minimal P-semiflow of the net in figure 5.a has the support $\{p_1, p_3, p_4, p_5, p_7\}$. The subnet generated by this support is a P-component. In figure 5.b is depicted the subsystem with the addition of self-loops to transitions t_1 and t_3, modelling the limited-server semantics imposed by the rest of the net system (obtained by computing the liveness bounds of transitions). In the original net system, the vector of visit ratios can be efficiently computed. If the routing rates defining the resolution of free conflict between t_4 and t_5 are equal, the visit ratio for transition t_2 is twice that of

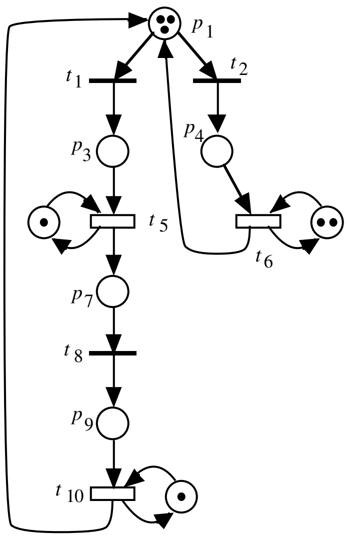


Figure 4: Subnet of the net in figure 1 generated by Y_1, with $L(t)$–server semantics for each involved transition t.

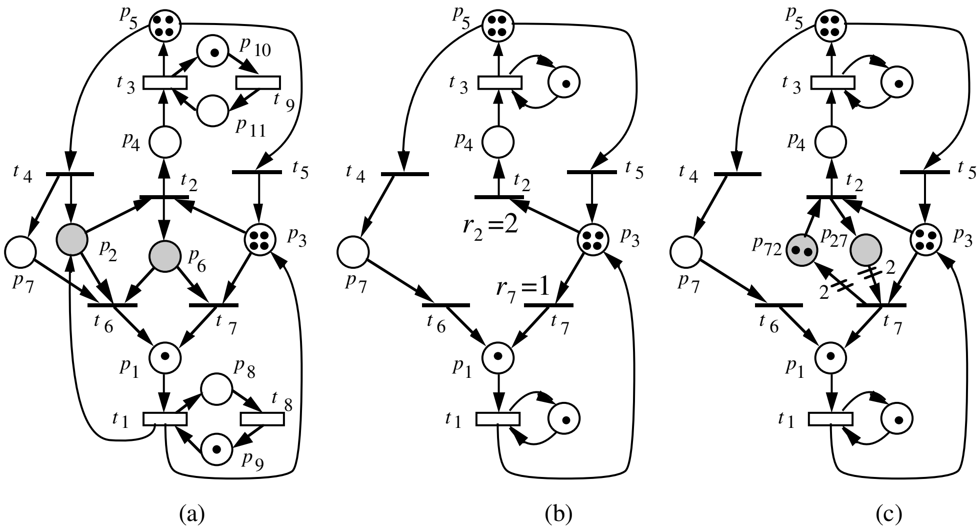


Figure 5: The improvement of the bound cannot be applied.

t_7. Then, the routing rates $r_2 = 2$ and $r_7 = 1$ could be used to solve the conflict between t_2 and t_7 in the subsystem depicted in figure 5.b. Assume that the service times of timed transitions are exponentially distributed with means $s_1 = s_3 = 1$ and $s_8 = s_9 = 0.1$. The exact mean interfiring time of transition t_1 in the original system (figure 5.a) is $\Gamma^{(1)} = 1.179$. The mean interfiring time of the same transition computed in the subsystem in figure 5.b is $\Gamma^{(1)'} = 1.209$, and obviously it is not a lower bound for the mean interfiring time of the original system. In figure 5.c, it is depicted the same subsystem than in figure 5.b, but now the conflict between t_2 and t_7 is solved with another policy (a deterministic policy modelled with a *regulation circuit* formed with p_{27} and p_{72}) preserving the same relative visit ratios for both transitions. The mean interfiring time of t_1 in figure 5.c is $\Gamma^{(1)''} = 1.182$, which is different from $\Gamma^{(1)'}$ and neither is a lower bound for $\Gamma^{(1)}$.

The problem with the net of figure 5.a is that the P-component that we consider has a conflict between t_2 and t_7 that comes from a more extensive non-free conflict in the whole net (involving also t_6): $PRE[t_2] \neq PRE[t_7]$. And the resolution of this global conflict is not precisely reflected in the subnet (neither in figure 5.b nor in figure 5.c). For the particular case at hand (figure 5.a), places p_2 and p_6 implement a conflict resolution policy that obviously differs from a marking independent discrete probability distribution, as used in figure 5.b to obtain a queueing network with product-form solution.

