Properties and Performance Bounds for Timed Marked Graphs

Javier Campos* Giovanni Chiola† José M. Colom* Manuel Silva*

Abstract

A class of synchronized queueing networks with deterministic routing is identified to be equivalent to a subclass of timed Petri nets called marked graphs. First some structural and behavioral properties of marked graphs are recalled and used to show interesting properties of this class of performance models. In particular, ergodicity is derived from boundedness and liveness of the underlying Petri net representation, which can be efficiently computed in polynomial time on the net structure. In case of unbounded (i.e., non-strongly-connected) marked graphs, ergodicity is computed as a function of the average transition firing delays. Then the problem of computing both upper and lower bounds for the steady-state performance of timed and stochastic marked graphs is studied. In particular, linear programming problems defined on the incidence matrix of the underlying Petri nets are used to compute tight (i.e., attainable) bounds for the throughput of transitions for marked graphs with deterministic or stochastic time associated with transitions. These bounds depend on the initial marking and the mean values of the delays but not on the probability distribution functions (thus including both the deterministic and the stochastic cases). The benefits of interleaving qualitative and quantitative analysis of marked graph models are shown.

Keywords: Petri nets, marked graphs, structural analysis, synchronized queueing networks, qualitative properties, ergodicity, performance evaluation, upper and lower bounds, throughput, linear programming.

1 Introduction

Queueing network models are one of the more popular and classical tools for the performance evaluation of computer systems [1]. With the advent of complex distributed systems, many proposals have been made to extend the modeling power of queueing networks by adding various synchronization constraints to the basic model [2, 3, 4]. One of the most important characteristics of queueing networks determining their popularity was the development of efficient, polynomial-complexity numerical solution algorithms, based on their "product form solution" [1]. Unfortunately, the introduction of synchronization constraints usually destroys this nice property.

More recently, timed and/or stochastic Petri net models have been introduced as a modeling tool capable of naturally representing synchronization and concurrency [5, 6, 7]. The intimate relation between some classes of synchronized queueing networks and some structural subclasses of timed Petri nets has already been recognized and studied by several authors [3, 8, 9, 10].

One of the main problems in the actual use of timed and stochastic Petri net models for the performance evaluation of large systems is the explosion of the computational complexity of the analysis algorithms. In general, exact performance measures are obtained from the numerical solution of a

*Departamento de Ingeniería Eléctrica e Informática, Universidad de Zaragoza, Maria de Luna 3, E-50015 Zaragoza, Spain. This work has been partially supported by the projects PA86-0028 of the Spanish Comisión Interministerial de Ciencia y Tecnología (CICYT), TIC 0358/89 of the Spanish Plan Nacional de Investigación, and the DEMON ESPRIT Basic Research Action 3148.

†Dipartimento di Informatica, Università degli Studi di Torino, corso Svizzera 185, I-10149 Torino, Italy. This work was performed while G. Chiola was visiting the University of Zaragoza, with the financial support of the Caja de Ahorros de la Inmaculada de Zaragoza under project Programa Europa of CAI–DGA.

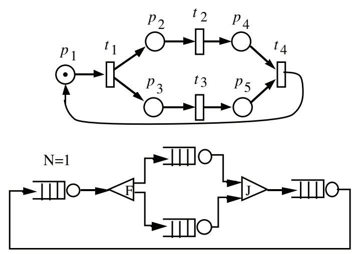


Figure 1: Example of a safe MG and its synchronized queueing network counterpart.

Markov chain, whose dimension is given by the size of the state space of the model. Simplified methods of computational complexity polynomial on the size of Petri net description have been developed by introducing restrictions in the modeling power of Petri nets [11].

In this paper, we obtain *upper and lower bounds* on the steady-state performance of marked graphs (MG), a well-known subclass of Petri nets that allows only concurrency and synchronization but no choice. Even if MG's is a restricted subclass of Petri nets, it is not difficult to realize the equivalence of strongly-connected stochastic MG's and *fork/join queueing networks with blocking* [10]. In particular, in this paper we study the throughput of transitions, defined as the average number of firings per unit time (or its inverse, that we call the mean cycle time of transitions). From this quantity, by applying Little's formula [1], it is possible to derive other average performance estimates of the model. Under these restrictions, we will show results that can be computed in polynomial time on the size of the net model, and that depend only on the mean values and not on the higher moments of the probability distribution functions (PDF) of the random variables that describe the timing of the system. This notion of independence of the computed mean measures on the form of the PDF is known as the *insensitivity property* in queueing networks literature. The independence of the probability distribution can be viewed as a practical estimation of the performance results, since higher moments of the delays are usually unknown for real cases, and difficult to estimate and assess. The bounds that we compute are based on proper *linear programming problems* (LPP) [12] that use the incidence matrix of the net. Previous works where linear programming formulation is used to solve qualitative problems on MG's can be found in [13, 14]. In [15], linear programming techniques are applied for the structural analysis of more general net classes.

Moreover, we show that both upper and lower bounds are attainable, in the sense that for any MG model it is possible to define families of MG models with stochastic timing such that the steady-state performances of the timed Petri net models are arbitrarily close to either bound.

Figure 1 depicts an example of a live and safe MG. In the same figure, the equivalent representation in terms of a queueing network with synchronization primitives [2] is also depicted. According to Figure 1, Petri net places correspond to queues (including waiting room and customer being served), while transitions represent servers and synchronization constraints. It is easily seen that only "sum" and "max" operators are needed to compute the performance: indeed the actual cycle time in this example is the random variable $\gamma = \tau_1 + \max(\tau_2, \tau_3) + \tau_4$, where τ_i denotes the *firing delay* of transition t_i (or its *service time*, with queueing networks terminology). Therefore, the average cycle time is

$$\Gamma \;=\; E[\tau_1] + E[\max(\tau_2,\tau_3)] + E[\tau_4] \;=\; \theta_1 + E[\max(\tau_2,\tau_3)] + \theta_4 \qquad (1)$$

(where θ_i denotes the average firing delay of transition t_i, i.e., its average service time). Cohen et al. developed a special algebra to formalize the properties of this kind of model in the deterministic case [16]. Baccelli et al. extended this approach to the stochastic case [4, 8].

Our idea is that of computing fast bounds for the throughput of transitions based only on the knowledge of the first moments of the PDF. This idea can be intuitively explained as follows. The sum is independent of the probability distribution (for linearity); since for non-negative variables $x_i \leq \max_i(x_i) \leq \sum_i x_i$, $E[\max_i(x_i)]$ can be bounded by $\max_i(E[x_i]) \leq E[\max_i(x_i)] \leq \sum_i E[x_i]$. Therefore, for the net in Figure 1 we can write:

$$\theta_1 + \max(\theta_2, \theta_3) + \theta_4 \leq \Gamma \leq \theta_1 + \theta_2 + \theta_3 + \theta_4 \tag{2}$$

In this paper, which is an improved version of [17], we show how LPP's based on the incidence matrix of the underlying Petri net structure can be solved to compute these kinds of bounds for marked graphs, even if they are not strongly-connected (i.e., if live, they are unbounded).

The paper is organized as follows. Section 2 contains a discussion of the implications that the introduction of timing semantics has on the behavior of an MG model. In particular the concepts of enabling and liveness bounds and weak ergodicity are defined. Sections 3 and 4 present the upper and lower bounds, respectively, for strongly-connected MG's. In Section 5 the bounds are extended to the non-strongly-connected case. An outline of the algorithm to find the derived bounds is presented in Section 6, in a step-by-step form. Section 7 presents a non-trivial application example taken from the literature for the evaluation of a complex multiprocessor computer architecture. Section 8 contains some concluding remarks and considerations on extensions of the work.

2 Stochastic interpretation of marked graphs

We start by giving a brief recall of the basic Petri net terminology and notation, and then we present some concepts concerning the introduction of the notion of time. The reader is referred to [18] for a nice tutorial on Petri nets.

2.1 Place/transition nets

A *place/transition* net $\mathcal{N}$ is defined as a 4-tuple $\mathcal{N} = \langle P, T, Pre, Post \rangle$, where P is the set of *places* ($|P| = n$), T is the set of *transitions* ($P \cap T = \emptyset, P \cup T \neq \emptyset$) ($|T| = m$), Pre ($Post$) is the pre- (post-) incidence function representing the input (output) arcs $Pre : P \times T \rightarrow \mathbb{N}$ ($Post : P \times T \rightarrow \mathbb{N}$).

The pre- and post-incidence functions can be represented as $n \times m$ matrices Pre and $Post$ with elements $Pre(p_i, t_j)$ and $Post(p_i, t_j)$, respectively. The *incidence matrix* C of the net is defined by $C(p_i, t_j) = Post(p_i, t_j) - Pre(p_i, t_j)$. We denote by $Pre[p]$, $Post[p]$, and $C[p]$ the row vectors of matrices Pre, $Post$, and C (respectively) corresponding to place p. $Pre[t]$, $Post[t]$, and $C[t]$ are the column vectors of Pre, $Post$, and C (respectively) corresponding to transition t.

The *pre-* and *post-set* of a transition $t \in T$ are defined respectively as ${}^\bullet t = \{p | Pre(p,t) > 0\}$ and $t^\bullet = \{p | Post(p,t) > 0\}$. The *pre-* and *post-set* of a place $p \in P$ are defined respectively as ${}^\bullet p = \{t | Post(p,t) > 0\}$ and $p^\bullet = \{t | Pre(p,t) > 0\}$.

A function $M : P \rightarrow \mathbb{N}$ is called a *marking*. A marking M can be represented in vector form, with the i^{th} component associated with the i^{th} element of P. A *marked net* $\langle \mathcal{N}, M_0 \rangle$ is a net $\mathcal{N}$ with an *initial marking* M_0.

A transition $t \in T$ is *enabled* at marking M iff $\forall p \in P$: $M(p) \geq Pre(p,t)$. A transition t enabled at M can *fire* yielding a new marking M' defined by $M'(p) = M(p) - Pre(p,t) + Post(p,t) = M(p) + C(p,t), \forall p \in P$. We denote as $M[t\rangle M'$ that transition t is enabled at M and that M' is *reached* from M by firing t.

A sequence of transitions $\sigma = t_1 t_2 \ldots t_n$ is a *firing sequence* of $\langle \mathcal{N}, M_0 \rangle$ iff there exist a sequence of markings such that $M_0[t_1\rangle M_1[t_2\rangle M_2 \ldots [t_n\rangle M_n$. In this case, marking M_n is said to be *reachable* from M_0 by firing σ, and this is denoted by $M_0[\sigma\rangle M_n$.

$M[\sigma\rangle$ denotes a firable sequence σ from marking M. The function $\vec{\sigma} : T \rightarrow \mathbb{N}$ is the *firing count vector* of the firable sequence σ, i.e., $\vec{\sigma}(t)$ represents the number of occurrences of $t \in T$ in σ. If $M_0[\sigma\rangle M$, then we can write in vector form $M = M_0 + C \cdot \vec{\sigma}$, which is referred to as the *linear* *state equation* of the net. A marking M' is said to be *potentially reachable* iff $\exists \vec{\sigma} \geq 0$ such that $M' = M_0 + C \cdot \vec{\sigma}$.

The *reachability set* $R(\mathcal{N}, M_0)$ is the set of all markings reachable from the initial one. Denoting by $PR(\mathcal{N}, M_0)$ the set of all potentially reachable markings we have the following relation: $R(\mathcal{N}, M_0) \subseteq PR(\mathcal{N}, M_0)$. $L(\mathcal{N}, M_0)$ is the set of all firing sequences and their suffixes in $\langle \mathcal{N}, M_0 \rangle$: $L(\mathcal{N}, M_0) = \{\sigma | M[\sigma\rangle \text{ and } M \in R(\mathcal{N}, M_0)\}$.

A place $p \in P$ is said to be *k–bounded* iff $\forall M \in R(\mathcal{N}, M_0)$: $M(p) \leq k$. A marked net $\langle \mathcal{N}, M_0 \rangle$ is said to be (marking) k–bounded iff each of its places is k–bounded, and it is said to be *bounded* iff it is k–bounded for some finite k. A net $\mathcal{N}$ is *structurally bounded* iff $\forall M_0$ the marked nets $\langle \mathcal{N}, M_0 \rangle$ are bounded.

