Performance Evaluation of Repetitive Automated Manufacturing Systems

Javier Campos, José M. Colom, and Manuel Silva

Departamento de Ingeniería Eléctrica e Informática *
Universidad de Zaragoza, Spain

Abstract

Steady-state performance evaluation of some repetitive automated manufacturing systems modelled by means of stochastic or deterministic timed Petri nets is considered. Basically, concepts and techniques developed by the authors in other works are applied to repetitive manufacturing systems in this paper. Linear programming problems defined on the incidence matrix of the underlying Petri nets are used to compute tight upper and lower bounds for the performance measures of job-shop systems and decision-free kanban systems in polynomial time on the net structure. The results can be extended to other models in which some decisions are allowed, such as producer-consumer systems with mutual exclusion. Exact performance measures for a class of systems containing sequential processes can be computed in polynomial time.

1 Introduction

Modelling tools for designing complex manufacturing systems must be selected such that qualitative and quantitative analysis can be achieved in an efficient way. A lot of work has been devoted to evaluating the performance measures of such systems. Markov chains and queueing networks models have been used for the exact quantitative analysis (see, e.g., [1]). Approximate methods for intractable large systems have been developed. Perturbation analysis methods [2] are useful simulation techniques which allow sensitivity analysis from the observation of a single sample path.

Moreover, Petri net models have been introduced for designing, validating and computing performance measures of complex manufacturing systems [3, 4, 5, 6]. In these models, a (stochastic or deterministic) timing interpretation is added to the autonomous Petri net scheme as well as a given decision policy for the resolution of conflicts. Petri nets provide *synchronization* capabilities that are not allowed with classical queueing networks. In this sense, stochastic Petri nets can be seen as monoclass queueing networks extended with some synchronization schemes (*synchronized queueing networks* [7, 8, 9, 10]), and the obtained results can be applied to the analysis of such models. Nevertheless, the introduction of these schemes destroys, in general, the local-balance property of product-form queueing networks [11], which has been found only for some net subclasses [12, 13, 14, 15]. Thus, exact computation is only possible in general with a large computational cost, originated from the state space explosion of the embedded Markov chain.

Structural computation (i.e. based on the net structure and not on the state space) of exact measures is only possible for some subclasses of nets (e.g., totally open systems of Markovian sequential processes [10]). For more general net subclasses, very efficient—of polynomial complexity—techniques for computing upper and lower bounds for the throughput of transitions and for the mean marking of places have been developed [7, 8, 9]. In particular, the more restricted subclass for which tight bounds can be computed (marked graphs [16, 17]) allows to generalize the results of the PERT/CPM method to repetitive systems with several resources. An alternative approach for the modelling of those systems can be found in [18, 19]. An extension of these results concerns subclasses of nets in which some decisions can be made that do not change the *relative firing frequences* of transitions (mono-T-semiflow nets). These models and those that are composed by complex servers communicating with buffers allow to analyse more complex manufacturing systems.

We assume the reader is familiarized with the structure, firing rules, and basic properties of net models (see [17] for a nice recent survey). Just for notational purposes, let us state the following: $\mathcal{N} = \langle P, T, Pre, Post\rangle$ is a net with $n = |P|$ places and $m = |T|$ transitions. PRE, $POST$, and $C = POST - PRE$ are $n \times m$ matrices representing the Pre, $Post$, and global incidence functions. A *T-semiflow* is a function (vector) $X{:}\, T \to I\!N$ such that $X \neq 0$ and $C \cdot X = 0$. The *support* of a T-semiflow is defined as $||X|| = \{t \in T | X(t) > 0\}$. A T-semiflow X is *minimal support* iff there exist no other T-semiflow X' such that $||X'|| \subset ||X||$. M (M_0) is a marking (initial marking). Finally, σ represents a fireable sequence, while $\vec{\sigma}$ is the firing count vector associated to σ. If M is reachable from M_0 (i.e. $\exists\sigma$ s.t. $M_0[\sigma\rangle M$), then $M = M_0 + C \cdot \vec{\sigma} \geq 0$ and $\vec{\sigma} \geq 0$.

*E.T.S. Ingenieros Industriales, calle María de Luna 3, ACTUR, 50015 Zaragoza, Spain.

The paper is organized as follows. In Section 2, marked graphs are considered as a modelling and analysing tool for the performance evaluation of manufacturing systems. A summary of efficient algorithms for the computation of bounds for the steady-state cycle time is included. These results can be applied for the evaluation of job-shop systems. In the same way, just-in-time—or kanban—systems performance evaluation can be achieved. Section 3 is devoted to the introduction of decisions in some particular way in the Petri net model. Mono-T-semiflow nets are defined and some results related with the computation of bounds for the performance measures are recalled. In particular, a producer-consumer system with finite store and mutual exclusion is modelled and evaluated. In Section 4, acyclic manufacturing systems with complex servers are modelled with a subclass of stochastic Petri nets. For this subclass, exact computation of the performance indexes of interest can be carried out. Section 5 contains some concluding remarks and considerations on possible extensions of the work.

2 Stochastic marked graphs: a generalization of PERT model

Marked graphs [16, 17] can be seen as a generalization of the classical PERT tool. With PERT, the relationship among the tasks of a project can be represented by a network of activities (arrows) and events (nodes). Timing interpretation can be added to activities for the purpose of evaluating the completion time of the project. The obtained network is an *acyclic* graph, i.e. repetitive systems cannot be modelled.

With marked graphs, *cyclic* behaviours can be modelled as well as many different classes of non shared resources for the realization of activities (tokens at places of the net).

Let us briefly recall what marked graphs are and some of their basic properties. Marked graphs allow to model *concurrency and synchronization but no decisions*: they are structurally (i.e. for all initial marking) decision-free nets.

Definition 2.1 *Marked graphs are ordinary Petri nets (pre and post incidence functions taking values in* $\{0,1\}$*) such that* $\forall p \in P, |{}^\bullet p| = |p^\bullet| = 1$.

Property 2.1 *Let* $\mathcal{N}$ *be a marked graph.*

1. $\mathcal{N}$ *is structurally bounded (i.e.* $\forall M_0$, $\langle \mathcal{N}, M_0 \rangle$ *is bounded) iff it is strongly connected.*

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.*

According to the above properties, strong connectivity and boundedness have equivalent meaning for live marked graphs.