4 Computing the throughput upper bound

In this section, we study algorithmic aspects of the application of the improvement presented in previous paragraphs. First, the general case is considered (Petri net systems without restrictions on their structure). After that, the interesting particular case of free choice systems is studied. Finally, additional improvements such as those obtained from the use of *implicit places* are mentioned.

4.1 General case

Stating properties 3.2 and 3.3 for Y^*, an optimum solution of (LPP1), the bound computed in property 2.1 can be eventually improved as follows:

Corollary 4.1 *An improvement of the throughput upper bound computed in property 2.1 can be obtained computing the value $\Gamma^{(i)}_{Y^*_L}$ of property 3.2 for an optimum solution Y^* of the problem (LPP1) that generates an RP-component verifying assumptions (i) or (ii) of property 3.2.*

As an example of the above improvement, let us consider the net system in figure 6. Assume that routing rates are equal to one for t_1, t_2, and t_3, and that t_7, t_8, t_9, t_{10}, t_{11}, t_{12} have exponentially distributed service times with mean values $s_7 = s_8 = s_9 = 10$, $s_{10} = s_{11} = s_{12} = 1$. The minimal P-semiflows of the net are:

$$\begin{array}{l} Y_1 = (1,1,1,1,1,1,1,0,0,0,0,0,0)^T \\ Y_2 = (0,0,0,0,1,0,0,1,0,0,1,0,0)^T \\ Y_3 = (0,0,0,0,0,1,0,0,1,0,0,1,0)^T \\ Y_4 = (0,0,0,0,0,0,1,0,0,1,0,0,1)^T \end{array} \tag{11}$$

All of them generate RP-components verifying condition (i) of property 3.2 (the liveness bound of all the timed transitions is one). Then, if the initial marking of p_{11}, p_{12}, and p_{13} is one token, and the initial marking of p_1 is N tokens, the lower bound for the mean interfiring time derived from (LPP1) is $\Gamma^{(1)}_{(\mathrm{LPP1})} = \max\{30/N, 11, 11, 11\}$. For $N = 1$, the previous bound, obtained from Y_1, gives the value 30, while the exact mean interfiring time is 31.06. For $N = 2$, the bound is 15 and it is derived also from Y_1 (mean interfiring time of the RP-component generated by Y_1, considered in isolation with infinite-server semantics for transitions). This bound does not take into account the queueing time at places due to synchronizations (t_4, t_5, and t_6), and the exact mean interfiring time of t_1 is $\Gamma^{(1)} = 21.05$. For larger values of N, the bound obtained from (LPP1) is equal to 11 (and is given by P-semiflows Y_2, Y_3 and Y_4).

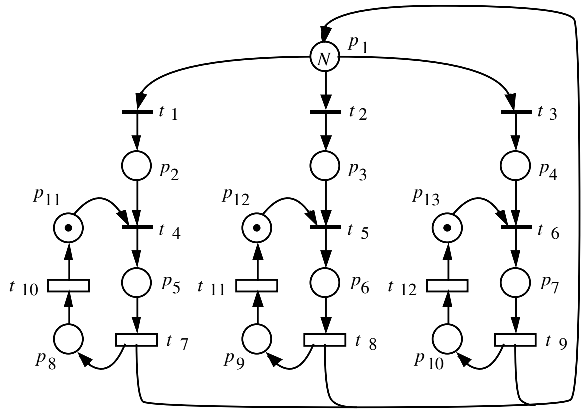


Figure 6: A live and bounded free choice system.

N	$\Gamma^{(1)}$	$\Gamma^{(1)}_{(Y_1)_L}$	$\Gamma^{(1)}_{(\mathrm{LPP1})}$
1	31.06	30	30
2	21.05	20	15
3	17.71	16.67	11
4	16.03	15	11
5	15.03	14	11
10	13.02	12	11
15	12.35	11.34	11

Table 1: Exact mean interfiring time of t_1 (for exponential distributions), bounds obtained using (LPP1), and the improvements derived from property 3.2, for different initial markings of p_1 in the net of figure 6.

This bound can be improved if the RP-component generated by Y_1 is considered with liveness bounds of transitions t_7, t_8, and t_9 reduced to one (which is the liveness bound of these transitions in the whole system). The results obtained for different values of N are collected in table 1. Bounds derived from the exact values of mean interfiring time of t_1 in the RP-component generated by Y_1 with limited number of servers were computed using the mean value analysis algorithm [21]. Exact computation takes several minutes of the CPU of a *SPARC Workstation* (using *GreatSPN* [9]) while the computation of bounds takes only a few seconds.

