State machine reduction for the approximate performance evaluation of manufacturing systems modelled with cooperating sequential processes

Carlos J. Pérez-Jiménez*, Javier Campos, and Manuel Silva
Departamento de Informática e Ingeniería de Sistemas †
María de Luna 3, E-50015 Zaragoza, Spain
e-mail: cjperez, fjcampos, msilva @mcps.unizar.es

Abstract

We concentrate on a family of discrete event systems obtained from a simple modular design principle that include in a controlled way primitives to deal with concurrency, decisions, synchronization, blocking, and bulk movements of jobs. Due to the functional complexity of such systems, reliable throughput approximation algorithms must be deeply supported on a structure based decomposition technique. We present a decomposition technique and a fixed-point search iterative process based on response time preservation of subsystems. An extensive battery of numerical experiments has shown that the error is less than 3%*, and that the state space is usually reduced by more than one order of magnitude.*

1 Introduction

Performance analysis of *discrete event systems* is an important subject where growing efforts are being devoted. The emerging problems are really complex if general models are considered, including *concurrency*, *decisions*, *synchronization*, *blocking*, and *bulk movements* of jobs.

Deterministic Systems of Sequential Processes (in the sequel DSSP) are obtained by the application of a simple modular design principle: several functional units (in this case, sequential processes modelled with *state machines*, SM's) execute concurrently and cooperate using asynchronous communication by message passing through a set of buffers (places with possibly weighted input and output arcs). Buffers are destination private (i.e., they go to a single sequential process). Thus competition among functional units is prevented. Moreover, buffers do not represent side conditions in conflicts of the functional units (i.e., choices are free). Several classes have been defined in the literature following this basic modular design principle [Rei82, Sou93, TSCC95, RTS95]. In this paper, we deal with approximate throughput computation of DSSP's, that include, in a controlled way, all the above enumerated features.

The approximation technique is based on *SM's reduction*, so it is a *local* aggregation technique and it preserves all the connections among functional units (i.e., it preserves the buffers and their input and output transitions). It is specially suited for DSSP's composed from large SM's. A cut is defined through buffers producing a partitioning the SM's. Per each of these partition an aggregated subsystem is obtained by reducing as much as possible all the SM's belonging to the rest. The method for the reduction of the tangible reachability set of SM's that we present is such that in the aggregated subsystems the reachable markings and firing sequences are projections of those in the original system (thus, other logical properties like steps, boundedness, liveness and the existence of home states are also preserved).

The approximation technique has two steps:
(1) *A structure based decomposition of the original model leading to two or more smaller aggregated subsystems* using the SM's reduction technique.
(2) *An iterative response time approximation method for the computation of the throughput.* At each iteration step, the underlying CTMC of every aggregated subsystems are solved, getting more information for the improvement of the response time approximation of the reduced nodes. A *basic skeleton* of the system is used in order to tune up the response time approxima-

*The work by this author is supported by a grant of the CONAI, the Aragonese Research Council.
†This work has been developed within the projects PRONTIC 94-0242 of the Spanish CICYT, and ESPRIT BRA 7269 (QMIPS) and HCM CT94-0452 (MATCH) of the European Community.