The introduction of (stochastic or deterministic) timing specification is essential if we want to use Petri net models for the performance evaluation of manufacturing 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 places or transitions. Since transitions represent activities that change the state (marking) of the net, it seems natural to associate a duration with these activities (transitions).

Another possible source of confusion in the definition of the timing 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 of tokens which is independent of the total number of them already present in the place, an 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 in a marking should work at conditional throughput 1 or k times that it would work in the case it was enabled only once. In the case of stochastic Petri nets with exponentially distributed firing time associated with transitions, the usual implicit hypothesis is to have "single server" semantics, and the case of "multiple server" is handled as a case of firing rate dependent on the marking; unfortunately this trick cannot work in the case of more general probability distributions, and in particular cannot be used in the case of deterministic timings. This is the reason why people working with deterministic timed transition Petri nets prefer an infinite server semantics. Of course an infinite server transition can always be constrained to a "k–server" behaviour just adding a simultaneously input and output place with k tokens to this transition. Therefore the infinite server semantics appears to be the most general one, and for this reason it will be adopted in this work.

2.1 Bounds for the steady-state performance of marked graphs

In order to be able to speak about steady-state performance we have to assume that some kind of "average behaviour" can be estimated on the long run of the system that we are studying. The usual assumption in this case is that the system model must be *ergodic*, meaning that at the limit when the observation period tends to infinity, the estimates of average values tend (almost surely) to the theoretical expected values of the (usually unknown) probability distributions that characterize the performance indexes of interest.

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. Thus we introduce the concept of *weak ergodicity* that allows the estimation of long run performance also in the case of deterministic models.

Definition 2.2 *The marking process of a stochastic or deterministic marked net is weakly ergodic iff the following limit exists:*

$$\overline{M} \stackrel{\text{def}}{=} \lim_{\tau\to\infty} \frac{1}{\tau}\int_0^\tau M(s)ds < \vec{\infty} \qquad (1)$$

The firing process of a stochastic or deterministic marked net is weakly ergodic iff the following limit exists:

$$\vec{\sigma}^* \stackrel{\text{def}}{=} \lim_{\tau\to\infty} \frac{\vec{\sigma}(\tau)}{\tau} < \vec{\infty} \qquad (2)$$

With the terminology of Petri nets, a home state is a marking that may be reached from all other reachable markings. The existence of home states is closely related with the ergodicity of the marking and firing processes of the timed model.

Property 2.2 *Live marked graphs are reversible nets, i.e. the initial state is a home state, and so all the reachable states are home states.*

Then, for live marked graphs, it makes sense to speak about the unique steady-state behaviour and to compute bounds for the performance of this steady-state.

In [7], the computation of upper and lower bounds for the steady-state performance of (stochastic or deterministic) timed strongly connected marked graphs is studied. In particular, the throughput of transitions, defined as the average number of firings per unit time, is considered.

Let us take into account just the first moments of the probability distribution functions associated with transitions. Let θ be the vector whose components θ_i are the mean values of the random variables associated with the firing of transitions t_i, $i = 1,\ldots,m$. The limit firing count vector per time unit is $\vec{\sigma}^* = \lim_{\tau\to\infty}(\vec{\sigma}(\tau)/\tau)$ and the mean time between two consecutive firings of a selected transition t_i is $\Gamma_i = 1/\vec{\sigma}^*_i$. For strongly connected marked graphs $\Gamma_i = \Gamma$, for all $i = 1,\ldots,m$, and Γ is called the *mean cycle time.*

Theorem 2.1 [7] *Let us consider any live strongly connected marked graph with arbitrary values of mean firing time for transitions, $\theta_i, i = 1,\ldots,m$.*

1. *A lower bound for the mean cycle time is:*

$$\begin{aligned} \Gamma^{lb} = \quad \max \quad & Y^T \cdot PRE \cdot \theta \\ \text{s.t.} \quad & Y^T \cdot C = 0 \\ & Y^T \cdot M_0 = 1 \\ & Y \geq 0 \end{aligned} \qquad \text{(LPP1)}$$

2. *An upper bound for the mean cycle time is*

$$\Gamma^{ub} = \sum_{j=1}^{m} \frac{\theta_j}{L(t_j)} \qquad (3)$$

where $L(t_j)$ is the optimum value of the linear programming problem

$$\begin{aligned} L(t_j) = \quad \min \quad & Y^T \cdot M_0 \\ \text{s.t.} \quad & Y^T \cdot C = 0 \\ & Y^T \cdot PRE[t_j] = 1 \\ & Y \geq 0 \end{aligned} \qquad \text{(LPP2)}$$

Linear programming problems are of worst case polynomial complexity [20], thus the previous bounds can be computed in polynomial time on the size of the net model. Moreover, they depend only on the mean values and not on the higher moments of the probability distributions of the random variables that describe the timing of the system. The independence of the probability distribution can be viewed as a useful generalization of the performance results, since higher moments of the delays are usually unknown for real cases, and difficult to estimate and assess. Moreover both upper and lower bounds, computed by means of proper linear programming problems, are tight, in the sense that for any marked graph model it is possible to define families of stochastic timings such that the steady-state performance of the timed models are arbitrarily close to either bound:

Theorem 2.2 [7] *Let us consider any marked graph with arbitrary values of mean firing time for transitions.*

1. *For any value of the variances of transition firing time, the lower bound for the mean cycle time obtained from the problem* (LPP1) *cannot be improved.*
2. *The upper bound for the mean cycle time obtained in Theorem 2.1.2 cannot be improved.*

As particular interesting cases it must be pointed out that: Γ^{lb} is reached under deterministic (i.e. null variance) timing, while the actual mean cycle time tends to the value of Γ^{ub} if random variables with variances tending to infinity are conveniently selected (see [7]) for the timing of transitions.

Under deterministic timing assumption, the resolution of the problem (LPP1) gives not only the actual cycle time, but a *bottleneck* circuit of the net. This is not the case for general timing, since the bottleneck circuit for which the lower bound is reached can be different from the actual bottleneck circuit.

As a by-product, the liveness of a marked graph in terms of non-null throughput for all its transitions can be characterized. This shows an example of possible interleaving between qualitative and quantitative analysis for (stochastic or deterministic) timed Petri nets.