Taking into account that the number of optimum solutions of (LPP1) (giving the same value of the objective function) that generate RP-components can be theoretically exponential on the net size (in practice this is very unlikely even for well balanced systems!), a first question to be answered is: Which RP-components should be considered in order to obtain a greater improvement with the application of corollary 4.1?

We now present an algorithm for the computation of an improvement of bounds given by problem (LPP1), based on a possible heuristic for the selection of some optimum solutions of (LPP1). The heuristic gives the possibility of selecting up to an arbitrary number K of optimum solutions of (LPP1) that generate RP-components. The way of selecting only optimum solutions among all feasible solutions of that linear programming problem consists of considering the following constraints:

$$\begin{array}{l} Y^T \cdot PRE \cdot \vec{D}^{(i)} = \Gamma_{PS}^{(i)} \\ Y^T \cdot C = 0;\ Y^T \cdot M_0 = 1;\ Y \geq 0 \end{array} \tag{12}$$

where $\Gamma_{PS}^{(i)}$ is the optimum value of (LPP1), and must be computed before.

Now, it is easy to understand that, among the above optimum solutions, those with lower liveness bounds for involved timed transitions should probably give slower embedded queueing networks. This is because, if only a few servers exist at a given transition, the waiting time of tokens in the input places will be larger. A "natural" way to select those RP-components with expected smaller number of servers at involved transitions is to solve a linear programming problem with expression (12) as constraints, and with the same objective function than in problem (LPP1) but modifying the vector $\vec{D}^{(i)}$, dividing the mean service time s_j of each transition t_j by its corresponding liveness bound $L(t_j)$. Intermediate situations can be considered dividing each s_j by a quantity ranging from $1+\delta$ (with $\delta > 0$) to $L(t_j)$.

P-components define minimal P-semiflows. Thus, it is very important that optimal solutions of (LPP1) are minimal P-semiflows. This is particularly true if (LPP1) is solved using any (revised) *simplex* method [20].

An algorithm for the previously argued heuristic can be as follows:

Step 0. Compute $L(t)$ for each timed t (or, in general, an upper bound for it, $SE(t)$), solving the problem (LPP2).

Step 1. Solve the problem (LPP1). Let $\Gamma_{PS}^{(i)}$ be its optimum value. Let $\mathcal{Y} := \emptyset$; $\Gamma^{(i)} := \Gamma_{PS}^{(i)}$.

Step 2. For $k := 1$ **to** K **do**

(2.1) Solve the linear programming problem (LPP$_{(k)}$):

$$\begin{array}{rll} \Gamma_k^{(i)} = & \text{maximum} & Y^T \cdot PRE \cdot \vec{G}_k^{(i)} \\ & \text{subject to} & Y^T \cdot PRE \cdot \vec{D}^{(i)} = \Gamma_{PS}^{(i)} \\ & & Y^T \cdot C = 0;\ Y^T \cdot M_0 = 1 \\ & & Y \geq 0 \end{array} \tag{13}$$

where $\vec{G}_k^{(i)}$ is a vector with dimension equal to the number of transitions and components

$$G_k^{(i)}(t_j) = \frac{s_j\ v^{(i)}(t_j)}{1 + k(L(t_j) - 1)/K} \tag{14}$$

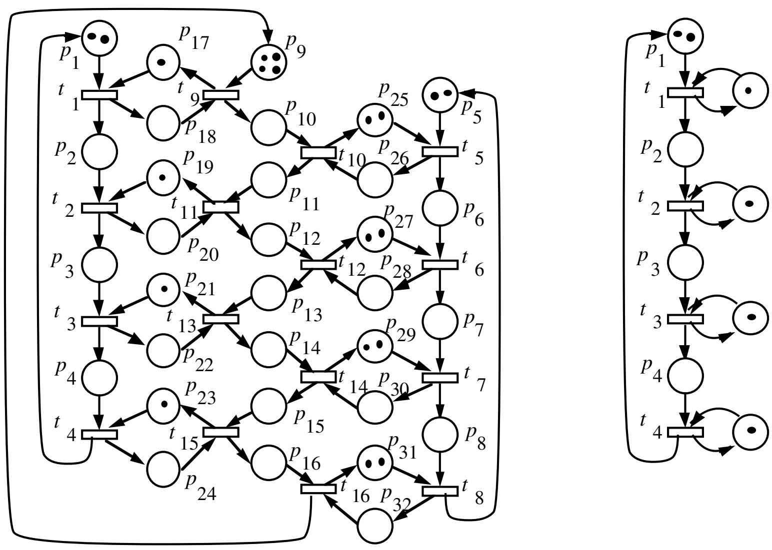
Let Y_k be one optimum solution of (LPP$_{(k)}$).