A transition $t \in T$ is *live* in $\langle \mathcal{N}, M_0 \rangle$ iff $\forall M \in R(\mathcal{N}, M_0)$: $\exists M' \in R(\mathcal{N}, M)$ such that M' enables t. The marked net $\langle \mathcal{N}, M_0 \rangle$ is live iff all its transitions are live (i.e., liveness of the net guarantees the possibility of an infinite activity of all transitions). A net $\mathcal{N}$ is *structurally live* iff $\exists M_0$ such that the marked net $\langle \mathcal{N}, M_0 \rangle$ is live. The marked net $\langle \mathcal{N}, M_0 \rangle$ is *deadlock-free* iff $\forall M \in R(\mathcal{N}, M_0)$: $\exists t \in T$ such that M enables t.

A *repetitive component* is a function (vector) $X : T \rightarrow \mathbb{N}$ such that $X \neq 0$ and $C \cdot X \geq 0$. A *consistent component* (or *T-semiflow*) is a repetitive component X such that $C \cdot X = 0$. A *conservative component* (or *P-semiflow*) is a function (vector) $Y : P \rightarrow \mathbb{N}$ such that $Y \neq 0$ and $Y^T \cdot C = 0$. The *supports* of (T- and P-) semiflows are defined by $||X|| = \{t \in T | X(t) > 0\}$ and $||Y|| = \{p \in P | Y(p) > 0\}$. A (T- or P-) semiflow I has a *minimal support* iff there exist no other semiflow I' such that $||I'|| \subset ||I||$. A (T- or P-) semiflow is *canonical* iff the greatest common divisor of its components is 1. A (T- or P-) semiflow is *elementary* iff it is canonical and has a minimal support.

A net $\mathcal{N}$ is *repetitive* if there exists a repetitive component $X \geq \mathbf{1}$ (where $\mathbf{1}$ is a vector with all its entries equal to 1). A net $\mathcal{N}$ is *consistent* if there exists a T-semiflow $X \geq \mathbf{1}$. A net $\mathcal{N}$ is *conservative* if there exists a P-semiflow $Y \geq \mathbf{1}$.

An *implicit* place is one which never is the only one that restricts the firing of its output transitions. Let $\mathcal{N}$ be any net and $\mathcal{N}^p$ be the net resulting from adding a place p to $\mathcal{N}$. If M_0 is an initial marking of $\mathcal{N}$, M_0^p denotes the initial marking of $\mathcal{N}^p$. The place p is implicit in the marked net $\langle \mathcal{N}^p, M_0^p \rangle$ iff $L(\mathcal{N}^p, M_0^p) = L(\mathcal{N}, M_0)$.

$M \in R(\mathcal{N}, M_0)$ is a *home state* iff $\forall M_i \in R(\mathcal{N}, M_0) : M \in R(\mathcal{N}, M_i)$. A marked net is *reversible* iff its initial marking is a home state.

2.2 Marked graphs

Marked graphs are a subclass of Petri nets characterized by the fact that each place has exactly one input and exactly one output arc, thus they are structurally decision-free nets.

Definition 2.1 [18] *A marked graph (MG) is an ordinary Petri net (pre- and post-incidence functions taking values in* $\{0, 1\}$*) such that* $|^\bullet p| = |p^\bullet| = 1, \forall p \in P$.

Theorem 2.1 [18] *Let* $\langle \mathcal{N}, M_0 \rangle$ *be an MG.*

1. *The minimal support P-semiflows of* $\mathcal{N}$ *are exactly its directed elementary circuits.*

2. $\langle \mathcal{N}, M_0 \rangle$ *is live iff all its directed circuits are marked.*

Corollary 2.1 *Let* $\langle \mathcal{N}, M_0 \rangle$ *be an MG system and* $\mathcal{N}_0$ *the net structure obtained deleting all places marked by* M_0. $\langle \mathcal{N}, M_0 \rangle$ *is live if and only if* $\mathcal{N}_0$ *has no circuit (acyclic graph).*

In the sequel, when we write *circuit* we refer to an *elementary circuit*. Using the equivalence between P-semiflows and circuits, the next result follows.

Corollary 2.2 *The liveness of an MG can be decided in polynomial time on its size, checking that there do not exist unmarked P-semiflows:* $\nexists Y \geq 0, Y \neq 0, Y^T \cdot C = 0 : \; Y^T \cdot M_0 = 0$.

Property 2.1 *Let $\mathcal{N}$ be an MG.*

1. *The following three statements are equivalent:*
   - *i) $\mathcal{N}$ is structurally bounded.*
   - *ii) $\mathcal{N}$ is strongly-connected.*
   - *iii) $\mathcal{N}$ is conservative (i.e., $\exists Y > 0, Y^T \cdot C = 0$).*
2. *Let $\langle \mathcal{N}, M_0 \rangle$ be live. Then $\langle \mathcal{N}, M_0 \rangle$ is bounded iff $\mathcal{N}$ is structurally bounded.*
3. *$\mathcal{N}$ is a consistent net and its unique minimal T-semiflow is $X = \mathbf{1}$.*

Theorem 2.2 [18] *Let $\langle \mathcal{N}, M_0 \rangle$ be a live (possibly unbounded) MG. The three following statements are equivalent:*

i) $M \in R(\mathcal{N}, M_0)$, i.e., M is reachable from M_0.

ii) $M = M_0 + C \cdot \vec{\sigma}$, with $M \in \mathbb{N}^n, \vec{\sigma} \geq 0$.

iii) $B_f \cdot M = B_f \cdot M_0$, with B_f the fundamental circuit matrix of the graph, and $M \in \mathbb{N}^n$.

According to the above theorem $M \in R(\mathcal{N}, M_0) \Leftrightarrow M_0 \in R(\mathcal{N}, M)$. In other words:

Corollary 2.3 *Live MG's are reversible.*

2.3 Timing and firing process

In the original definition, Petri nets did not include the notion of time, and tried to model only the logical behavior of systems by describing the causal relations existing between otherwise unrelated events. This approach showed its power in the specification and analysis of concurrent systems. Nevertheless, the introduction of timing specification is essential if we want to use this class of models for an evaluation of the performance of distributed systems.

Since Petri nets are bipartite graphs, historically there have been two ways of introducing the concept of time in them, namely, associating a time interpretation with either transitions [19] or places [20]. Moreover, in the case of timed transition models, two different firing rules have been defined: single phase (atomic) firing, and three phase (start-firing with deletion of the input tokens, delay, and end-firing with creation of the output tokens). Since in the context of MG's no *conflict* situation can ever arise, all these alternatives become equivalent for our purposes: we only say that we consider timed-transition MG's, without further specification of the firing semantics.

Another possible source of confusion in the definition of the timed interpretation of a Petri net model is the concept of *degree of enabling* of a transition (or *re-entrance*). In the case of timing associated with places, it seems quite natural to define an *unavailability time* which is independent of the total number of tokens already present in the place, and this can be interpreted as an *infinite server* policy from the point of view of queueing theory. In the case of time associated with transitions, it is less obvious a-priori whether a transition enabled k times simultaneously at a marking should work at conditional throughput 1 or k times that in the case it was enabled only once. In the case of stochastic Petri nets with exponentially distributed firing times associated with transitions, the usual implicit hypothesis is to have *single server* semantics (see, e.g., [21, 22]), and the case of *multiple server* is handled as a case of firing rate dependent on the marking; this trick cannot work in the case of more general probability distributions. This is the reason why people working with deterministic timed transition Petri nets prefer an infinite server semantics (see, e.g., [23, 24, 25]). Of course a transition with infinite server semantics can always be constrained to a *k-server* behavior by just reducing its enabling bound to k. This can be obtained by adding one place that is both input and output (self-loop whose arcs are weighted with 1) for that transition and marking it with k tokens.

Therefore the infinite server semantics appears to be the most general one, and for this reason it is adopted in this work.

From a Petri net perspective, the queueing network stations are represented by timed transitions. The maximum number of servers working in parallel in a station will be characterized with the *enabling bound* concept. Since we are interested in the steady-state performance of models, only the maximum number of servers working in parallel in steady-state must be considered. The *liveness bound* concept will give that index.

Definition 2.2 *Let $\langle \mathcal{N}, M_0 \rangle$ be a marked Petri net and $t \in T$ a given transition of $\mathcal{N}$. The enabling bound of t is* $EB(t) \stackrel{\text{def}}{=} \max\{k \mid \exists M \in R(\mathcal{N}, M_0) : M \geq k\ Pre[t]\}$. *The liveness bound of t is* $LB(t) \stackrel{\text{def}}{=} \max\{k \mid \forall M' \in R(\mathcal{N}, M_0),\ \exists M \in R(\mathcal{N}, M') : M \geq k\ Pre[t]\}$.

Definition 2.3 *A marked Petri net $\langle \mathcal{N}, M_0 \rangle$ is said to be k–live iff* $\max\{LB(t) | t \in T\} = k$.

A transition t is live iff $LB(t) > 0$, i.e., if there is at least one server associated with it in steady-state conditions. Due to the reversibility property of live MG's (see Corollary 2.3), the enabling and liveness bounds yield the same value in all cases considered here.

The above definition of enabling bound refers to a behavioral property that, in the general case, must be computed on the reachability graph of a Petri net. Since we are looking for computational techniques at the structural level, we can also introduce the structural counterpart of the concept. Structural net theory has been developed from two complementary points of view: graph theory [26] and mathematical programming (or more specifically linear programming and linear algebra) [15]. Let us introduce our structural definition using mathematical programming arguments; essentially, in this case the reachability condition is relaxed to the potential reachability condition defined by the net state equation: $M = M_0 + C \cdot \vec{\sigma}$, with $M, \vec{\sigma} \geq 0$.

Definition 2.4 *Let $\langle \mathcal{N}, M_0 \rangle$ be a marked Petri net. The structural enabling bound of a given transition t of $\mathcal{N}$ is*

$$SEB(t) \stackrel{\text{def}}{=} \begin{array}{ll} \textit{maximum} & k \\ \textit{subject to} & M_0 + C \cdot \vec{\sigma} \geq k\ Pre[t] \\ & \vec{\sigma} \geq 0 \end{array} \qquad \text{(LPP1)}$$

Note that the definition of structural enabling bound reduces to the formulation of a linear programming problem (LPP1) [12] with *decision variables* $k, \vec{\sigma}$, and constraints based on incidence matrices of the net and on the initial marking. According to Theorem 2.2, for live MG's the next property follows:

Property 2.2 *Let $\langle \mathcal{N}, M_0 \rangle$ be an MG. Then* $EB(t) = LB(t) = SEB(t), \forall t \in T$.

This allows an efficient computation of enabling and liveness bounds based on the problem (LPP1) that characterizes the structural enabling bound.

In case of non-strongly-connected MG's, it is quite possible to obtain $SEB(t) = \infty$ for some transition t; this creates no harm: because of the assumption of an infinite server semantics, it only means that the timing of that transition does not affect the steady-state performance of the model.

2.4 Ergodicity and measurability

In order to define the steady-state performance of a system we have to assume that some kind of "average behavior" can be estimated on the long run of the system we are studying. The usual assumption in this case is that the system models must be (strongly) *ergodic* (see definitions of ergodicity in [21]). This assumption is very strong and difficult to verify in general; moreover, it creates problems when we want to include the deterministic case as a special case of a stochastic model (as an example, consider a customer going round a cycle of two queues with deterministic service; in this case, the state of each queue is a periodic function of time, hence the limit of expected values does not exist [27]). Thus, we introduce the concept of *weak ergodicity* that allows the estimation of long run performances also in the case of deterministic models.

Definition 2.5 *The marking process M_τ, where $\tau \geq 0$ represents the time, of a stochastic marked net is weakly ergodic (or measurable in the long run) iff the following limit exists:*

$$\lim_{\tau \to \infty} \frac{1}{\tau} \int_0^\tau M_u \, du = \overline{M} < \vec{\infty}, \; a.s. \tag{3}$$

and the constant vector $\overline{M}$ is called the limit mean marking.