Theorem 2.3 [7] *A strongly connected marked graph is live if and only if the optimum value of the problem* (LPP1) *is finite.*

Linear programming problems give an easy way to derive results and interpret them. Just looking at the expressions of Γ^{lb} and Γ^{ub} in Theorem 2.1, two reasonable *monotonicity* properties are obtained: decreasing θ (i.e. faster processing elements) or increasing M_0 (i.e. more resources) does not increase the values of Γ^{lb} and Γ^{ub}:

Property 2.3 *Let $\mathcal{N}$ be a strongly connected marked graph and θ the mean firing time vector.*

1. *For a fixed M_0, if $\theta' \leq \theta$, $\theta' \neq \theta$, then the lower (upper) bound for the mean cycle time of $\langle \mathcal{N}, M_0', \theta' \rangle$ is smaller than or equal to (smaller than) the one of $\langle \mathcal{N}, M_0, \theta \rangle$ (i.e. $\Gamma^{lb'} \leq \Gamma^{lb}$ and $\Gamma^{ub'} < \Gamma^{ub}$).*

2. *For a fixed θ, if $M_0' \geq M_0$, $M_0' \neq M_0$, then both the lower and upper bounds for the mean cycle time of $\langle \mathcal{N}, M_0', \theta \rangle$ are smaller than or equal to the ones of $\langle \mathcal{N}, M_0, \theta \rangle$ (i.e. $\Gamma^{lb'} \leq \Gamma^{lb}$ and $\Gamma^{ub'} \leq \Gamma^{ub}$).*

The next property is strongly related to the reversibility of live marked graphs (Property 2.2).

Property 2.4 *For live strongly connected marked graphs, the bounds obtained in Theorem 2.1 do not change for any reachable marking, considered as the initial one.*

The problem of minimizing the resources in a given manufacturing system for obtaining the same upper bound on throughput of productivity can also be studied. Let us consider the dual problem of (LPP1):

$$\begin{aligned} \Gamma^{lb} = \quad & \min \quad \gamma \\ & \text{s.t.} \quad C \cdot z + \gamma M_0 \geq PRE \cdot \theta \end{aligned} \tag{LPP3}$$

Then, for a given cost function w for the amount of resources (e.g., the marking weighted with a cost vector W), the initial cost $w(M_0)$ can be minimized without increasing the lower bound Γ^{lb} of the mean cycle time by solving the following problem:

$$\begin{aligned} \min \quad & w(M_0') = W^T \cdot M_0' \\ \text{s.t.} \quad & C \cdot z + \Gamma^{lb} M_0' \geq PRE \cdot \theta \\ & M_0' \leq M_0 \\ & M_0' \geq 0 \end{aligned} \tag{LPP4}$$

The restriction $M_0' \leq M_0$ is introduced because, otherwise, a reachable marking from M_0 (with less number of tokens that M_0) could have been obtained as optimum solution, when, in fact the amount of resources is the same for all reachable markings (and the bounds do not change considering any of them as initial marking, see Property 2.4). This restriction could be deleted if consistency of W is assumed (i.e. $W^T \cdot C = 0$, then $w(M) = w(M_0)$, for all M reachable from M_0). In general, the optimum solution for this problem is non-integer, therefore classical techniques for finding the optimum integer solution could be applied

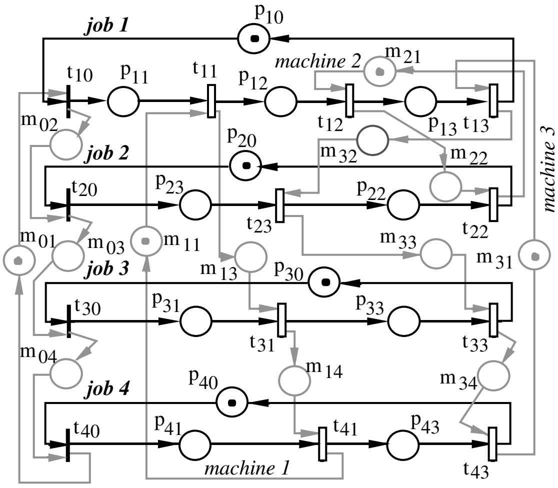


Figure 1: A job-shop system modelled with a marked graph [22].

[21]. In any case, the optimum value of the objective function (in the non-integer case) is a lower bound for the cost of resources for which a given throughput is obtained.

The problem of minimizing initial cost without increasing the upper bound Γ^{ub} of the mean cycle time, can be also considered. In this case, due to the non-linear expression of this bound, only a partial minimization can be expressed in terms of a single linear programming problem. Taking into account that the computation of $L(t_j)$ can be formulated in terms of the dual problem of (LPP2):

$$\begin{aligned} L(t_j) = \quad & \max \quad k \\ & \text{s.t.} \quad M_0 + C \cdot X \geq kPRE[t_j] \end{aligned} \tag{LPP5}$$

We can consider the problem of minimizing the initial cost without decreasing none of the values of $L(t_j)$, as follows:

$$\begin{aligned} \min \quad & w(M_0') = W^T \cdot M_0' \\ \text{s.t.} \quad & M_0' + C \cdot X \geq L(t_j) PRE[t_j], \quad \forall t_j \in T \end{aligned} \tag{LPP6}$$

As for the problem (LPP4), integer programming techniques could be applied for assuring the integrality of the solutions.

2.2 Analysis of job-shop systems

In a job-shop system, a production route through a sequence of machines is carried on for each job. The set of different products as well as the sequences of visits to machines must be completely defined.

In [22], performance evaluation of such systems modelled with marked graphs is studied under a deterministic assumption of the time spent by the jobs on the machines. Using the results presented in the previous section, the deterministic assumption can be relaxed and reachable bounds for the performance of stochastic models can be computed in polynomial time, from the knowledge of the mean values of the duration of jobs.