(2.2) If ($Y_k \notin \mathcal{Y}$) and ($\mathcal{N}_{Y_k}$ is an RP-component verifying (i) or (ii) of property 3.2)

then compute the mean interfiring time $\Gamma_k^{(i)}$ of $\mathcal{N}_{Y_k}$ assuming $L(t)$–server semantics for each timed transition t, using (for instance) the mean value analysis algorithm. $\mathcal{Y} := \mathcal{Y} \cup \{Y_k\}$; $\Gamma^{(i)} := \max\{\Gamma^{(i)}, \Gamma_k^{(i)}\}$.

Output. $\Gamma^{(i)}$ is a lower bound for the mean interfiring time of t_i.

We remark that if our conjecture about the generalization of property 3.2 holds, the test of conditions (i) and (ii) in Step 2.2 can be avoided. In the sequel, we consider also RP-components that verify the statement $\Gamma^{(i)} \geq \Gamma_{Y_L}^{(i)}$ of property 3.2 even if conditions (i) and (ii) of that property do not hold. As an example, let us consider the net system depicted in figure 7.a. In fact, we have selected a marked graph for simplicity: in this case the subnets generated by minimal P-semiflows are *elementary circuits*. From a queueing theory perspective, the example does not lose generality because even though the embedded



(a) (b)

Figure 7: An example of application of the heuristic algorithm: condition (i) of property 3.2 holds for the subsystem (b).

queueing networks have visit ratios equal to one for all transitions, arbitrary average service demands can be obtained changing the associated mean service times. Assume that service times of transitions are exponential with means $s_1 = s_2 = s_3 = s_4 = s_5 = s_6 = s_7 = s_8 = 2$ and $s_9 = s_{10} = s_{11} = s_{12} = s_{13} = s_{14} = s_{15} = s_{16} = 1$. Then, the application of (LPP1) gives $\Gamma^{(1)} \geq \frac{8}{2} = 4$. This optimum value is obtained for two different feasible solutions (circuits), generated by the P-semiflows:

$$
\begin{array}{l}
Y_1 = (1,1,1,1,0,0,0,0,0,\ldots,0)^T \\
Y_2 = (0,0,0,0,1,1,1,1,0,\ldots,0)^T
\end{array}
\tag{15}
$$

The application of the above algorithm for $K = 1$ selects the first one, because the liveness bounds of the involved transitions in Y_1 are equal to one (less than those in Y_2): (LPP$_{(1)}$) gives the optimum value equal to 4 for Y_1, while the other feasible solution Y_2 gives only 2 (the service times of the involved transitions in Y_2 are divided by their corresponding liveness bounds, which are all equal to 2). Then, the queueing network generated by Y_1, with liveness bounds of transitions equal to one, must be solved (figure 7.b). Mean value analysis applied to this network gives the following bound for the mean interfiring time of t_1: $\Gamma^{(1)} \geq 5$. While the exact mean interfiring time in the whole net (obtained solving the embedded continuous time Markov chain, with 10515 states) is $\Gamma^{(1)} = 5.87$. The same analysis applied for the network generated by Y_2 gives $\Gamma^{(1)} \geq 4$; in this case no improvement is obtained because the number of servers at each transition allowed by the rest of the net is equal to the number of customers within the subnetwork.

Let us remark that, in the particular case in which the liveness bounds of all timed transitions were equal ($L(t) = L$, for all timed transition t), the problems (LPP$_{(k)}$) would not select any "better" solution. All the feasible solutions would give the same value for the objective function of each (LPP$_{(k)}$): $\Gamma_{PS}^{(i)}/(1+k(L-1)/K)$. Fortunately, this case is easy to detect (at Step 0), and there exists an alternative heuristic for the selection of an optimum solution of (LPP1). Step 2 in previous algorithm must be substituted by the following:

Step 2bis. **If** $L(t) = L$, for all timed transition t

then Let Y_1 be an optimum solution of

$$
\begin{array}{ll}
\text{maximize} & Y^T \cdot M_0 \\
\text{subject to} & Y^T \cdot (PRE \cdot \vec{D}^{(i)} - \Gamma_{PS}^{(i)} M_0) = 0 \\
 & Y^T \cdot C = 0; Y \geq 0
\end{array}
\tag{LPP3}
$$

If $\mathcal{N}_{Y_1}$ is an RP-component

then compute the mean interfiring time $\Gamma_1^{(i)}$ of $\mathcal{N}_{Y_1}$ assuming $L(t)$–server semantics for each timed transition t, using (for instance) the mean value analysis algorithm. $\Gamma^{(i)} := \Gamma_1^{(i)}$.

else Execute Step 2 of the previous algorithm.