The firing process $\vec{\sigma}_\tau$, where $\tau \geq 0$ represents the time, of a stochastic marked net is weakly ergodic (or measurable in the long run) iff the following limit exists:

$$\lim_{\tau \to \infty} \frac{\vec{\sigma}_\tau}{\tau} = \vec{\sigma}^* < \vec{\infty}, \; a.s. \tag{4}$$

and the constant vector $\vec{\sigma}^*$ is the limit firing flow vector.

According to the above definitions and the properties listed in Section 2.2 the next result follows:

Corollary 2.4 *Strongly-connected live MG's have weakly ergodic firing and marking processes, independently of the transition firing delays. For non-strongly-connected live MG's, weak ergodicity is guaranteed for the firing process.*

In the above statement, it is assumed that at least one transition has non-null firing delay (in order to obtain a finite value of the limit firing flow).

3 Throughput upper bounds for strongly-connected MG's

In this and the next Section, performance bounds for the steady-state behavior of strongly-connected (and thus structurally bounded, by Property 2.1) MG's are presented. We recall that strong connectivity of a graph is a well-known problem of polynomial time complexity [28].

3.1 Computation of the bound

Let us take into account only the first moment of the PDF's associated with transitions. In the following, let θ_i be the mean value of the random variable associated with the firing time of transition t_i (service time of t_i, with queueing networks terminology) and θ the vector with components θ_i, $i = 1, \ldots, m$. The *mean cycle time*, Γ_i, of transition t_i is the mean time between two consecutive firings of t_i:

$$\Gamma_i = \frac{1}{\vec{\sigma}^*(t_i)} \tag{5}$$

where $\vec{\sigma}^*$ is the limit firing flow vector (or vector of transitions' throughputs).

The *relative firing frequency vector* (or vector of *visit ratios*) is the limit firing flow vector normalized for having a given component equal to 1. Since MG's are consistent nets and their unique minimal T-semiflow is **1** (cfr. Property 2.1.3), their relative firing frequency vector is also **1** (assuming weak ergodicity of marking and firing), therefore $\Gamma_i = \Gamma, \forall i = 1, \ldots, m$, and Γ is the mean cycle time of the MG.

The following Little's formula for stochastic Petri nets [21] holds under the weak ergodicity assumption:

$$\overline{M}(p_i) = (Pre[p_i] \cdot \vec{\sigma}^*)\ \overline{R}(p_i) \tag{6}$$

where $\overline{M}(p_i)$ is the limit mean marking of place p_i, $Pre[p_i]$ is the i^{th} row of the pre-incidence matrix, and $\overline{R}(p_i)$ is the mean response time at place p_i (i.e., the mean sojourn time of tokens: sum of the waiting time and the service time). The response times at places are unknown but can be bounded from below by the mean firing time associated with transitions:

$$\Gamma\ \overline{M} \geq Pre \cdot \theta \tag{7}$$

Since the vector $\overline{M}$ is unknown, (7) cannot be solved. However, the following structural marking invariant can be written using a P-semiflow Y:

$$Y^T \cdot M_0 = Y^T \cdot M = Y^T \cdot \overline{M}, \quad \forall M \in R(\mathcal{N}, M_0) \tag{8}$$

Now, from (7) and (8):

$$\Gamma\ (Y^T \cdot M_0) \geq Y^T \cdot Pre \cdot \theta \tag{9}$$

And a lower bound for the mean cycle time in steady-state is:

$$\Gamma^{min} = \max_{Y \in \{P-semiflow\}} \frac{Y^T \cdot Pre \cdot \theta}{Y^T \cdot M_0} \tag{10}$$

Of course, an upper bound for the throughput of transitions is $1/\Gamma^{min}$.

Let us formulate the previous lower bound for the mean cycle time in terms of a particular class of optimization problems called *fractional programming problems* [12]:

$$\begin{array}{rll} \Gamma^{min} = & \text{maximum} & \dfrac{Y^T \cdot Pre \cdot \theta}{Y^T \cdot M_0} \\ & \text{subject to} & Y^T \cdot C = 0, \quad \mathbf{1}^T \cdot Y > 0 \\ & & Y \geq 0 \end{array} \tag{11}$$

The above problem can be rewritten as follows:

$$\begin{array}{rll} \Gamma^{min} = & \text{maximum} & \dfrac{Y^T \cdot Pre \cdot \theta}{q} \\ & \text{subject to} & Y^T \cdot C = 0, \quad \mathbf{1}^T \cdot Y > 0 \\ & & q = Y^T \cdot M_0, \quad Y \geq 0 \end{array} \tag{12}$$

Then, because $Y^T \cdot M_0 > 0$ (guaranteed for live MG's, by Corollary 2.2), we can change Y/q to Y and obtain the linear programming formulation stated in the next theorem (in which $\mathbf{1}^T \cdot Y > 0$ is removed because $Y^T \cdot M_0 = 1 \Longrightarrow \mathbf{1}^T \cdot Y > 0$).

Theorem 3.1 *A lower bound for the mean cycle time for live strongly-connected MG's can be obtained by solving the following LPP:*

$$\begin{array}{rll} \Gamma^{min} = & \textit{maximum} & Y^T \cdot Pre \cdot \theta \\ & \textit{subject to} & Y^T \cdot C = 0, \quad Y^T \cdot M_0 = 1 \\ & & Y \geq 0 \end{array} \tag{LPP2}$$

The following result concerns a special class of optimum solutions of (LPP2) that will be used later in the interpretation of this LPP: *the minimal P-semiflows.* In order to prove this result, we use the concept of *basic feasible solution* from linear programming [12], and the problem (LPP2) rewritten in the following way:

$$\Gamma^{min} = \begin{array}{ll} \text{maximum} & Y^T \cdot Pre \cdot \theta \\ \text{subject to} & Y^T \cdot [C|M_0] = (0|1) \\ & Y \geq 0 \end{array} \qquad \text{(LPP3)}$$

Let $\mathcal{Y}$ be the set of feasible solutions Y of (LPP3). If $Y \in \mathcal{Y}$, the set of row vectors of $A = [C|M_0]$ that Y uses is $\{A[j] \mid j \text{ is such that } Y[j] > 0\}$. The feasible solution $Y \in \mathcal{Y}$ is said to be a basic feasible solution for (LPP3) iff the set of row vectors of A that Y uses is a linearly independent set.

Theorem 3.2 *Under the conditions of Theorem 3.1, if (LPP2) has an optimum solution, then it has an optimum solution which is a minimal P-semiflow.*

Proof. Taking into account Theorem 3.3 in [12], if (LPP3) has an optimum feasible solution, then it has a basic feasible solution Y that is optimum. Therefore, the set of rows that are used by Y is linearly independent (i.e., full rank). Considering that $Y^T \cdot C = 0$, we obtain that the number of non-null entries of vector Y (i.e., the number of rows used by Y) is equal to the rank of rows of C used by Y plus one. This last statement is precisely the characterization of a minimal P-semiflow, presented in [29]. ∎

It is well known that the simplex method [12] for the solution of LPP's gives good results in practice, even if it has exponential worst-case complexity. Moreover, the simplex method gives feasible solutions being basic solutions. In any case, a discussion on algorithms of polynomial worst-case complexity can be found in [30].

Theorem 3.1 shows that the problem of finding an upper bound for the steady-state throughput (lower bound for the mean cycle time) in a strongly-connected stochastic MG can be solved looking at the mean cycle time associated with each minimal P-semiflow (circuits for MG's) of the net, considered in isolation. These mean cycle times can be computed by taking the summation of the average firing times of all the transitions involved in the P-semiflow (service time of the whole circuit), and dividing by the number of tokens present in it (customers in the circuit). Therefore, an alternative approach to linear programming for the computation of the bound could be based on graph theory. We select linear programming approach (also with polynomial solution algorithms) because of the possibility of easily interpret, derive new results, and extend to other net subclasses (e.g., live and bounded free choice nets [9]).

The above bound, that holds for any stochastic interpretation, happens to be the same as that obtained for strongly-connected deterministic MG's by other authors (see for example [19, 31]), but here it is considered in a practical LPP form. For deterministically timed nets, the attainability of this bound has been shown [19, 31]. Since deterministic timing is just a particular case of stochastic timing, the attainability of the bound is assured for our purposes as well. Even more, the next result shows that the previous bound cannot be improved only on the basis of the knowledge of the coefficients of variation for the transition firing times.

Theorem 3.3 *For live strongly-connected MG's with arbitrary values of mean and variance for transition firing times, the bound for the mean cycle time obtained from (LPP2) cannot be improved.*

Proof. We know from [19] that for deterministic timing the bound is reached. Only "max" and sum operators are needed to compute the mean cycle time. Therefore, let us construct a family of random variables with arbitrary means and variances behaving in the limit like deterministic timing for both operators (max and sum).

This is the case for the following family of random variables, for varying values of the parameter α $(0 \leq \alpha \leq 1)$:

$$X_{\theta_i,\sigma_i}(\alpha) = \begin{cases} \theta_i \alpha & \text{with probability } 1 - \epsilon_i \\ \theta_i(\alpha + \frac{1-\alpha}{\epsilon_i}) & \text{with probability } \epsilon_i \end{cases} \qquad (13)$$

where

$$\epsilon_i = \frac{\theta_i^2(1-\alpha)^2}{\theta_i^2(1-\alpha)^2+\sigma_i^2} \tag{14}$$

These variables are such that $E[X_{\theta_i,\sigma_i}(\alpha)] = \theta_i$, $Var[X_{\theta_i,\sigma_i}(\alpha)] = \sigma_i^2$, and they satisfy:

$$\lim_{\alpha\to 1} E[\max(X_{\theta_i,\sigma_i}(\alpha), X_{\theta_j,\sigma_j}(\alpha))] = \max(\theta_i,\theta_j) \tag{15}$$

and, of course, $\forall\ 0 \le \alpha < 1$: $E[X_{\theta_i,\sigma_i}(\alpha) + X_{\theta_j,\sigma_j}(\alpha)] = \theta_i + \theta_j$.

Then, if random variables $X_{\theta_i,\sigma_i}(\alpha)$ are associated with transitions t_i, $i = 1,\ldots,m$, taking α closer to 1, the mean cycle time tends to the bound given by (LPP2). ∎

We remark that the contribution of the above theorem is the attainability of the bound for *any given means and variances* of involved random variables. In other words, even with the knowledge of second order moments, it is not possible to improve the bound given by (LPP2), computed only with mean values.

A polynomial computation of the minimal cycle time for deterministic timed strongly-connected MG's was proposed in [32], solving the following linear programming problem:

$$\begin{array}{rll} \Gamma^{min} = & \text{minimum} & \gamma \\ & \text{subject to} & -C\cdot z + \gamma M_0 \ge Post\cdot\theta \\ & & \gamma \ge 0, \quad z \ge 0 \end{array} \tag{LPP4}$$

where the decision variables are γ and z: γ represents the mean cycle time while z is the vector of instants at which each transition initiates its first firing.

To investigate the relationship between (LPP2) and (LPP4) let us consider the *dual* problem of (LPP4), as usually defined in mathematical programming [12]:

$$\begin{array}{rll} \Gamma^{min} = & \text{maximum} & Y^T\cdot Post\cdot\theta \\ & \text{subject to} & Y^T\cdot C \le 0, \quad Y^T\cdot M_0 \le 1 \\ & & Y \ge 0 \end{array} \tag{LPP5}$$

Since strongly-connected MG's are conservative (Property 2.1), there is no $Y \ge 0$ such that $Y^T\cdot C \le 0$, $Y^T\cdot C \ne 0$ and then the restriction $Y^T\cdot C \le 0$ of (LPP5) becomes $Y^T\cdot C = 0$ (i.e., the restriction of (LPP2)). For all Y such that $Y^T\cdot C = 0$, $Y^T\cdot Post = Y^T\cdot Pre$ holds. For live MG's, $\forall Y \in \mathbb{N}^n$, $Y \ne 0$ such that $Y^T\cdot C = 0$ then $Y^T\cdot M_0 \ge 1$ (Corollary 2.2). Thus, the restriction $Y^T\cdot M_0 \le 1$ of (LPP5) becomes $Y^T\cdot M_0 = 1$ for live nets (i.e., the restriction of (LPP2)).