Let us consider, for instance, the model depicted in Figure 1. A job-shop system with three machines and four jobs is considered (see [22]). The routings of jobs through the machines are modelled with the horizontal circuits, *job 1*: $\langle p_{10}t_{10}p_{11}t_{11}p_{12}t_{12}p_{13}t_{13}p_{10}\rangle$; *job 2*: $\langle p_{20}t_{20}p_{23}t_{23}p_{22}t_{22}p_{20}\rangle$; *job 3*: $\langle p_{30}t_{30}p_{31}t_{31}p_{33}t_{33}p_{30}\rangle$; *job 4*: $\langle p_{40}t_{40}p_{41}t_{41}p_{43}t_{43}p_{40}\rangle$. Since each machine is assumed to process only one job at a time, other circuits are added which determine the sequencing of the jobs on the corresponding machines; *mach. 1*: $\langle m_{11}t_{11}m_{13}t_{31}m_{14}t_{41}m_{11}\rangle$; *mach. 2*: $\langle m_{21}t_{12}m_{22}t_{22}m_{21}\rangle$; *mach. 3*: $\langle m_{31}t_{13}m_{32}t_{23}m_{33}t_{33}m_{34}t_{43}m_{31}\rangle$. They are marked with a token that represents the availability of the machine to process a job.

Let us suppose that only mean values of the processing time associated with the machines have been estimated, as follows: $\theta_{11}=1$; $\theta_{12}=3$; $\theta_{13}=3$; $\theta_{23}=1$; $\theta_{22}=2$; $\theta_{31}=2$; $\theta_{33}=1$; $\theta_{41}=2$; $\theta_{43}=1$. Transitions t_{10}, t_{20}, t_{30}, and t_{40} are immediate (time duration equal to zero), since they account for the loading of the job into the system.

The bounds presented in the previous section can be computed for this model. The optimum value of the problem (LPP1) is $\Gamma^{lb}=9$, which is the cycle time for the slowest elementary circuit: $\langle m_{21}t_{12}p_{13}t_{13}m_{32}t_{23}p_{22}t_{22}m_{21}\rangle$. The upper bound for the mean cycle time which follows from Theorem 2.1.2 is $\Gamma^{ub}=16$ (in this case, the sum of the mean firing time, because $L(t)=1$ for all transition t). The lower bound for the mean cycle time can be reached, for example, if deterministic timing is assumed (null coefficient of variation for the random variables which define the timing of transitions). On the other hand, the mean cycle time tends to the value of the upper bound if random variables with variances tending to infinity are conveniently selected (see [7]) for the timing of transitions. If exponentially distributed random variables (coefficients of variation equal one) are associated with transitions, the actual value for the mean cycle time is $\Gamma=9.985$, which is quite close to the lower bound (reached with coefficients of variation equal zero).

2.3 Analysis of kanban systems

The just-in-time philosophy for the control of manufacturing systems consists of producing just the needed parts at each production stage and at just the right time. Kanban control is a way to implement a just-in-time manufacturing system.

A kanban is a ticket that accompanies a part through the several stages of the production system (see Figure 2.a). When a part of a given stage is consumed by the succeeding stage, the ticket is sent back to trigger the production of a new part. The inventory of a given stage is controlled by the number of kanban tickets at this stage.

In [23], Petri nets have been shown to be well adapted to provide a unified modelling of kanban systems. Most models encountered in literature can be easily represented by marked graph models.

The steady-state performance results presented in Section 2.1 can be applied for analysing quantitatively these models. Without any assumption on the probability distributions associated with transitions, just using their mean values, reachable bounds for the measures of interest can be computed in polynomial time on the net structure.

Let us consider, as an example, the kanban system modelled with a marked graph of Figure 2.b. We are interested on the computation of the average processing time of a whole part in steady-state, provided that both demands and materials exist for the continual production. In other words we consider the computation of the mean cycle time of the subsystem in which places m and d have been deleted. Assume that mean values of random variables associated with transitions are: $\theta(t_{p1})=2$; $\theta(t_{p2})=5$; $\theta(t_{p3})=3$. Transitions t_{w0}, t_{w1}, t_{w2}, and t_{w3} are immediate. Infinite-server semantics is assumed for transitions, and this means that each one of the machines modelled with transitions t_{p1}, t_{p2}, and t_{p3} can process two parts simultaneously (if markings of places c_1, c_2, and c_3 equals 2).

For the marking depicted in Figure 2.b, the lower bound for the cycle time (which is reached for deterministic timing) is $\Gamma^{lb}=2.5$, the inverse of the throughput of transition t_{p2}. This transition models the *bottleneck* machine of the system, and the *utilization* of this machine is 1 (always busy).

For the same number of kanban tickets at each stage, the problem of minimizing resources at places c_1, c_2, and c_3 (capacity of machines), preserving Γ^{lb}, can be considered using (LPP4). The result is that the number of tokens at place c_1 can be reduced to 1 without modifying the bound for the mean cycle time.

On the other hand, if the optimization cost criterion consists of reducing as much tokens as possible (i.e. both kanban tickets and capacity of machines), the resolution of problem (LPP4) gives that the capacity of machine 1 (marking of place c_1) can be reduced to 1, and the number of kanbans at stages 1 and 2, can be reduced to 1 and 2, respectively (instead of 2 and 3, as is depicted in Figure 2.b), without changing the bound for the mean cycle time. In fact the optimum real solution of the problem (LPP4) for $W=\vec{1}$, says that both the capacity of machine 1 and the number of kanbans at the first stage can be reduced to 0.8 units. For the capacity of machine 3 and for the number of kanbans at the third stage, the optimum value is 1.2. In this case, the optimum integer solution is just the excess round of the optimum real solution.