That is, since all RP-components include transitions with the same maximum number of servers, we can expect that likely the slowest RP-component is the one with the maximum number of tokens, and thus with maximum residence time at places, waiting for an available server. As in Step 2.2 of the first algorithm, it is necessary to check if the obtained solution generates an RP-component.

As an example, look at the net system depicted in figure 8.a. Assume exponential service time distributions with means $s_1 = s_2 = s_3 = s_4 = 2$ and $s_5 = s_6 = s_7 = s_8 = 4$. The application of (LPP1) gives $\Gamma^{(1)} \geq \max\left\{\frac{8}{1}, \frac{16}{2}, \frac{6}{1}\right\} = 8$. The optimum value is reached with two different feasible solutions, the circuits generated by:

$$
\begin{array}{l}
Y_1 = (1,1,1,1,0,0,0,0)^T \\
Y_2 = (0,0,0,0,1,1,1,1)^T
\end{array}
\tag{16}
$$

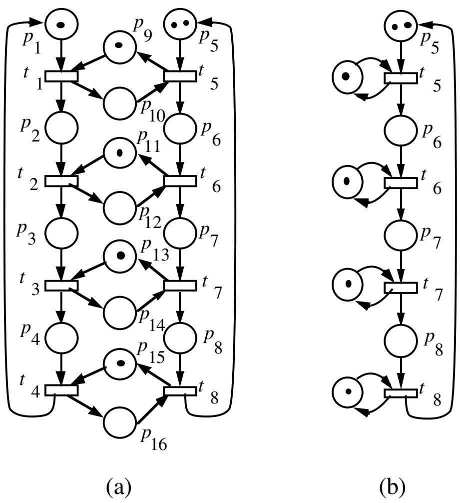


Figure 8: Application of the heuristic when the liveness bounds of all transitions are equal.

The liveness bound of all transitions is equal to one (thus any minimal P-semiflow generates an RP-component verifying condition (i) of property 3.2). Therefore the problem (LPP$_{(1)}$) does not help to improve the bound derived from (LPP1). However, the application of problem (LPP3) selects the circuit generated by Y_2, because it contains a greater number of tokens than the circuit generated by Y_1. Then, the application of the mean value analysis algorithm to the network generated by Y_2, with liveness bound of transitions equal to one (figure 8.b) gives the bound $\Gamma^{(1)} \geq 10$. And the exact mean interfiring time of transition t_1 in the original net is $\Gamma^{(1)} = 13.14$.

4.2 Free choice case

In Step 2.2 of the algorithm presented in previous section, it is necessary to check: 1) if a given minimal P-semiflow generates a P-component, and 2) if all conflicts in this P-component are free in the whole system (as we remarked before, both tests can be done in linear time on the number of transitions). If we consider a net subclass where minimal P-semiflows always generate P-components, then the first part of the test can be avoided. Additionally, if the structure of the whole system assures that all conflicts in the P-components are free in the original system, then the other part of the test can be also avoided.

For example, for the net system depicted in figure 1 (a live and bounded free choice system provided with a mutual exclusion semaphore, p_{11}) all minimal P-semiflows generate P-components. However, it is not true that all conflicts in the P-components are free in the whole system: the P-component labelled $\mathcal{N}_3$ in figure 2 does not verify this condition. Therefore, the test for the condition of being RP-component cannot be omitted for this net system.

Let us consider once more the system in figure 6. In this case the test for being RP-component in Step 2.2 of the algorithm can be completely omitted because all minimal P-semiflows of the net generate RP-components. In fact, this system belongs to an interesting subclass of systems that always verify that property: live and bounded free choice systems. Moreover, this subclass of systems can be characterized in polynomial time, as stated in the next property.