Hence for live strongly-connected MG's, the problem (LPP2) is equivalent to (LPP4) formulated in [32] for deterministic systems.

3.2 Interpretation and derived results

Linear programming problems give an easy way to derive results and interpret them. Just looking at the objective function of the problem (LPP2), the following monotonicity property is obtained: the lower bound for the mean cycle time does not increase if θ decreases or if M_0 increases.

Property 3.1 *Let $\mathcal{N}$ be a live strongly-connected MG and θ the mean firing times vector.*

i) For a fixed θ, if $M_0' \ge M_0$ (i.e., increasing the number of initial resources) then the lower bound for mean cycle time of $\langle\mathcal{N}, M_0', \theta\rangle$ is less than or equal to that of $\langle\mathcal{N}, M_0, \theta\rangle$ (i.e., $\Gamma^{min'} \le \Gamma^{min}$).

ii) For a fixed M_0, if $\theta' \le \theta$ (i.e., for faster servers) then the lower bound for mean cycle time of $\langle\mathcal{N}, M_0, \theta'\rangle$ is less than or equal to the one of $\langle\mathcal{N}, M_0, \theta\rangle$ (i.e., $\Gamma^{min'} \le \Gamma^{min}$).

Note that, since ergodicity is assumed, the mean cycle time of the MG does not change for any reachable marking considered as initial marking. The next property, which is strongly related to the reversibility of live MG's (cfr. Corollary 2.3), states an analogous result for the bound computed in (LPP2).

Property 3.2 [17] *For any live strongly-connected MG $\langle \mathcal{N}, M_0 \rangle$, the bound obtained with the problem (LPP2) does not change for any marking reachable from M_0.*

The proof of the above property can be derived from Theorem 3.1 using the problem (LPP2). The condition over a marking M for being reachable is $M = M_0 + C \cdot \vec{\sigma} \in \mathbb{N}^n$ for some $\vec{\sigma} \geq 0$ (Theorem 2.2). Taking into account that $Y^T \cdot M = Y^T \cdot M_0$ for all marking M reachable from M_0 and for all Y such that $Y^T \cdot C = 0$, the statement of the property follows.

Since the upper bound on throughput is computed based on the total mean service time of isolated circuits and on the marking contained in them, it is easy to see that the *reverse net* of $\mathcal{N} = \langle P, T, Pre, Post \rangle$ defined as $\mathcal{N}^{-1} = \langle P, T, Post, Pre \rangle$ yields the same bound in case of strongly-connected MG's.

Property 3.3 *Let $\mathcal{N}$ be a strongly-connected MG and $\mathcal{N}^{-1}$ its reverse net. Then, the upper bounds on throughput obtained for both nets, preserving M_0, with the problem (LPP2) are the same.*

In particular, if deterministic timing is considered (since the bound gives the exact throughput in this case), the throughput of the original and the reverse net are equal (an analogous result under non-deterministic assumption for the distributions of timing is presented in [10]).

The next result is a characterization of liveness for MG's in terms of the finiteness of the mean cycle time.

Theorem 3.4 *A strongly-connected MG is live iff the value Γ^{min} given by (LPP2) in Theorem 3.1 is finite.*

Proof. For strongly-connected live MG's, all circuits contain at least one token (i.e., $Y^T \cdot M_0 > 0$ with $Y^T \cdot C = 0$). Therefore, there is no solution such that $Y^T \cdot C = 0, Y^T \cdot M_0 = 0$ and then (LPP2) is bounded (i.e., there is no extremal direction). If the optimum value of (LPP2) is finite, since it is attainable for some deterministic [19] as well as stochastic (cfr. Theorem 3.3) timing, the net must be deadlock-free. We know that for strongly-connected MG's, liveness and deadlock-freeness are equivalent. Thus the finiteness of the optimum value of (LPP2) is sufficient to establish the liveness of a strongly-connected MG. ■

4 Throughput lower bounds for strongly-connected MG's

In this section, we present the computation of lower bounds on throughput for strongly-connected MG's. We start by presenting an attainable lower bound for 1–live MG's, and then we extend the result to bounded MG's. Finally, we propose a polynomial complexity computation based on linear programming.

4.1 Basic result for 1–live strongly-connected MG's

A trivial lower bound on steady-state throughput for a live MG is of course given by the inverse of the sum of the firing times of all the transitions. Since the net is live all transitions must be firable, and the sum of all firing times corresponds to any *complete sequentialization* of all the activities represented in the model. This lower bound is always reached in an MG consisting of a single loop of transitions and containing a single token in one place, independently of the higher moments of the PDF's (this observation can be trivially confirmed by the computation of the upper bound, which in this case gives the same value).

To improve this trivial lower bound let us first consider the case of 1–live strongly-connected MG's. If we specify only the mean values of the transition firing times and not the higher moments, we may always find an stochastic model whose steady-state throughput is arbitrarily close to the trivial lower bound, independently of the topology of the MG (only provided that it is 1–live). A formal proof of this (somewhat counter-intuitive) result stated in the next theorem can be found in the Appendix. It is based on the definition of the family of random variables:

$$x_\mu^i(\epsilon) = \begin{cases} 0 & \text{with probability } 1-\epsilon^i \\ \mu/\epsilon^i & \text{with probability } \epsilon^i \end{cases} \tag{16}$$

for $\mu \geq 0; 0 < \epsilon \leq 1; i \in \mathbb{N}$. It is straightforward to see that $E[x_\mu^i(\epsilon)] = \mu$, and $E[(x_\mu^i(\epsilon))^2] = \mu^2/\epsilon^i$. This implies that the coefficient of variation is 0 for $\epsilon = 1$, and that it tends to ∞ as $\epsilon \to 0$ provided that $i > 0$. Then, the proof derives from considering that each transition t_j in the MG has $x_{\theta_j}^{j-1}(\epsilon)$ as random firing time distribution.

Theorem 4.1 *For any 1–live strongly-connected MG with a given specification of the mean firing times θ_j for each $t_j \in T$, it is possible to assign PDF's to the transition firing times such that the mean cycle time is $\Gamma = \sum_j \theta_j - O(\epsilon)$, $\forall \epsilon : 0 < \epsilon \leq 1$, independently of the topology of the net (and thus independently of the potential maximum degree of parallelism intrinsic in the MG). (We use here the notation $O(f(x))$ to indicate any function $g(x)$ such that $\lim_{x\to 0} \frac{g(x)}{f(x)} \leq k \in \mathbb{R}$.)*

In the previous result, the upper bound for the mean cycle time (thus lower bound on throughput) is reached in a limit case ($\epsilon \to 0$) in which the random variables associated with transitions have infinite coefficient of variation. This is a way to obtain the minimum throughput if firing times associated with transitions are assumed mutually uncorrelated. It can be shown that it is also possible to reach the lower bound in performance for finite coefficient of variation if a *maximum negative correlation* is assumed among the firing times of transitions.

4.2 Extension to bounded MG's

Until now, we have shown that the trivial sum of the mean firing times of all transitions in the net constitutes a tight (attainable) lower bound for the performance of a live and safe MG (or more generally of a 1–live strongly-connected MG, but otherwise independently of the topology) in which only the mean values and neither the PDF's nor the higher moments are specified for the transition firing times. Let us now extend this result to the more general case of k–live strongly-connected MG's.

An intuitive idea is to derive a lower bound on throughput for an MG containing transitions with liveness bound $k \geq 1$ (remember that, for MG's, $EB(t) = LB(t) = SEB(t)$, Property 2.2) by taking the method used for the computation of the upper bound in the case of non-safe MG's, and substitute in it the "max" operator for the sum of the firing times of all transitions involved. After some manipulation to avoid counting more than once the contribution of the same transition, one can arrive at the formulation of the following value for the maximum cycle time:

$$\Gamma^{max} = \sum_{j=1}^{m} \frac{\theta_j}{LB(t_j)} \tag{17}$$

The proof of this result requires the following Lemma.

Lemma 4.1 *Any strongly-connected MG with arbitrary initial marking can be constrained to contain a main circuit including all transitions, without changing their liveness bound. This main circuit (which, in general, is not unique) contains a number of tokens equal to the maximum of the liveness bounds among all transitions. In addition there are other minor circuits that preserve the liveness bounds for transitions with bound lower than the maximum.*

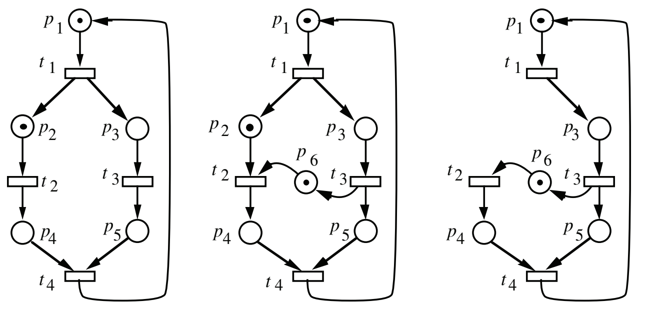


Figure 2: Example of structural sequentialization.

The idea behind this constraint is to introduce a structural sequentialization among all transitions, thus potentially reducing the degree of concurrency between the activities modeled by the transitions. In other words, from the partial order given by the initial MG structure, we try to derive a total order without changing the liveness bound. The proof of the previous Lemma can be found in the Appendix.

An example of application of the Lemma follows, in order to clarify the procedure. Consider the net depicted in Figure 2.a. This net contains only two circuits, namely t_1, t_2, t_4, and t_1, t_3, t_4; we can then add either the circuit t_1, t_2, t_3, t_4 or t_1, t_3, t_2, t_4; Figure 2.b depicts the resulting net in case we choose to add the second circuit. In this case only place p_6 (from t_3 to t_2) needs to be added to obtain the longer circuit, and it should be marked with one token, so that the new circuit comprising places p_1, p_3, p_6, p_4 contains two tokens, as the original circuit p_1, p_2, p_4 (while the other original circuit p_1, p_3, p_5 contained only one). In our example, we need not iterate the procedure since we have already obtained a circuit containing all transitions of the MG. At this point we can identify and eliminate the implicit places that have been created during the circuits interleaving procedure. In the present example, we can easily see that place p_2 becomes implicit in Figure 2.b, so that it can be eliminated, finally leading ourselves to the MG depicted in Figure 2.c.

It should be evident that the MG transformed by applying the above Lemma has a mean cycle time which is greater than or equal to the mean cycle time of the original one, since some additional constraints have been added to the enabling of transitions: hence the mean cycle time of the transformed MG is a lower bound for the performance of the original one. Now if $N_M = \max_{t \in T} EB(t) = 1$ in the above Lemma, we re-find the lower bound of Theorem 4.1. In the case of $N_M > 1$, we can show that the mean cycle time of the transformed net cannot exceed Γ^{max}.

Theorem 4.2 *For any live and bounded MG with a given specification of the mean firing times θ_j for each $t_j \in T$, it is not possible to assign PDF's to the transition firing times such that the mean cycle time is greater than*

$$\Gamma^{max} = \sum_{j=1}^{m} \frac{\theta_j}{LB(t_j)} \tag{18}$$

independently of the topology of the net.

A detailed proof of the above result is presented in the Appendix. The application of Theorem 4.2 to the example of Figure 2 gives: $\Gamma^{max} = \theta_1 + \theta_2/2 + \theta_3 + \theta_4$.

Now, two results for the lower bound on throughput, analogous to those in Properties 3.2 and 3.3, can be derived.

Property 4.1 *For any live strongly-connected MG $\langle \mathcal{N}, M_0 \rangle$, the bound obtained as in Theorem 4.2 does not change for any marking reachable from M_0.*

Property 4.2 *Let $\mathcal{N}$ be a strongly-connected MG and $\mathcal{N}^{-1}$ its reverse net. Then, the lower bounds on throughput obtained for both nets, as in Theorem 4.2, are the same.*

4.3 Attainability of the lower bound

The lower bound on steady-state throughput given by the computation of $1/\Gamma^{max}$, as defined in Theorem 4.2, can be shown to be attainable for any MG topology and for some assignment of PDF to the firing delay of transitions, exploiting the attainability of the trivial bound shown in Theorem 4.1 for 1–live MG's.