It can be pointed out that for the deterministic timing case the previous minimization of resources preserves the actual mean cycle time. This is not the case for general distribution timing (with non null coefficient of variation). For example, for exponentially distributed timing

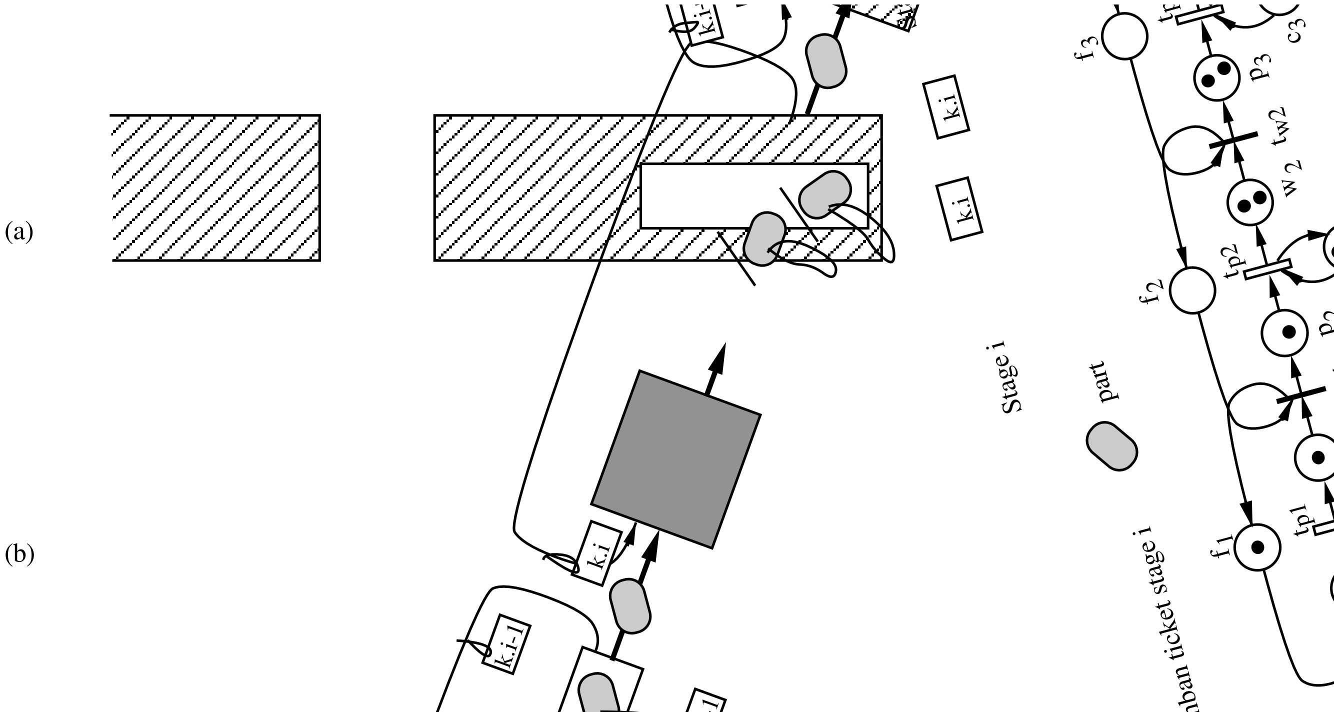


Figure 2: A kanban system and its marked graph representation [23].

of transitions, the actual mean cycle time for the initial marking depicted in Figure 2.b) is $\Gamma = 2.674$. However, considering as initial marking the one which minimizes the resources (both capacities and kanbans) the actual mean cycle time is $\Gamma' = 3.290$ (i.e. greater than Γ). The result, $\Gamma = 2.674 < \Gamma' = 3.290$, is easily explained by the uncertainty introduced by the stochastic assumption (non-null coefficients of variation).

3 Introducing decisions

Structurally speaking, a strongly connected marked graph has only one possible infinite behaviour defined by its T-semiflow. But marked graphs are structurally decision-free systems. In this section, a net subclass more general than marked graphs, with only one T-semiflow but introducing decisions, is considered in the context of repetitive automated manufacturing systems.

In [8], mono-T-semiflow nets are introduced as a structurally defined subclass of nets with a unique repetitive firing count vector:

Definition 3.1 *A structurally bounded Petri net $\mathcal{N}$ is called mono-T-semiflow iff there exists a unique minimal T-semiflow that contains all transitions.*

In a mono-T-semiflow net, conflicts may be reached, and so different behaviours can occur. However, the same set of transitions is fired in all different behaviours, perhaps in a different order. Therefore, from the steady-state performance point of view, the decisions lead to a unique result (provided that the net is live).

3.1 Computing bounds for performance measures

Unfortunately, the existence of a unique minimal T-semiflow containing all transitions does not assure the ergodicity of the marking process (in general, mono-T-semiflow nets may have no home state [8], so ergodicity is not assured).

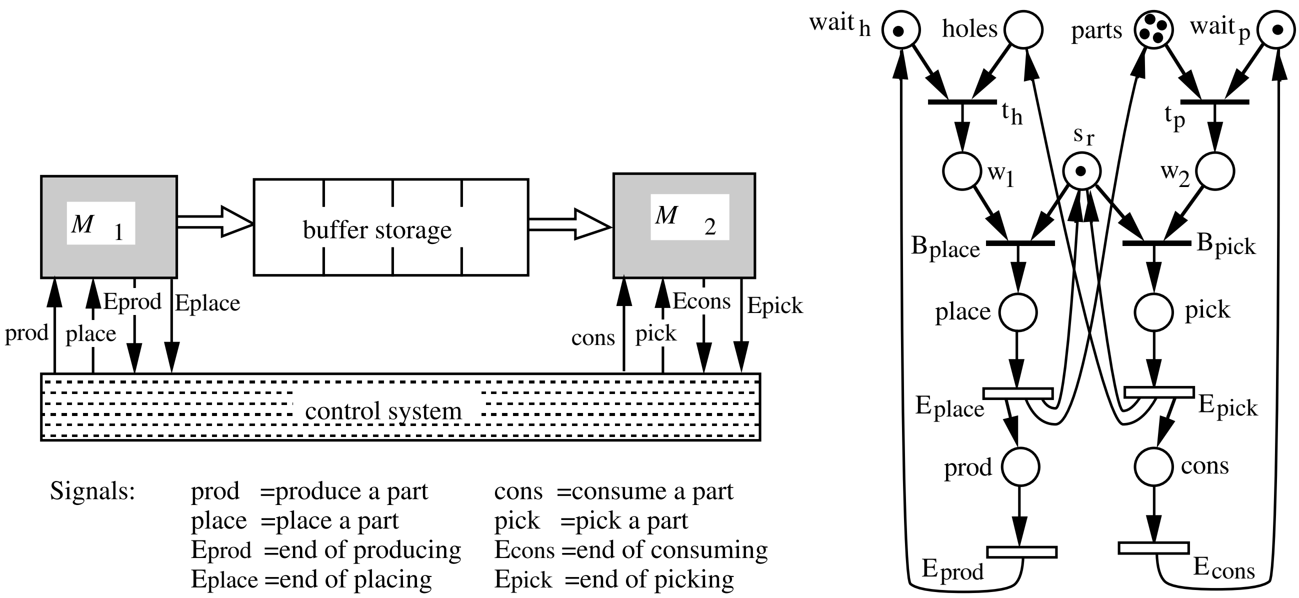
Even in the case in which ergodicity is not assured the problem of computing bounds for the throughput makes sense. The computed values are bounds for all possible steady-state behaviours of the net.