Property 4.2 [5, 13] *A net system is live and bounded free choice iff the following statements, that can* *be checked in polynomial time, are verified:*

1. *It is free choice:* $\forall p \in P, |p^\bullet| > 1 \Rightarrow {}^\bullet(p^\bullet) = \{p\}$.
2. *It is conservative:* $\exists Y > 0, Y^T \cdot C = 0$.
3. *It is consistent:* $\exists X > 0, C \cdot X = 0$.
4. *It verifies the following rank equation:* $rank(C) = m - 1 - a - n$, *where* m *is the number of transitions,* $n =$ *the number of places, and* a *the number of arcs of the Pre incidence function.*
5. *All the P-semiflows are marked:* $\nexists Y \geq 0, Y^T \cdot C = 0$ *such that* $Y^T \cdot M_0 = 0$.

We remark that the fifth statement in the previous property is implicitly checked when solving the problem (LPP1) for the computation of a throughput upper bound.

Now, the interesting property of live and bounded free choice systems is that their minimal P-semiflows always generate P-components (see, for example, [13]). Because in a free choice net all choices are free, the P-components are RP-components. The reverse is always true, i.e., RP-components define minimal P-semiflows. Therefore:

Property 4.3 *Let* $\langle \mathcal{N}, M_0 \rangle$ *be a live and bounded free choice system.* Y *is a minimal P-semiflow of* $\mathcal{N}$ *iff* $\mathcal{N}_Y$*, the subnet generated by* Y*, is an RP-component.*

As a conclusion of the above property, the algorithm presented in the previous section can be applied for live and bounded free choice systems without executing the RP-component test in Step 2.2.

Now we just recall a result which provides an efficient method for the computation of liveness bounds of transitions (Step 0 in the algorithm) for the case of live and bounded free choice systems: the liveness bounds of transitions (actual number of servers needed at transitions in steady-state) of live and bounded free choice systems can be obtained by solving the problem (LPP2).

Property 4.4 [5] *Let* $\langle \mathcal{N}, M_0 \rangle$ *be a live and bounded free choice system. Then, for all transition* t *of* $\mathcal{N}$, $SE(t) = E(t) = L(t)$.

In the next section some additional improvements and refinements are summarized.

4.3 Additional improvements

In [6], an improvement of the insensitive throughput upper bounds for general stochastic net systems is presented by considering *implicit places.* Place p is implicit in a net system iff its elimination preserves the firing sequences of the net system; in other words, p is never the unique place that prevents the firing of a transition [10].

The addition of implicit places generates new minimal P-semiflows. Therefore, the space of feasible solutions of the problem (LPP1) is increased and the insensitive bound can be eventually improved.

As an example consider the net system in figure 9.a. The mean service times associated with transitions t_3 and t_4 are $s_3 = s_4 = 5$. Transitions t_1, t_2, and t_5 are immediate. Assuming that the conflict at p_1 is solved with equal probability for t_1 and t_2, the vectors of visit ratios and average service demands to transitions are $\vec{v}^{(1)} = (1,1,1,1,2)^T$ and $\vec{D}^{(1)} = (0,0,5,5,0)^T$. The minimal P-semiflows are $Y_1 = (1,1,0,0,1)^T$ and $Y_2 = (1,0,1,1,0)^T$. The problem (LPP1) gives $\Gamma^{(1)} \geq \max\left\{\frac{5}{2}, \frac{5}{2}\right\} = 2.5$.

Now, if the place p_6 is added to the net with initial marking equal to one in order to be implicit (see figure 9.b), the following P-semiflow is generated: $Y_3 = (1,1,1,0,0,1)^T$. The application of (LPP1) yields: $\Gamma^{(1)} \geq \max\left\{\frac{5}{2}, \frac{5}{2}, \frac{10}{3}\right\} = 3.3$.

The same technique can be used for the improvement of the throughput upper bounds presented in this paper. The new P-semiflows generated after the addition of implicit places can lead to new "slower" RP-components. For example, for the net system of figure 9, the addition of the implicit place p_6

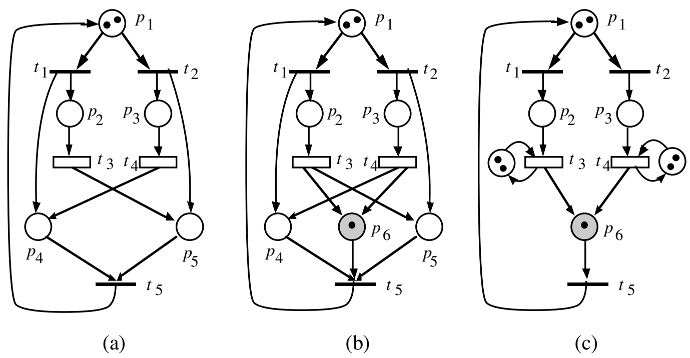


Figure 9: The addition of implicit places improves the bound.