Theorem 4.3 *For any strongly-connected MG with a given specification of the mean firing times θ_j for each $t_j \in T$, and $\forall \epsilon : 0 < \epsilon \leq 1$, it is possible to assign PDF's to the transition firing times such that the average cycle time is:*

$$\Gamma^{max} = \sum_{j=1}^{m} \frac{\theta_j}{LB(t_j)} - O(\epsilon) \tag{19}$$

independently of the topology of the net.

A formal proof of Theorem 4.3 can be found in the Appendix.

4.4 A polynomial computation of the lower bound

First of all, we recall (cfr. Property 2.2) that in the case of live MG's the liveness bound equals the enabling and the structural enabling bounds for each transition (i.e., $LB(t) = EB(t) = SEB(t)$); thus we present a characterization for the determination of the structural enabling bound in terms of an LPP, which is known to be solvable in polynomial time.

For any transition $t \in T$, the computation of the structural enabling bound $SEB(t)$ is formulated in Definition 2.4 in terms of problem (LPP1). Alternatively, we can switch to the dual LPP:

$$\begin{array}{rll} SEB(t) = & \text{minimum} & Y^T \cdot M_0 \\ & \text{subject to} & Y^T \cdot C \leq 0, \quad Y^T \cdot Pre[t] = 1 \\ & & Y \geq 0 \end{array} \tag{LPP6}$$

We recall that MG's are consistent nets with a single minimal T-semiflow which is the vector **1**, so that the constraint $\vec{\sigma} \geq 0$ can be relaxed in the primal problem. The effect on the dual problem of this relaxation is the transformation of the first constraint into $Y^T \cdot C = 0$. In other words, the dual problem for the computation of $SEB(t)$ can be rewritten as follows:

$$\begin{array}{rll} SEB(t) = & \text{minimum} & Y^T \cdot M_0 \\ & \text{subject to} & Y^T \cdot C = 0, \quad Y^T \cdot Pre[t] = 1 \\ & & Y \geq 0 \end{array} \tag{LPP7}$$

This LPP does not require slack variables while this is not the case for (LPP6).

According with the above considerations, the next result follows:

Corollary 4.1 *The lower bound on steady-state throughput for live and bounded MG's given by Theorem 4.2 can be obtained in polynomial time, by computing $LB(t) = SEB(t)$ as the optimum solution of (LPP7), for each transition t of the net.*

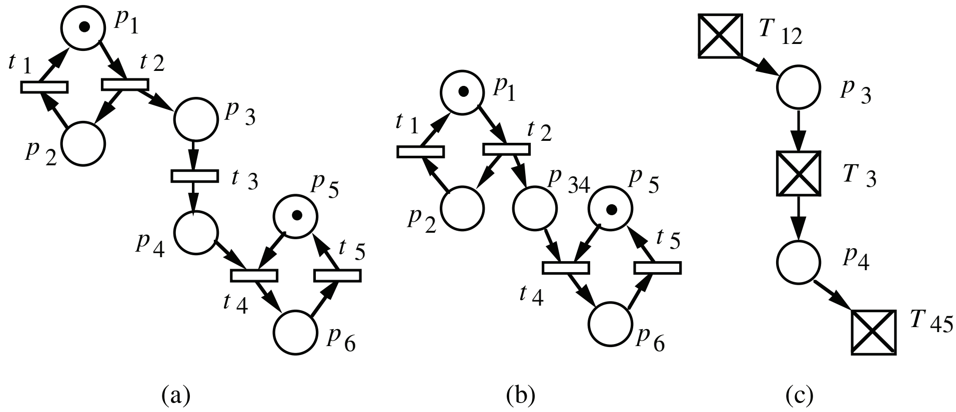


Figure 3: Non-strongly-connected MG's.

For all strongly-connected MG's there exists an elementary P-semiflow for which the optimum of the objective function of (LPP7) is achieved, as shown in Theorem 3.2. In the case of MG's, these elementary P-semiflows can only be circuits, so that we can give the following interpretation of the dual LPP in net terms: *the liveness bound for a transition t of a strongly-connected MG is given by the minimum number of tokens contained in any circuit of places containing transition t.* In a non-strongly-connected MG there may be no such circuit, so that this number can be infinite. The advantage of (LPP7) lies in its compact statement and the polynomial time complexity of its solution. An alternative approach that we do not consider here would be the use of *shortest path* algorithm from graph theory.

As final remarks, we can state the following:

Property 4.3 *Let $\langle \mathcal{N}, M_0 \rangle$ be an MG.*

1) Liveness for $\langle \mathcal{N}, M_0 \rangle$ can be a by-product of a more general (also with polynomial complexity) computation: $\langle \mathcal{N}, M_0 \rangle$ is a live MG $\iff \forall t \in T : \quad SEB(t) > 0$.

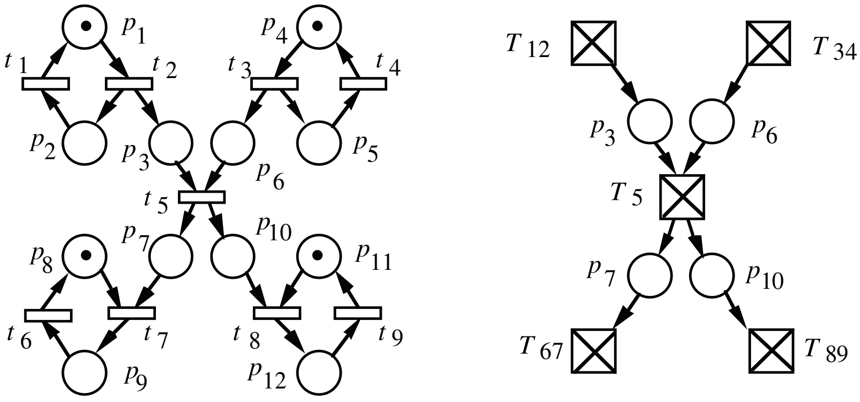
2) If $\langle \mathcal{N}, M_0 \rangle$ is live and $\exists t \in T$ such that $SEB(t) = 1$, then $\forall t' \in T$ belonging to the same circuit characterized by Y in (LPP7), $SEB(t') = 1$.

Note that the application of Property 4.3.2 reduces the computational complexity of the structural enabling bounds of transitions (in the sense that, in many cases, it is not necessary to solve m different LPP's, one for each transition, but a few less).

5 Extending results to non-strongly-connected MG's

In the literature on deterministically-timed MG models, the case of non-strongly-connected nets is usually considered a trivial extension to be left to the imagination of the reader [31, 32]. In this section, we argue that the question is less trivial than one can perceive at first glance, and in fact, we shall derive some examples to show that "direct" extensions of the results obtained in the case of strongly-connected MG's, in general, make no sense. For the upper bound on throughput, we obtain a result similar to that proposed by F. Baccelli et al. [8], even though their work is situated in a quite different framework.

Example 1. Let us first consider, as an example, the non-strongly-connected MG in Figure 3.a. First of all, we can see that transition t_3 has an infinite liveness bound, so that in steady-state it *should not contribute* to the computation of the mean cycle time. Indeed, suppose that t_3 has a deterministic service time of 1000 time units, while transitions t_1 and t_2 have a deterministic service time of 1 time



(a) (b)

Figure 4: A more general non-strongly-connected MG.

unit; thus the circuit t_1–t_2 starts generating tokens at a rate of one token every 2 time units, so that, initially, the tokens accumulate in place p_3. At time 1001, eventually, the first instance of transition t_3 fires, and at that point, we reach the steady-state condition in which 499 instances of firing of t_3 are concurrently enabled, with the remaining enabling time shifted by two time units between each pair of subsequent firing instances. As we can see, the actual firing rate in steady-state for transition t_3 is 1/2 firings per second, i.e., it is determined by the mean cycle time of transitions t_1–t_2, completely independent of the service time of t_3 itself. Therefore, from the steady-state performance point of view, transition t_3 behaves as if it were an *immediate transition*, and it can be reduced by fusing places p_3 and p_4 into a single place p_{34}, as shown in Figure 3.b.

Now let us consider the behavior of the other two transitions t_4 and t_5. Their actual firing rate is determined both by their own service times and the rate with which the circuit t_1–t_2 is able to produce the tokens that are consumed by t_4 from place p_{34}. Thus, the mean cycle time in steady-state condition for transitions t_4–t_5 is given by the maximum of the mean cycle time of t_1–t_2 and the sum of the service times of t_4 and t_5 (this sum would be the mean cycle time of the subnet generated by t_4 and t_5 if it were considered in isolation, i.e., the *potential mean cycle time* of t_4–t_5). In the case in which the mean cycle time of t_1–t_2 were greater than the one of t_4–t_5, the number of tokens at place p_{34} would remain bounded and the firing rate of t_4–t_5 would be the inverse of the mean cycle time of t_1–t_2. On the other hand, in the case in which the mean cycle time of t_1–t_2 were smaller than the one of t_4–t_5, place p_{34} would accumulate tokens and marking process of this place would not be (even weakly) ergodic. However, firing rate of transitions t_4–t_5 would be, in that case, equal to the inverse of their potential mean cycle time. In the case of equality between mean cycle time of t_1–t_2 and t_4–t_5, marking ergodicity at place p_{34} depends on the probability distribution of service time of transitions. In the particular case of deterministic timing, the marking process is weakly ergodic, while in the case of exponentially distributed service times the marking process is non-ergodic (because the embedded Markov process is *null-recurrent*).

Example 2. Let us consider the more general example shown in Figure 4.a. Also in this case it is easy to accept that transition t_5 does not contribute to the steady-state cycle time because it has an infinite liveness bound (it behaves as an immediate transition). However, in this case, we cannot just delete it because of the synchronization constraint due to its multiple input places (p_3 and p_6). On the other hand, it is clear that the two subnets composed of t_1–t_2 and t_3–t_4 behave completely independently of each other and of the rest of the net. If the mean cycle times of these two subnets are not exactly equal (let us assume without loss of generality that the mean cycle time of t_1–t_2 is greater than that of t_3–t_4), then one of the input places of t_5 (p_6 with our assumption) accumulates an infinite number of tokens in steady-state (in other words, the marking process at this place is not ergodic);

thus it becomes redundant (in steady-state) since it cannot constrain the enabling condition of t_5, and it can be deleted without altering the steady-state behavior of the net. In the case of exactly equal mean cycle times of the two subnets (t_1–t_2 and t_3–t_4), marking ergodicity depends on the distribution functions associated with transitions. For instance, for deterministic timing, the marking process at p_3 and p_6 remains bounded (i.e., it is weakly ergodic). On the other hand, for exponential timing, the marking of both places is a null-recurrent Markov process, thus non-ergodic. Deleting all the places that become unbounded in steady-state due to the average transition firing times, we obtain that the net is partitioned into disconnected subnets that can be studied independently of one another. Of course, not only the input but also the output places of t_5 (p_7 and/or p_{10}) may accumulate an infinite number of tokens in steady-state, provided that the potential mean cycle time of their output subnets (respectively, t_6–t_7 and t_8–t_9) are greater than the actual firing time of t_5. In this case, also the output places become redundant and can be deleted, and we may study the steady-state behaviors of the four disconnected subnets in isolation.

From the analysis of the above examples, we can draw two considerations.

First: Marking ergodicity is not assured in the case of non-strongly-connected MG's. Places having non-ergodic marking process can be found among structurally unbounded places, i.e., places do not belonging to any *strongly-connected component* (SCC, in what follows), in two cases: (1) after the comparison between the actual input firing rate and the potential firing rate of the output SCC (Example 1), or (2) after the comparison among the actual firing rate of all SCC's being synchronized by a given transition (Example 2). Some SCC's of the MG can be seen as *producers* of *parts* (or *data*) for other components. Other SCC's act as *consumers* of parts produced by other components. Other may be producers and consumers. Connections among these producers/consumers subsystems are modeled by means of places (or *buffers*). A place is marking ergodic if the throughput of the corresponding producer is less than (or equal to, in the deterministic case) the service rate of the consumer.