Theorem 3.1 [8] *A lower bound for the mean cycle time of a given transition t_i (taken as the reference station) of a mono-T-semiflow net can be obtained by solving the following linear programming problem:*

$$\begin{array}{rll} \Gamma_i^{lb} = & \max & Y^T \cdot PRE \cdot D \cdot X \\ & \text{s.t.} & Y^T \cdot C = 0 \\ & & Y^T \cdot M_0 = 1 \\ & & Y \geq 0 \end{array} \qquad \text{(LPP7)}$$

where D is the diagonal matrix of mean values of the random variables associated with transitions and X is the minimal T-semiflow with $X(t_i) = 1$ (which can be computed in polynomial time).

The optimum value of the previous problem is a non reachable bound in general (i.e. there exist net models such that neither stochastic nor deterministic interpretation allows to reach the computed bound, Γ_i^{lb}). Moreover, a mono-T-semiflow net can be non live and the obtained lower bound for the cycle time be finite. In other words:



(a) (b)

Figure 3: A producer-consumer system and its mono-T-semiflow net representation [24].

Theorem 3.2 [8] *For mono-T-semiflow nets, liveness is not characterized by the finiteness of the lower bound for the mean cycle time computed by means of the problem* (LPP7).

Concerning the upper bound for the mean cycle time, at the moment only the trivial one can be computed (the sum of all transitions firing time weighted by the T-semiflow vector), provided that the net is live.

3.2 A producer-consumer system

Let us consider the problem of modelling and evaluating a producer-consumer system composed by two machines and a buffer storage (Figure 3.a) [24]. The machine M_1 produces parts that are placed at the buffer. The maximum capacity of the buffer is four parts. The machine M_2 picks parts from the buffer for processing them. The control system for the production and consumption of parts is depicted in Figure 3.b by means of a Petri net. Machines M_1 and M_2 cannot operate simultaneously with the buffer, i.e. the pick and place operations are in mutual exclusion (modelled with place s_r).

Obviously, the net in Figure 3.b is not a marked graph. Transitions B_{place} and B_{pick} can be in an effective conflict. The net is mono-T-semiflow (the unique minimal T-semiflow is the vector with all components equal 1), and the results presented in the previous section can be applied. They allow to compute in polynomial time on the net size, performance bounds for the throughput of transitions in steady-state.

Let us suppose that transitions t_h, t_p, B_{place}, and B_{pick} are immediate and that the mean values of random variables associated with the rest of transitions are: $\theta(E_{place}) = 2$; $\theta(E_{pick}) = 3$; $\theta(E_{prod}) = 4$; and $\theta(E_{cons}) = 2$.

The problem (LPP7) gives a lower bound for the mean cycle time of the net: $\Gamma^{lb} = \max\{\theta(E_{place}) + \theta(E_{pick}), (\theta(E_{place}) + \theta(E_{pick}))/4, \theta(E_{place}) + \theta(E_{prod}), \theta(E_{pick}) + \theta(E_{cons})\} = 6$. As it is remarked in the previous section, this bound is non-reachable, in general. However, in this case, the lower bound is equal to the actual cycle time for deterministic timing. In fact, if deterministic timing is considered, the buffer storage capacity (initial number of tokens at place *parts*) can be reduced to 1, without modifying the actual mean cycle time. This is because, in this particular case, there exist two different P-semiflows Y_1 and Y_2 (with $||Y_1|| = \{s_r, place, pick\}$ and $||Y_2|| = \{holes, parts, w_1, w_2, s_r, place, pick\}$) involving the same set of timed transitions (E_{place} and E_{pick}). Since $Y_1^T \cdot M_0 = 1$, then the number of tokens at place *parts* can be reduced to 1, and the same optimum value in problem (LPP7) is preserved.

As for the marked graph case, the minimization of tokens preserving the lower bound for the mean cycle time does not preserves the actual mean cycle time for general (non deterministic) timing. For example, considering exponentially distributed timing for the net in Figure 3.b, the actual mean cycle time decreases if the initial number of tokens at place *parts* (capacity of the storage) increases. In fact, the increase of the cycle time is stopped when the capacity of storage makes insignificant the portion of time during which the machines are waiting for a hole at the buffer or for a part (see Figure 4).

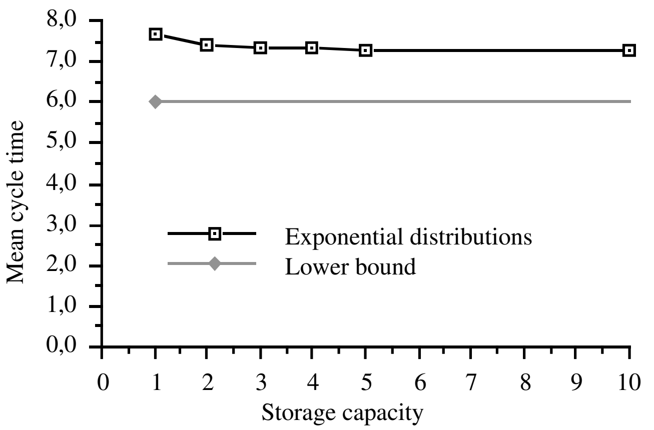


Figure 4: Mean cycle time for different storage capacities, under exponentially distributed timing.

4 A class of buffer acyclic manufacturing systems with complex servers

In this section, let us consider a particular class of manufacturing systems that can be modelled and evaluated by means of totally open systems of Markovian sequential processes [10]. This subclass of stochastic Petri nets can be viewed as a generalization of a subclass of queueing networks in which complex sequential servers can be synchronized according to some particular schemes. On the one hand, totally open systems of Markovian sequential processes cannot model buffer cyclic systems, and in this sense they are more restrictive than marked graphs. Nevertheless, the complex sequential servers admit the introduction of decisions. In this sense, this subclass of nets allows the modelling of systems which cannot be described with marked graphs.