(figure 9.b) leads to a new RP-component (subnet generated by Y_3). The exact solution of this embedded queueing network (the RP-component with liveness bounds of t_3 and t_4 equal to 2, see figure 9.c) gives the bound: $\Gamma^{(1)} \geq 3.75$. The exact mean interfiring time for exponentially distributed service times is $\Gamma^{(1)} = 4$. Therefore, the relative error has been reduced from 37.5% in the first bound ($\Gamma^{(1)} \geq 2.5$) to 6.25% in the last one ($\Gamma^{(1)} \geq 3.75$).

Another additional improvement of the throughput upper bounds can be obtained with a small generalization of the algorithm of the previous section. The idea is the following: in Step 2 of that algorithm, some optimal solutions of the problem (LPP1) are selected, in order to consider those that generate RP-components. But, it may happen that the slowest queueing network embedded in a given stochastic Petri net system is generated by a P-semiflow which is *not optimum* for the problem (LPP1).

The selection of RP-components generated by non-optimal P-semiflows of (LPP1) is possible if Step 2.1 of the algorithm is modified in order to allow the consideration of near-optimal solutions: this can be done by substituting the constraint $Y^T \cdot PRE \cdot \vec{D}^{(i)} = \Gamma_{PS}^{(i)}$ with the weaker constraint $Y^T \cdot PRE \cdot \vec{D}^{(i)} \geq \alpha \Gamma_{PS}^{(i)}$, with a parameter α selected in $(0, 1]$. In this way, additional improvements can be obtained, of course at the expense of an additional computational cost (considering more embedded queueing networks). A classical (computational) cost-quality (of the bound) *tradeoff* appears.

For example, for the net system in figure 6, if the initial marking of place p_1 is $N \geq 3$, the optimum solution of (LPP1) for transition t_1 is $\Gamma_{(LPP1)}^{(1)} = 11$ (see table 1), and it is obtained for the P-semiflows Y_2, Y_3, and Y_4 of those enumerated in (11). However, the improvement of this bound, presented in the second column of table 1, has been obtained from the RP-component generated by the P-semiflow Y_1, which was not optimum for (LPP1). This RP-component is considered by our generalized algorithm, for instance in the case of $N = 3$, using the constraint $Y^T \cdot PRE \cdot \vec{D}^{(i)} \geq \alpha \Gamma_{PS}^{(i)}$ at Step 2, with $\alpha = 0.9$.

5 Conclusions

We have addressed the problem of computing upper bounds for the throughput of transitions in stochastic Petri net models (or the corresponding synchronized queueing networks). Our approach is based on a decomposed view of these models.

Until now, the net structure, the initial marking, the long run routing rates, and the mean service time of transitions had been used for the solution of a linear programming problem in order to compute throughput upper bounds. An improvement has been presented in this paper. It is achieved by considering the throughput of some "slower" embedded queueing networks generated by P-semiflows, assumed in "partial isolation", i.e., taking into account the maximum reentrance in steady state (or liveness bound) of their timed transitions, allowed by the rest of the net. These embedded networks can be seen as closed product-form monoclass queueing networks, and efficient algorithms can be applied for the computation of their exact (or upper bound on) throughput. A particularly interesting case is that of live and bounded free choice systems, since all minimal P-semiflows generate RP-components (i.e., strongly connected "state machine-topology" subnets, the choices being free in the global net by definition). The above improvement has been derived under some technical conditions, (i) or (ii) in property 3.2, over the considered RP-components. It is an open problem to derive a formal proof of the non-necessity of any of these conditions.

The addition of implicit places to the original net system can generate new embedded queueing networks, allowing additional improvements in the bounds. A cost-quality tradeoff appears if the search for the "slowest" embedded queueing network considers also non-optimal solutions of (LPP1).

Acknowledgements

The authors are indebted to three anonymous referees whose comments and corrections have been really useful.

References

[1] M. Ajmone Marsan, G. Balbo, A. Bobbio, G. Chiola, G. Conte, and A. Cumani. The effect of execution policies on the semantics and analysis of stochastic Petri nets. *IEEE Transactions on Software Engineering*, 15(7):832–846, July 1989.

[2] F. Baccelli, G. Cohen, and B. Gaujal. Recursive equations and basic properties of timed Petri nets. *Discrete Event Dynamic Systems: Theory and Applications*, 1:415–439, 1992.