Second: There exists a *partial order relation* (POR, for short) "$\succ$" among subsets of transitions defined as: $T_i \succ T_j$ iff the firing delay of transitions in T_i can affect the actual firing rate of transitions in T_j but not vice versa. This POR can be computed by applying a standard algorithm for the derivation of a *condensation* of the original graph, as we explain below.

The previous considerations suggest that the first step that must be taken, in order to check marking ergodicity and to compute actual throughput of transitions, is the construction of the *condensation* of the net. The condensation of a given directed graph [28] represents the interconnections among the SCC's of the original graph. Therefore, the vertices v_i of the condensation correspond with the SCC's C_i of the original one. There is an arc from one vertex v_1 to a different vertex v_2 in the condensation iff there is an arc in the original graph from some vertex in the component C_1 to some vertex in the component C_2.

Definition 5.1 *Let $\mathcal{N}$ be an MG. The MG resulting from $\mathcal{N}$ after the substitution of each SCC C_i by a single transition T_i is called condensation of $\mathcal{N}$, and denoted $\mathcal{N}_c$. There is a place p_{ij} connecting two transitions T_i, T_j in the condensation of an MG ($p_{ij} \in T_i^\bullet \cap {}^\bullet T_j$) iff p_{ij} connects, in the original net, one transition of the SCC associated with T_i with another one of the component associated with T_j ($p_{ij} \in t_{i_1}^\bullet \cap {}^\bullet t_{j_1}$, with $t_{i_1} \in T_i$ and $t_{j_1} \in T_j$).*

The condensation of a directed graph is always a *directed acyclic graph*, because if there were a cycle in it, then all the components in the cycle would really correspond to one SCC in the original graph. An efficient algorithm for the computation of SCC's and the condensation of a directed graph can be found, for instance, in [28].

Now, let us remark that two kinds of transitions can be found in the condensation of a given non-strongly-connected MG: those with infinite liveness bound (corresponding with trivial SCC's having only one transition) and those with finite liveness bound (obtained from the substitution of a non-trivial SCC, i.e., having more than one transition). The first ones have null potential mean cycle time (i.e., infinite throughput if they are considered in isolation), while the potential mean cycle times of the second are always finite.

Figure 3.c represents the condensation of the MG depicted in Figure 3.a. Its transitions can be considered as *complex servers* in a producers/consumers system, from a queueing theory point of view. Transitions T_{12} and T_{45} have finite liveness bound while transition T_3 has infinite liveness bound. Considering the net of Figure 4.a, its condensation is depicted in Figure 4.b, where transitions T_{12}, T_{34}, T_{67}, and T_{89} have finite liveness bound, while the one of T_5 is infinite.

The condensation of a given MG defines a relation on the set of its SCC's:

Definition 5.2 *Let $\mathcal{N}$ be an MG and $\mathcal{N}_c$ its condensation. We denote "$\succ$" the binary relation among transitions of $\mathcal{N}_c$ defined as follows: $T_i \succ T_j$ iff there is a directed path of length one or more from T_i to T_j in $\mathcal{N}_c$ (i.e., T_j can be reached from T_i).*

From previous definition and from the fact that a condensation of a directed graph is always a directed acyclic graph, the next property follows:

Property 5.1 *Relation "$\succ$" is a POR on the set of transitions of the condensation $\mathcal{N}_c$ of an MG, because it is irreflexive and transitive.*

The method for the computation of steady-state throughput of transitions of a non-strongly-connected MG that we present now is based on the previous considerations, using the above defined POR, and considers the liveness bounds of transitions and their potential mean cycle time (i.e., their mean cycle time if they were in isolation). Before the presentation of the computation method we recall the concept of *maximal* element for a POR:

Definition 5.3 *Let $\mathcal{C}$ be a set and "$\succ$" be a POR defined on $\mathcal{C}$. Then, $c \in \mathcal{C}$ is a maximal element of $\mathcal{C}$ for the relation "$\succ$" iff $\nexists c' \in \mathcal{C}$ such that $c' \succ c$.*

For the previously introduced POR on the set of SCC's of an MG, maximal elements are the *source* transitions of the condensation of the graph.

Theorem 5.1 *Let $\langle \mathcal{N}, M_0 \rangle$ be a non-strongly-connected MG with some given average service times associated with transitions. Let $\mathcal{N}_c$ be the condensation of $\mathcal{N}$. Let T_i, $i = 1, \ldots, K$, be a transition of $\mathcal{N}_c$ and $\Gamma_{(i)}^{pot}$ its potential mean cycle time (mean cycle time of the SCC associated with T_i, considered in isolation). The actual mean cycle time $\Gamma_{(i)}$ of T_i, is*

i) If T_i is a maximal element for "$\succ$" then $\Gamma_{(i)} = \Gamma_{(i)}^{pot}$.

ii) If T_i is not a maximal element for "$\succ$", let $\gamma_i = \max\{\Gamma_{(i_1)}, \ldots, \Gamma_{(i_r)}\}$ where $T_{i_j}, j = 1, \ldots, r$, are such that $T_{i_j}^{\bullet} \cap {}^{\bullet}T_i \neq \emptyset$ (thus, in particular, $T_{i_j} \succ T_i$). Then $\Gamma_{(i)} = \max\{\Gamma_{(i)}^{pot}, \gamma_i\}$.

We remark that transitions T_i with infinite liveness bound have null potential mean cycle time ($\Gamma_{(i)}^{pot} = 0$). The exact mean cycle time of transitions can be computed according to the above theorem, starting from the maximal SCC's, which are independent of the others, and then iteratively using the results to solve the subsequent components.

Note that, in practice, the potential mean cycle time of SCC's ($\Gamma_{(i)}^{pot}, i = 1, \ldots, K$) are not known. Moreover, their computation for general distributions is not possible, so far, in polynomial time on the net size. However, the bounds for the mean cycle time of strongly-connected MG's derived in previous sections could be applied for deriving upper and lower bounds for the mean cycle time of transitions in the whole net, *substituting in Theorem 5.1 the exact values $\Gamma_{(i)}^{pot}$ of the mean cycle time of isolated components by their upper and lower bounds, respectively.*

Finally, we remark that, as a by-product of Theorem 5.1, *necessary and sufficient conditions for the marking ergodicity at places can be deduced*. Let us define the input flow at a given place in the condensation of an MG as the actual throughput of its input transition, and the potential service rate of a transition in the condensation as the inverse of its potential mean cycle time. Two cases arise: (1) If a transition of the condensation has only one input place, then the results from queueing theory can be applied (comparing the input flow to the place and the potential service rate of the transition). (2) If a transition of the condensation has several input places, then every place whose input flow is not the minimum is marking non-ergodic; the place with minimum input flow must be studied according with case (1). Some cases are not well-characterized and it depends on the PDF's whether the marking process is ergodic or not. For instance, in the case of deterministic timing, equality between input flow to a place and potential service rate of its output transition assures weak ergodicity. On the other hand, in the case of exponential distributions, such equality leads to non-ergodicity (null-recurrent embedded Markov process).

6 Algorithm for the computation of bounds

In this section, we outline the algorithm to compute the upper and lower bounds for the mean cycle time of MG's, in a step-by-step form.

Input: Let $\langle \mathcal{N}, M_0 \rangle$ be an MG and θ the vector of mean service times of its transitions.

Step 1: Obtain the SCC's $\mathcal{N}_i$, $i = 1, \ldots, k$, and the condensation of the MG (see Definition 5.1) using, for instance, the polynomial time algorithm presented in [28]. For each $\mathcal{N}_i$, let T_i be its corresponding transition in the condensation graph, and label these transitions in such a way that: $\forall T_i, T_j : i < j \Longrightarrow T_j \not\succ T_i$.

Step 2: For each SCC $\mathcal{N}_i$, $i = 1, \ldots, k$, considered in isolation:

1. Solve the linear programming problem (LPP2) (for example, using one of the polynomial algorithms presented in [30]). Let $\Gamma_{(i)}^{min-pot}$ be its optimum value.
2. For each transition t of $\mathcal{N}_i$, solve the linear programming problem (LPP7) (for example, using one of the polynomial algorithms presented in [30]). Let be $\Gamma_{(i)}^{max-pot} = \sum_t (\theta_t / SEB(t))$.

Step 3: For each SCC $\mathcal{N}_i$, $i = 1, \ldots, k$:

a) If T_i is a maximal element for the relation "$\succ$" in the condensation graph (see Definitions 5.2 and 5.3) then:

$$\begin{aligned} \Gamma_{(i)}^{min} &= \Gamma_{(i)}^{min-pot} \\ \Gamma_{(i)}^{max} &= \Gamma_{(i)}^{max-pot} \end{aligned}$$

b) If T_i is not a maximal element for the relation "$\succ$" in the condensation graph, let be $\gamma_i^{min} = \max\{\Gamma_{(i_1)}^{min}, \ldots, \Gamma_{(i_r)}^{min}\}$ and $\gamma_i^{max} = \max\{\Gamma_{(i_1)}^{max}, \ldots, \Gamma_{(i_r)}^{max}\}$ where T_{i_j}, $j = 1, \ldots, r$, are such that $T_{i_j}^{\bullet} \cap {}^{\bullet}T_i \neq \emptyset$ (see Theorem 5.1). Then:

$$\begin{aligned} \Gamma_{(i)}^{min} &= \max\{\Gamma_{(i)}^{min-pot}, \gamma_i^{min}\} \\ \Gamma_{(i)}^{max} &= \max\{\Gamma_{(i)}^{max-pot}, \gamma_i^{max}\} \end{aligned}$$

Output: The lower and upper bounds for the mean cycle time of each SCC, $\mathcal{N}_i$, of the MG are $\Gamma_{(i)}^{min}$ and $\Gamma_{(i)}^{max}$, respectively.

Note that if the input MG of the previous algorithm is strongly-connected, the computation reduces to solve Step 2.1 and Step 2.2 for the unique SCC (the whole net) and to apply Step 3.a. Finally, we remark the polynomial complexity on the net size of all the derived methods.

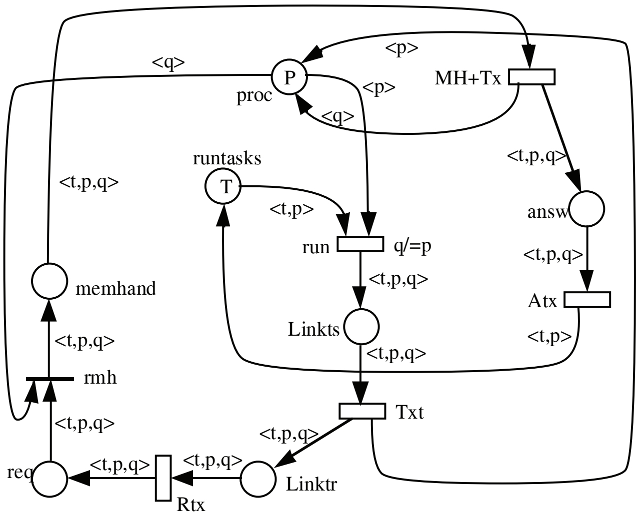


Figure 5: CPN model of two-CPU per processor PADMAVATI architecture.

7 Multiprocessor Architecture Example

As an example of application of the MG's for the performance evaluation of complex multiprocessor computer systems, let us consider a non-trivial model taken from the literature. By taking an already existing example, developed without consideration of the structural restrictions posed by the techniques proposed in this paper, we hope to convince the reader that "in nature" there exist some interesting and nontrivial problems that are amenable to overcome such restrictions. Many other interesting examples can be shown in the fields of computer architecture, communications, and manufacturing systems.

In particular, we consider one of the colored Petri net models of the base software architecture of the PADMAVATI machine, developed in [33]. In that paper, a class of Petri net models was derived directly from a pseudo-code specification of the base software implementing the inter-processor communication software. The models were then completed by adding constraints representing the hardware resources.