Some interesting qualitative properties of totally open systems have been studied in [10] making special emphasis in those that assure the possibility of ergodic behaviour of the system:

Theorem 4.1 [10] *Necessary and sufficient conditions for the ergodicity of totally open systems of Markovian sequential processes can be computed in polynomial time on the net size.*

Let us remark that the complexity of the computation of conditions is polynomial on the net structure size and not on the state space size (which is infinite, in fact).

The most important result for this subclass of stochastic Petri nets is that, under ergodicity assumption, the exact steady-state performance measures can also be computed in polynomial time:

Theorem 4.2 [10] *If a totally open system of Markovian sequential processes is ergodic then the steady-state throughput of transitions can be computed in polynomial time on the net size.*

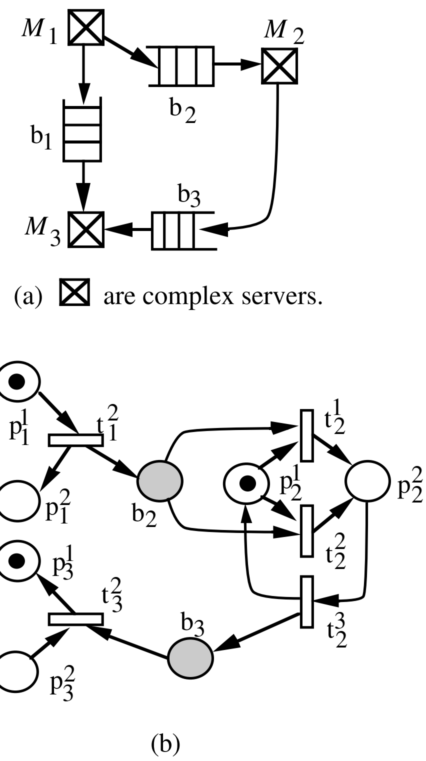


Figure 5: A manufacturing open system and its Petri net representation.

Let us consider a manufacturing system composed by three sequential machines M_1, M_2, M_3 such that M_3 needs two classes of parts that are produced by M_1 and M_2; and M_2 consumes other kind of parts produced by M_1 according to a given synchronization scheme (see Figure 5.a). The Petri net representation is depicted in Figure 5.b. The machines are modelled with one token marked state graphs (i.e. Petri nets that can model decisions but neither synchronizations nor concurrency) while the shaded places are "unbounded" stores between them. The timed model is obtained by adding independent exponentially distributed (constant rate) firing time to transitions. From a queueing network perspective, state graphs represent "complex servers" while stores represent queues.

For this model, the following ergodicity condition can be computed in polynomial time on the net structure (λ_i^k is the constant rate of the random variable associated with t_i^k):

$$\frac{\lambda_1^1\lambda_1^2}{\lambda_1^1+\lambda_1^2} < \min\left\{\frac{(\lambda_2^1+\lambda_2^2)\lambda_2^3}{\lambda_2^1+\lambda_2^2+\lambda_2^3}, \frac{\lambda_3^1\lambda_3^2}{\lambda_3^1+\lambda_3^2}\right\} \quad (4)$$

This condition assures that for each store the input flow of parts is less than the consuming rate of the output machine.

Under the ergodicity assumption (i.e. if equation (4) holds), the exact steady-state performance measures for machine M_1 can be computed in polynomial time on the net structure (Theorem 4.2). This can be done by solving the following system, in which $\overline{M}(p_i^k)$ represents the limit mean marking of place p_i^k and $\vec{\sigma}^*(t_i^k)$ is the limit firing count vector per time unit of transition t_i^k:

$$\begin{aligned} &\overline{M}(p_1^1) + \overline{M}(p_1^2) = 1 \\ &\vec{\sigma}^*(t_1^1) = \lambda_1^1 \overline{M}(p_1^2); \quad \vec{\sigma}^*(t_1^2) = \lambda_1^2 \overline{M}(p_1^1) \\ &\vec{\sigma}^*(t_1^1) = \vec{\sigma}^*(t_1^2) \end{aligned} \tag{5}$$

The solution of the system is:

$$\begin{aligned} &\overline{M}(p_1^1) = \frac{\lambda_1^1}{\lambda_1^1 + \lambda_1^2}; \quad \overline{M}(p_1^2) = \frac{\lambda_1^2}{\lambda_1^1 + \lambda_1^2} \\ &\vec{\sigma}^*(t_1^1) = \vec{\sigma}^*(t_1^2) = \frac{\lambda_1^1 \lambda_1^2}{\lambda_1^1 + \lambda_1^2} \end{aligned} \tag{6}$$

Now, for computing the steady-state measures of the other state machines under the assumption of ergodicity (equation (4)), it is necessary to take into account that $\vec{\sigma}^*(t_1^2) = \vec{\sigma}^*(t_2^1) + \vec{\sigma}^*(t_2^2)$ and $\vec{\sigma}^*(t_3^1) = \vec{\sigma}^*(t_1^1)$, i.e. the input flow of tokens to each store in steady-state must be equal to the output flow:

$$\begin{aligned} &\overline{M}(p_2^1) = 1 - \overline{M}(p_2^2); \quad \overline{M}(p_2^2) = \frac{\lambda_1^1 \lambda_1^2}{(\lambda_1^1 + \lambda_1^2)\lambda_2^3} \\ &\vec{\sigma}^*(t_2^1) = \frac{\lambda_1^1 \lambda_1^2 \lambda_2^1}{(\lambda_1^1 + \lambda_1^2)(\lambda_2^1 + \lambda_2^2)} \\ &\vec{\sigma}^*(t_2^2) = \frac{\lambda_1^1 \lambda_1^2 \lambda_2^2}{(\lambda_1^1 + \lambda_1^2)(\lambda_2^1 + \lambda_2^2)} \\ &\vec{\sigma}^*(t_2^3) = \vec{\sigma}^*(t_3^1) = \vec{\sigma}^*(t_3^2) = \frac{\lambda_1^1 \lambda_1^2}{\lambda_1^1 + \lambda_1^2} \end{aligned} \tag{7}$$

5 Conclusions

The problem of computing the steady-state performance measures of some classes of automated manufacturing systems modelled with stochastic or deterministic timed Petri nets has been addressed in this paper.