[3] F. Baskett, K. M. Chandy, R. R. Muntz, and F. Palacios. Open, closed, and mixed networks of queues with different classes of customers. *Journal of the ACM*, 22(2):248–260, April 1975.

[4] J. Campos, G. Chiola, and M. Silva. Ergodicity and throughput bounds of Petri nets with unique consistent firing count vector. *IEEE Transactions on Software Engineering*, 17(2):117–125, February 1991.

[5] J. Campos, G. Chiola, and M. Silva. Properties and performance bounds for closed free choice synchronized monoclass queueing networks. *IEEE Transactions on Automatic Control*, 36(12):1368–1382, December 1991.

[6] J. Campos, J. M. Colom, and M. Silva. Improving throughput upper bounds for net based models. In *Proceedings of the IMACS-IFAC SYMPOSIUM Modelling and Control of Technological Systems*, pages 573–582, Lille, France, May 1991. To appear in *IMACS Transactions*.

[7] J. Campos, B. Sánchez, and M. Silva. Throughput lower bounds for Markovian Petri nets: Transformation techniques. In *Proceedings of the* 4^{rd} *International Workshop on Petri Nets and Performance Models*, pages 322–331, Melbourne, Australia, December 1991. IEEE-Computer Society Press.

[8] J. Campos and M. Silva. Throughput upper bounds for Markovian Petri nets: Embedded subnets and queueing networks. In *Proceedings of the* 4^{rd} *International Workshop on Petri Nets and Performance Models*, pages 312–321, Melbourne, Australia, December 1991. IEEE-Computer Society Press.

[9] G. Chiola. A graphical Petri net tool for performance analysis. In *Proceedings of the* 3^{rd} *International Workshop on Modeling Techniques and Performance Evaluation*, Paris, France, March 1987. AFCET.

[10] J. M. Colom and M. Silva. Improving the linearly based characterization of P/T nets. In G. Rozenberg, editor, *Advances in Petri Nets 1990*, volume 483 of *LNCS*, pages 113–145. Springer-Verlag, Berlin, 1991.

[11] Y. Dallery, Z. Liu, and D. Towsley. Equivalence, reversibility and symmetry properties in fork/join queueing networks with blocking. Technical report, MASI 90-32, University Paris 6, 4 Place Jussieu, Paris, France, June 1990.

[12] D. L. Eager and K. C. Sevcik. Performance bound hierarchies for queueing networks. *ACM Transactions on Computer Systems*, 1(2):99–115, May 1983.

[13] J. Esparza and M. Silva. On the analysis and synthesis of free choice systems. In G. Rozenberg, editor, *Advances in Petri Nets 1990*, volume 483 of *LNCS*, pages 243–286. Springer-Verlag, Berlin, 1991.

[14] P. W. Glynn, and W. Whitt. A central-limit-theorem version of $L = \lambda W$. *Queueing Systems*, 2:191–215, 1986.

[15] M. H. T. Hack. Analysis of production schemata by Petri nets. M. S. Thesis, TR-94, M.I.T., Boston, USA, 1972.

[16] W. Henderson and P. G. Taylor. Embedded processes in stochastic Petri nets. *IEEE Transactions on Software Engineering*, 17(2):108–116, February 1991.

[17] F. Kelly. *Reversibility and stochastic networks.* Wiley, New York, 1979.

[18] A. A. Lazar and T. G. Robertazzi. Markovian Petri net protocols with product form solution. *Performance Evaluation*, 12:66–77, 1991.

[19] T. Murata. Petri nets: Properties, analysis, and applications. *Proceedings of the IEEE*, 77(4):541–580, April 1989.

[20] G. L. Nemhauser, A. H. G. Rinnooy Kan, and M. J. Todd, editors. *Optimization*, volume 1 of *Handbooks in Operations Research and Management Science.* North-Holland, Amsterdam, The Netherlands, 1989.

[21] M. Reiser and S. S. Lavenberg. Mean value analysis of closed multichain queueing networks. *Journal of the ACM*, 27(2):313–322, April 1980.

[22] C. H. Sauer, E. A. MacNair, and J. F. Kurose. The research queueing package: past, present, and future. In *Proceedings of the 1982 National Computer Conference.* AFIPS, 1982.

[23] J. G. Shanthikumar and D. D. Yao. The effect of increasing service rates in a closed queueing network. *Journal of Applied Probability*, 23:474–483, 1986.

[24] J. Zahorjan, K. C. Sevcik, D. L. Eager, and B. Galler. Balanced job bound analysis of queueing networks. *Communications of the ACM*, 25(2):134–141, February 1982.