We report here in Figure 5 the colored Petri net model in the case of a multiprocessor architecture in which each processor is composed of two Transputer microprocessors, one devoted to the execution of communication and memory handler processes, and the other one devoted to the execution of "client" application tasks. The unfolding of this colored model yields the MG depicted in Figure 6 in case of a two-processor configuration. In [33] it was shown that a "tandem" model composed of only two processors could be used to accurately estimate the performance of a larger multiprocessor configuration, so that the MG model in Figure 6 can be considered as an accurate performance model of the architecture independently of the number of processors.

In the case studied in [33], the evaluation was made before the actual implementation of the prototype of the machine, and the objective of the performance study was the assessment of the effectiveness of multiprogramming in compensating for the large latency of the multistage interconnection network. Only estimates of the average delays of the components (based on their hardware characteristics) were available; no information was, instead, available on the higher moments and on the form of the probability distributions. In the original work, an exponential distribution assumption was adopted in order to apply Markovian analysis techniques, but this choice was clearly arbitrary.

This example represents a classical case in which the computation of performance bounds based

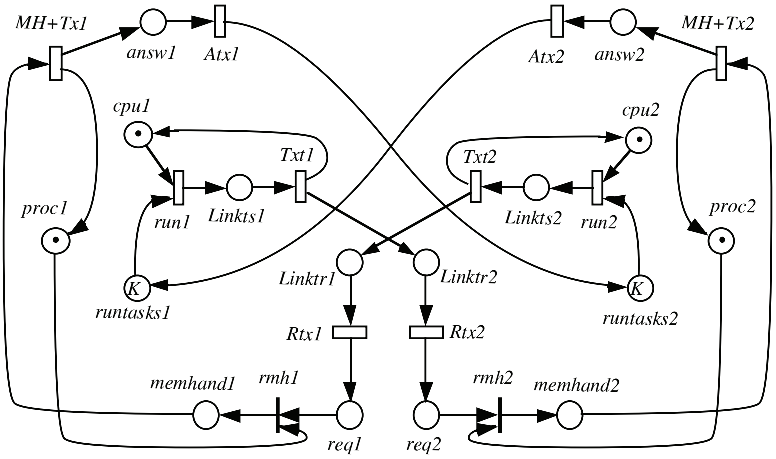


Figure 6: Unfolded MG model of two-CPU per processor PADMAVATI tandem architecture.

K	markings	CPU (sec.)
1	25	< 1
2	196	2
3	900	11
4	3025	47
8	81225	2540

Table 1: Number of reachable markings and CPU time for the computation of the exact values (in seconds, for a SUN 3/60 Workstation), for different number of tasks.

on the assumption that only mean values are known is a good answer to the questions posed by the system designers: the "true" value computed by exact numerical solution of a Markov chain is neither needed nor particularly meaningful in this case.

The obtained results for exponentially-distributed timing of transitions of the model of Figure 6 are summarized in Figure 7. The exact (mean) values, the upper and the lower bounds for the throughput of this MG are superposed, for different values of the mean service time of transitions labeled *run1* and *run2*, and for different number of tasks (K = initial marking of places *runtasks1* and *runtasks2*). See Table 1 for the CPU time measured on a SUN 3/60 Workstation using the *GreatSPN* software package [34] for the analysis and solution of GSPN models. It must be pointed out that, while the bounds can be computed practically in zero time independently of the number of tasks, the results in Table 1 constitute an indicator of the exponential increase with K of the computation time of mean values. Related with the accuracy of the bounds, in the case of only one task ($K = 1$) the lower and the upper bound are trivially equal (thus equal to the exact value). Assuming the mean service times of transitions *run1* and *run2* are equal to 10^{-4} and $K = 2$, the exact (mean) value of the throughput is 7690, while the lower and the upper bounds are 5814 and 9009, respectively, i.e, the exact value is not very close to either of the bounds. For a number of tasks greater than or equal to 8, the exact value coincides with the upper bound (both curves are superposed in the Figure 7). This means that for higher token populations (i.e., under saturation conditions), that are the cases in which the Markovian analysis is practically intractable, the upper bound becomes a very good approximation of

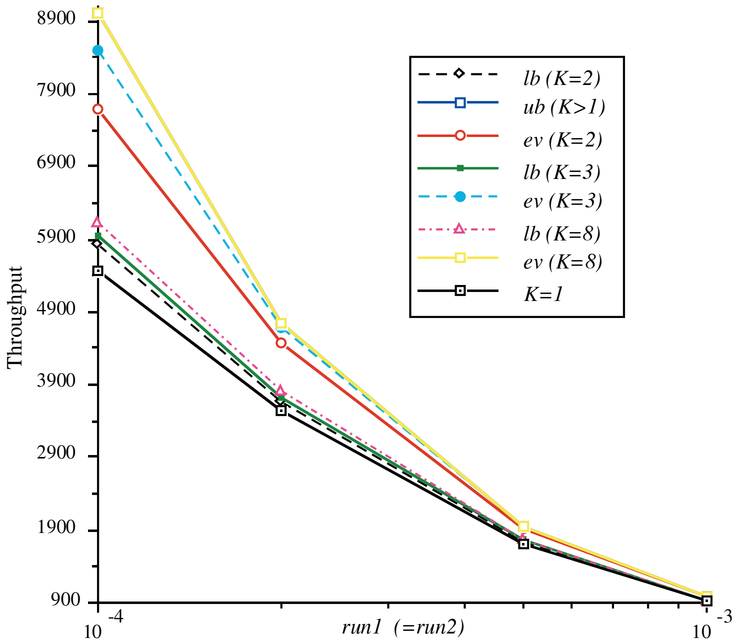


Figure 7: Exact (mean) values, upper and lower bounds for the throughput of the MG in Figure 6 (with exponential timing), for different service time of transitions *run1* and *run2*, and different number of tasks ($K = 1, 2, 3, 8$).

the mean value. In other words, for higher token populations, the maximum parallelism represented in the net system is achieved on the average (this is intuitive, since infinite server semantics is assumed for transitions).

8 Conclusions

In this paper we have addressed the problem of computing upper and lower bounds for the throughput of systems modeled by means of stochastic marked graphs. Both bounds can be computed by means of proper linear programming problems on the incidence matrix of the net, whose solution is known to be of worst-case polynomial complexity. As a by-product, we can characterize the liveness of a marked graph in terms of non-null throughput for all its transitions. This shows an example of possible interleaving between qualitative and quantitative analysis for timed and stochastic Petri nets.

The use of a net formalism can allow to derive reversibility properties similar to those developed in the framework of synchronized queueing networks. It has been pointed out that the reverse net yields the same bounds in the case of strongly-connected marked graphs. This result, which is analogous to one presented in [10], can be considered a generalization of the reversibility property of “tandem finite buffer queues” [35, 36, 37]. This shows an example in which the use of structural considerations on the Petri net structure can easily produce otherwise non-intuitive properties of performance models.

The upper bound on throughput for marked graphs was first proposed by Ramchandani in 1974, and then re-discovered and/or re-interpreted by many others, in the framework of the study of the exact performance of timed Petri nets with deterministic timing. The contributions given by this paper in this sense are three: an alternative reformulation in terms of linear programming problems; the proof that the deterministic case represents an upper bound on performance independently of the probability distribution also in the framework of stochastic Petri nets; the proof that the upper bound is attainable for the mean not only by deterministic but also by stochastic models, with arbitrary values of coefficients of variation.

The lower bounds on throughput presented in this paper as well as the concepts of enabling and liveness bounds for transitions are new. The lower bound on throughput consisting of the inverse of the sum of the firing times of all transitions divided by their respective liveness bounds reduces to the trivial sequentialization of all transitions in the case of safe nets, but has been shown to be attainable with some probability distribution when the coefficient of variation increases. The concept of liveness bound generalizes the usual one of liveness for a transition, and provides another example of possible interleaving between qualitative and quantitative analysis for Petri net models.

Finally, we have taken one performance modeling example from the literature on stochastic Petri nets, showed that it could be restated in terms of a marked graph, and compared numerically our bounds with the exact solution in the case of exponential distribution of firing times. This comparison shows several interesting things: first, the bound computation time was always negligible compared to the Markovian analysis; second, there were cases in which the upper and lower bounds were very close (even identical) to each other, making useless the computation of the "exact" result; third, the upper bound became a very good approximation of the "exact" value for higher token populations (i.e., exactly in the cases in which the Markovian analysis is more expensive or practically intractable).

The computational framework provided by linear programming theory for the performance analysis of marked graphs can be also used for other net subclasses, such as structurally bounded nets with a unique consistent firing count vector [38] or live and bounded free choice nets [9]. Finally, we want to stress the fact that the theoretical results presented in this paper, being easier not only to compute but also to understand and to interpret than classical "exact" ones, can have a substantial impact on the application of performance evaluation techniques in the early design phases of complex distributed systems.

A Appendix

A.1 Proof of Theorem 4.1

By construction, we will show that the association of the family of random variables $x_{\theta_j}^{j-1}(\epsilon)$, defined in (16), with each transition $t_j \in T$ yields exactly the cycle time Γ claimed by the theorem. To give the proof, we will consider a sequence of models ordered by the index of transitions, in which the q-th model of the sequence has transitions $t_1, t_2, \dots t_q$ timed with the random variables $x_{\theta_j}^{j-1}(\epsilon)$, and all other transitions immediate (firing in zero time); the $|T|$-th model in the sequence represents an example of attainability of the lower bound on throughput, independent of the net topology. Now we will prove by induction that the q-th model in the sequence has a cycle time $\Gamma_q = \sum_{j=1}^{q} \theta_j - O(\epsilon)$

Base: $q = 1$: trivial since the repetitive cycle that constitutes the steady-state behavior of the MG contains only one (single-server) deterministic transition with average firing time $\Gamma_1 = \theta_1$.

Induction step: $q > 1$: taking the limit $\epsilon \to 0$, the newly timed transition t_q will fire most of the time with time zero, thus normally not contributing to the computation of the cycle time, that will be just $\Gamma_{q-1} = \sum_{j=1}^{q-1} \theta_j - O(\epsilon)$ (as in the case of model $q-1$) with probability $1 - \epsilon^{q-1}$. On the other hand, the newly timed transition has a (very small) probability ϵ^{q-1} of delaying its firing by a time θ_q/ϵ^{q-1}, which is at least of order $1/\epsilon$ bigger than any other firing time in the circuit, so that in this case all other transitions will wait for the firing of t_q, after having completed their possible current firings in a time which is $O(\epsilon)$ lower than the firing time of t_q itself (i.e., $\theta_q/\epsilon^{q-1} = \Gamma_{q-1}/O(\epsilon)$). Therefore we obtain that $\Gamma_q = (1 - \epsilon^{q-1})\Gamma_{q-1} + \epsilon^{q-1}(\frac{\theta_q}{\epsilon^{q-1}} - O(\epsilon)) = \sum_{j=1}^{q} \theta_j - O(\epsilon)$.

A.2 Proof of Lemma 4.1

To construct an MG of the desired form we can apply the following iterative procedure that interleaves two non-disjoint circuits into a single one. Since the MG is strongly-connected each node belongs to at least one circuit; moreover, since the original MG is finite and each circuit cannot contain the same node more than once, this circuit interleaving procedure must terminate after a finite number of iterations. To reduce the number of circuits, implicit places created after each iteration can be removed. The iteration step is the following:

1. Take two arbitrary non-disjoint circuits (unless the MG already contains a main circuit including all nodes, there always exists such a pair of circuits because the MG is strongly-connected).
2. Combine them in a single circuit in such a way that the partial order among transitions given by the two original circuits is substituted by a compatible but otherwise arbitrary total order. This combination can be obtained by adding new places that are connected as input for a transition of one circuit and output for a transition of the other circuit that we decide must follow in the sequence determined by the new circuit we are creating.
3. Mark the new added places in such a way that the new circuit contains the same number of tokens as the maximum of the number of tokens in the two original circuits.

The above procedure is applied iteratively until all transitions are constrained into a single main circuit. At this point, we can identify and eliminate the implicit places that have been created during the circuits interleaving procedure. We obtain then an MG composed of one main circuit containing $N_M = \max_{t \in T} EB(t)$ tokens that connects all transitions, and a certain number of minor circuits containing less tokens than N_M that maintain the liveness bound of the other transitions.