For those systems that can be modelled with marked gaphs, upper and lower bounds for the mean cycle time are obtained. The lower bound is thight in the sense that it is reachable not only by deterministic but also by stochastic models, with arbitrary values of coefficients of variation. The upper bound cannot be improved only with the knowledge of the mean firing time values. This is the case for job-shop systems and most models of kanban systems encountered in literature.

All bounds are 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. The problem of minimizing resources preserving the productivity of the system can be achieved by means of linear integer programming problems.

An extension of the results to other subclasses of nets allows to evaluate models in which some decision schemes are introduced that are not relevant from the performance point of view.

Finally, for a subclass of buffer acyclic manufacturing systems with Markovian timing interpretation, ergodicity conditions and exact steady-state performance measures can be computed in polynomial time on the net structure.

Extensions of the results presented in this paper are already being considered. In particular, reachable upper and lower bounds for the case of stochastic or deterministic timed live and bounded free choice Petri nets have been obtained [9]. Related with the computation of exact measures, work is in progress to generalize the results obtained for open systems of state graphs synchronized with buffers to open systems of bounded nets synchronized in the same way. Partially open systems must be studied. Ergodicity conditions can be computed for the open components of these systems. The computation of the exact performance of closed subsystems seems not to be possible in polynomial time. In any case, efficient techniques for computing tight bounds or approximate values could be applied.

Acknowledgments. The authors wish to express their gratitud to Prof. Giovanni Chiola of the Università di Torino, co-author of some of the theoretical results used in this work [7, 8, 9], for his valuable discussions and cooperation. The exact solutions for exponentially distributed stochastic Petri nets were computed with the aid of *GreatSPN*, a tool for the edition and analysis of timed and stochastic Petri net models, developed by G. Chiola.
This work has been supported by project PA86-0028 of the Spanish *Comisión Interministerial de Ciencia y Tecnología* (CICYT).

References

[1] J. A. Buzacott and D. D. Yao. Flexible manufacturing systems: a review of analytical models. *Management Science*, 32(7):890–905, July 1986.

[2] Y. C. Ho. Performance evaluation and perturbation analysis of discret event dynamic systems. *IEEE Trans. on Automatic Control*, 32(7):563–572, July 1987.

[3] J. Martínez, H. Alla, and M. Silva. Petri nets for the specification of FMSs. In *Modelling and Design of Flexible Manufacturing Systems* (A. Kusiak, ed.), pp. 389–406, Elsevier Science Pub., New York, 1986.

[4] T. Murata and M. Silva (organizers). Invited sessions: Petri nets and flexible manufacturing. In *Proc. of the 1987 IEEE Intern. Conference on Robotics and Automation*, pp. 999–1018 and 1160–1185, Raleigh, North Carolina, March 1987.

[5] A. Desrochers. *Modeling and Control of Automated Manufacturing Systems.* IEEE Computer Society Press, Washington, 1990.

[6] M. Silva and R. Valette. Petri nets and flexible manufacturing. *Research Report GISI-RR-90-3. Dpto. de Ingeniería Eléctrica e Informática, Universidad de Zaragoza*, Spain, January 1990.

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

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

[9] J. Campos, G. Chiola, and M. Silva. Properties and performance bounds for closed free choice synchronized monoclass queueing networks. *Research Report GISI-RR-90-2. Dpto de Ingeniería Eléctrica e Informática, Universidad de Zaragoza*, Spain, January 1990.

[10] J. Campos and M. Silva. Steady-state performance evaluation of totally open systems of Markovian sequential processes. In *Proc. of the IFIP Working Conference on Decentraliced Systems* (C. Girault, ed.), Elsevier Science Pub., Lyon, France, December 1989.

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

[12] M. Ajmone Marsan, G. Balbo, G. Chiola, and S. Donatelli. On the product form solution of a class of multiple-bus multiprocessor system models. *The Journal of Systems and Software*, 1(2):117–124, 1986.

[13] A. A. Lazar and T. G. Robertazzi. Markovian Petri net protocols with product form solution. In *Proc. of the Conference on Computer Communications*, IEEE Computer Society Press, pp. 1054–1062, Washington DC, 1987.

[14] G. Florin and S. Natkin. Matrix product form solution for closed synchronized queueing networks. In *Proc. of the 3^{rd} Intern. Workshop on Petri Nets and Performance Models*, IEEE Computer Society Press, Kyoto, Japan, December 1989.

[15] W. Henderson and P. G. Taylor. Aggregation methods in exact performance analysis of stochastic Petri nets. In *Proc. of the 3^{rd} Intern. Workshop on Petri Nets and Performance Models*, IEEE Computer Society Press, Kyoto, Japan, December 1989.

[16] F. Commoner, A. W. Holt, S. Even, and A. Pnueli. Marked directed graphs. *Journal of Computer and Systems Science*, 5:511–523, 1971.

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

[18] 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.

[19] D. Dubois and K. Stecke. Dynamic analysis of repetitive decision-free discrete event processes: (I) The algebra of timed marked graphs and algorithmic issues; (II) Applications to production systems. *Working papers 542-b and 543-b, School of Business Administration, University of Michigan*, October 1989.

[20] N. Karmarkar. A new polynomial time algorithm for linear programming. *Combinatorica*, (4):373–395, 1984.

[21] R. S. Garfinkel and G. L. Nemhauser. *Integer Programming*, John Wiley & Sons, 1972.

[22] H. P. Hillion and J. M. Proth. Performance evaluation of jobshop systems using timed event-graphs. *IEEE Trans. on Automatic Control*, 34(1):3–9, January 1989.

[23] M. Di Mascolo, Y. Frein, Y. Dallery, and R. David. Modeling of kanban systems using Petri nets. In *Proc. of the 3^{rd} ORSA/TIMS Conference on Flexible Manufacturing Systems* (K. Stecke and R. Suri, eds.), Elsevier Science Pub., pp. 307–312, 1989.

[24] M. Silva. *Las Redes de Petri en la Automática y la Informática.* Editorial AC, Madrid, 1985.