A.3 Proof of Theorem 4.2

Without loss of generality, assume that transitions in the net resulting from the application of Lemma 4.1 are partitioned in two classes S_2 and S_1, with liveness bounds $K_2 = N_M > 1$ and $K_1 < N_M$, respectively (the proof is easily extended to the case of more than two classes). Construct a new model containing only K_1 tokens in the main circuit; at this point all transitions behave as K_1–servers, so that the cycle time is given by the sum of the firing times of all transitions, divided by the total number of customers in the main loop K_1; moreover, the delay time for the transitions belonging to class S_1 is simply given by $D_1 = \sum_{t_j \in S_1} \theta_j$. Now if we increase the number of tokens in the main loop from K_1 to K_2, the delay time of S_1 cannot increase, so that the contribution of S_1 to the cycle time cannot exceed D_1 for each of the first K_1 tokens. Under the hypothesis that the throughput of the system is given by the inverse of Γ^{max} (i.e., assuming $X = \frac{1}{\Gamma^{max}}$), the average number of tokens of the main loop computed using Little's formula cannot exceed $N_1 = XD_1$, therefore the average number of tokens available to fire transitions in S_2 cannot be lower than

$$N_2 = K_2 - N_1 = K_2 \frac{\frac{K_2 - K_1}{K_1} \sum_{t_j \in S_1} \theta_j + \sum_{t_j \in S_2} \theta_j}{\sum_{t_j \in S_2} \theta_j + \frac{K_2}{K_1} \sum_{t_j \in S_1} \theta_j}$$

On the other hand, we need only

$$N_2 = XD_2 = K_2 \frac{D_2}{\sum_{t_j \in S_2} \theta_j + \frac{K_2}{K_1} \sum_{t_j \in S_1} \theta_j}$$

tokens to sustain throughput X in subnet S_2, so that we are assuming a delay in S_2

$$D_2 \leq \frac{K_2 - K_1}{K_1} \sum_{t_j \in S_1} \theta_j + \sum_{t_j \in S_2} \theta_j$$

Now we claim that this is the actual maximum delay because the first K_1 tokens can proceed at the maximum speed in the whole net, thus experiencing only delay $\sum_{t_j \in S_2} \theta_j$ in subnet S_2, while the remaining $K_2 - K_1$ tokens can also queue up for traveling through S_1, thus experiencing an additional delay of $\frac{1}{K_1} \sum_{t_j \in S_1} \theta_j$ each.

A.4 Proof of Theorem 4.3

We proceed by construction, in a way very similar to that of Theorem 4.1. The only technical difference is that now, without any loss of generality, we assume first of all to enumerate transitions in non-increasing order of liveness bound, i.e., rename the transitions in such a way that $\forall t_i, t_j \in T$, $i > j \implies LB(t_i) \leq LB(t_j)$. Then, as in the case of Theorem 4.1, we can show that the association of the family of random variables $x_{\theta_j}^{j-1}(\epsilon)$ with each transition $t_j \in T$ yields exactly the cycle time Γ^{max} claimed by the theorem. To give the proof we consider a sequence of models ordered by the index of transitions, in which the q–th model of the sequence has transitions $t_1, t_2, \ldots t_q$ timed with the random variables $x_{\theta_j}^{j-1}(\epsilon)$, and all other transitions immediate (firing in zero time); the $|T|$-th model in the sequence represents the resulting model that is expected to provide the example of attainability of the lower bound. By induction we prove that the q-th model in the sequence has a cycle time

$$\Gamma_q = \sum_{j=1}^{q} \frac{\theta_j}{LB(t_j)} - O(\epsilon)$$

Base: $q = 1$: trivial since the repetitive cycle that constitute the steady-state behavior of the MG contains only one ($LB(t_1)$–server) deterministic transition with average firing time $\Gamma_1 = \theta_1 / LB(t_1)$. *Induction step:* $q > 1$: taking the limit $\epsilon \to 0$, each server of the newly timed transition t_q will fire most of the times with time zero, thus normally not disturbing the behavior of the other timed transitions, and not contributing to the computation of the cycle time, that will be just $\Gamma_q = \sum_{j=1}^{q-1} \frac{\theta_j}{LB(t_j)} - O(\epsilon)$ (as in the case of model $q-1$) with probability $1 - \epsilon^{q-1}$. On the other hand, each of the servers of the newly timed transition has a (very small) probability ϵ^{q-1} of delaying its firing of a time $\theta_q / \epsilon^{q-1}$, which is at least order of $1/\epsilon$ bigger than any other firing time in the circuit. Now if $LB(t_q) = 1$, then the proof is completed, since also $\forall j > q$, $LB(t_j) = 1$ by hypothesis, and we reduce to the induction step of the proof of Theorem 4.1. Instead if $LB(t_q) > 1$ then we can consider $LB(t_q)$ consecutive firings of t_q, and compute the average firing time as the total time to fire $LB(t_q)$ times the transition, divided by $LB(t_q)$. Now if we consider m consecutive firings of instances of transition t_q, we obtain an average delay:

$$\sum_{j=0}^{m-1} (1 - \epsilon^{q-1})^j \epsilon^{(q-1)(m-j)} \frac{(m-j)\theta_q}{\epsilon^{(q-1)}} = \theta_q (1 + O(\epsilon))$$

Therefore, the average cycle time of the q–th model will be

$$\Gamma_q = (1 - O(\epsilon^{q-1}))\Gamma_{q-1} + \frac{\theta_q}{LB(t_q)}(1 + O(\epsilon)) = \sum_{j=1}^{q} \frac{\theta_j}{LB(t_j)} - O(\epsilon).$$

References

[1] E. Gelenbe and G. Pujolle. *Introduction to Queuing Networks.* John Wiley & Sons, 1987.

[2] 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.

[3] M. Vernon, J. Zahorjan, and E. D. Lazowska. A comparison of performance Petri nets and queueing network models. In *Proceedings of the 3^{rd} International Workshop on Modeling Techniques and Performance Evaluation*, pages 181–192, Paris, France, March 1987. AFCET.

[4] F. Baccelli and A. Makowski. Queueing models for systems with synchronization constraints. *Proceedings of the IEEE*, 77(1):138–161, January 1989.

[5] *Proceedings of the International Workshop on Timed Petri Nets*, Torino, Italy, July 1985. IEEE Computer Society Press.

[6] *Proceedings of the International Workshop on Petri Nets and Performance Models*, Madison, WI, USA, August 1987. IEEE Computer Society Press.

[7] *Proceedings of the 3^{rd} International Workshop on Petri Nets and Performance Models*, Kyoto, Japan, December 1989. IEEE Computer Society Press.

[8] F. Baccelli, N. Bambos, and J. Walrand. Flow analysis of stochastic marked graphs. In *Proceedings of the IEEE Conference on Decision and Control*, 1989.

[9] J. Campos, G. Chiola, and M. Silva. Properties and performance bounds for closed free choice synchronized monoclass queueing networks. *IEEE Transactions on Automatic Control*, Special Mini-Issue on Modeling and Analysis of Multidimensional Queueing Systems, December 1991.

[10] 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.

[11] J. Campos and M. Silva. Steady-state performance evaluation of totally open systems of Markovian sequential processes. In M. Cosnard and C. Girault, editors, *Decentralized Systems*, pages 427–438. Elsevier Science Publishers B.V. (North-Holland), Amsterdam, The Netherlands, 1990.

[12] K. G. Murty. *Linear Programming.* John Wiley & Sons, 1983.

[13] K. Thulasiraman and M. A. Comeau. Maximum-weight markings in marked graphs: Algorithms and interpretations based on the simplex method. *IEEE Transactions on Circuits and Systems*, 34(12):1535–1545, December 1987.

[14] M. A. Comeau and K. Thulasiraman. Structure of the submarking-reachability problem and network programming. *IEEE Transactions on Circuits and Systems*, 35(1):89–100, January 1988.

[15] M. Silva and J. M. Colom. On the computation of structural synchronic invariants in P/T nets. In G. Rozenberg, editor, *Advances in Petri Nets 1988*, volume 340 of *LNCS*, pages 386–417. Springer-Verlag, Berlin, 1988.

[16] G. Cohen, P. Moller, J. P. Quadrat, and M. Viot. Algebraic tools for the performance evaluation of discrete event systems. *Proceedings of the IEEE*, 77(1):39–58, January 1989.

[17] J. Campos, G. Chiola, J. M. Colom, and M. Silva. Tight polynomial bounds for steady-state performance of marked graphs. In *Proceedings of the 3^{rd} International Workshop on Petri Nets and Performance Models*, pages 200–209, Kyoto, Japan, December 1989. IEEE Computer Society Press.

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

[19] C. Ramchandani. *Analysis of Asynchronous Concurrent Systems by Petri Nets.* PhD thesis, MIT, Cambridge, MA, USA, February 1974.

[20] J. Sifakis. Use of Petri nets for performance evaluation. *Acta Cybernetica*, 4(2):185–202, 1978.

[21] G. Florin and S. Natkin. Les réseaux de Petri stochastiques. *Technique et Science Informatiques*, 4(1):143–160, February 1985. In French.

[22] M. K. Molloy. Performance analysis using stochastic Petri nets. *IEEE Transaction on Computers*, 31(9):913–917, September 1982.

[23] W. M. Zuberek. Performance evaluation using timed Petri nets. In *Proceedings of the International Workshop on Timed Petri Nets*, pages 272–278, Torino,Italy, July 1985. IEEE Computer Society Press.

[24] R. R. Razouk and C.V. Phelps. Performance analysis using timed Petri nets. In *Proceedings of the International Conference on Parallel Processing*, pages 126–129, August 1984.

[25] M. A. Holliday and M. K. Vernon. A generalized timed Petri net model for performance analysis. In *Proceedings of the International Workshop on Timed Petri Nets*, pages 181–190, Torino, Italy, July 1985. IEEE Computer Society Press.

[26] E. Best. Structure theory of Petri nets: The free choice hiatus. In W. Brauer, W. Reisig, and G. Rozenberg, editors, *Advances in Petri Nets 1986 - Part I*, volume 254 of *LNCS*, pages 168–205. Springer-Verlag, Berlin, 1987.

[27] J. Campos, G. Chiola, and M. Silva. Properties and steady-state performance bounds for Petri nets with unique repetitive firing count vector. In *Proceedings of the 3^{rd} International Workshop on Petri Nets and Performance Models*, pages 210–220, Kyoto, Japan, December 1989. IEEE Computer Society Press.

[28] A. Gibbons. *Algorithmic Graph Theory.* Cambridge University Press, London, Great Britain, 1985.

[29] J. M. Colom and M. Silva. Convex geometry and semiflows in P/T nets. A comparative study of algorithms for computation of minimal p-semiflows. In G. Rozenberg, editor, *Advances in Petri Nets 1990*, volume 483 of *LNCS*, pages 79–112. Springer-Verlag, Berlin, 1991.

[30] 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.

[31] C. V. Ramamoorthy and G. S. Ho. Performance evaluation of asynchronous concurrent systems using Petri nets. *IEEE Transactions on Software Engineering*, 6(5):440–449, September 1980.

[32] J. Magott. Performance evaluation of concurrent systems using Petri nets. *Information Processing Letters*, 18:7–13, 1984.

[33] M. Ajmone Marsan, G. Balbo, G. Chiola, and G. Conte. Modeling the software architecture of a prototype parallel machine. In *Proceedings of the 1987 SIGMETRICS Conference*, Banff, Alberta, Canada, May 1987. ACM.

[34] 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.

[35] E. G. Muth. The reversibility property of production lines. *Management Science*, 25(2):152–158, 1979.

[36] G. Yamazaki, T. Kawashiha, and H. Sakasegawa. Reversibility of tandem blocking queueing systems. *Management Science*, 31(1):78–83, 1985.

[37] B. Melamed. A note on the reversibility and duality of some tandem blocking queueing systems. *Management Science*, 32(12):1648–1650, 1986.

[38] 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.