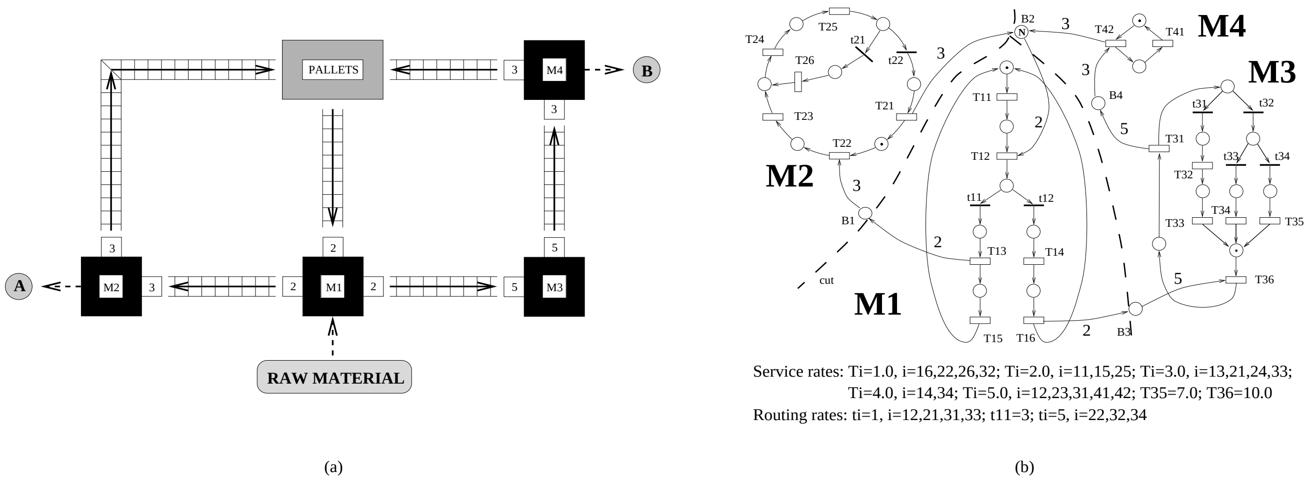

Figure 1: (a) Manufacturing system and (b) its DSSP model. (N: Number of pallets).

tion. An extensive battery of numerical experiments has shown that this is a very robust and fast approximation technique.

The method will be applied to the manufacturing system depicted in Fig. 1.a. The process plans are:(A) machines 1,2;(B) machines 1, 3, 4. Machine 1 can produce two different partial products, one of them for final product A and the other one for product B. Each machine needs a minimum quantity of its initial product per operation. This quantity is measured by the number of pallets needed to transport it. For example, machine 1 needs two pallets of raw material to work, machine 2 needs three pallets, etc. The raw material comes into the system throughout machine 1 and the final products leave the system throughout machine 2 (kind A) or machine 4 (kind B).

We are going to approximate the performance of the system varying the number of pallets that are used.

A net model for the system is that in Fig. 1.b. The complex behaviour of each machine is modelled with a safe SM. Buffer $B2$ contains N tokens, the total number of pallets in the system (a parameter). Weights of the input and output arcs of the buffers model the number of pallets needed for a machine to work. The minimum number of N for getting a live system is 10.

The paper is organized as follows. The formal definition of DSSP's and the considered stochastic interpretation are introduced in Section 2. Section 3 is devoted to the structure-based decomposition technique. The iterative response time approximation method for the computation of the throughput is explained in Section 4. In Section 5, we present some numerical results for the manufacturing example. Some concluding remarks are outlined in Section 6.

2 The model

In this section we formally define the class of Deterministic Systems of Sequential Processes as a subclass of Petri net systems, and we explain how time is introduced in the model. We assume the reader is familiar with concepts of P/T nets. For further extensions the reader is referred to [Mur89, Sil93]. In this paper we use the same notation of that in [CCJS94]. The net class that we consider is an extension of that in [Sou93] which itself was an extension of that in [Rei82], although we keep the same name. In [Sou93], sequential processes are modelled with *safe state machines* while the communication among them is described by their connection through particular places called *buffers*: the buffers are private in the sense that each of them has only one input and only one output state machine. Our extension allows that several state machines deposit messages (tokens) in a common buffer.

2.1 DSSP's

State machines are ordinary Petri nets such that every transition has only one input and only one output place ($\forall t \in T$: $|^{\bullet}t| = |t^{\bullet}| = 1$). SM's allow the modelling of sequences, decisions (or conflicts), and re-entrance (when they are marked with more than one token) but not synchronization.

DSSP's are used for the modelling and analysis of distributed systems composed by sequential processes communicating through output-private buffers. Each sequential process is modelled by a *safe* (1–bounded) SM. The communication among them is described by *buffers* (places) which contain *products/messages* (tokens), that are produced by some processes and consumed by others. Each buffer is *output-private* in the sense that it is an input place of only one SM.

Definition 2.1 *A marked Petri net* $(\mathcal{N}, M_0) = (P_1 \cup \ldots \cup P_q \cup B, T_1 \cup \ldots \cup T_q, Pre, Post, M_0)$ *is a* Deterministic System of Sequential Processes *if:*
1. $P_i \cap P_j = \emptyset$*,* $T_i \cap T_j = \emptyset$*,* $P_i \cap B = \emptyset$*,* $\forall i, j \in \{1, \ldots, q\}, i \neq j$*;*
2. $(\mathcal{SM}_i, M_{0i}) = (P_i, T_i, Pre_i, Post_i, M_{0i})$*,* $\forall i \in \{1, \ldots, q\}$ *is a strongly connected and 1–bounded state machine (where* Pre_i*,*$Post_i$*, and* M_{0i} *are the restrictions of* Pre*,* $Post$*, and* M_0 *to* P_i *and* T_i*);*
3. The set B *of buffers is such that* $\forall b \in B$*: (a)* $|{}^\bullet b| \geq 1$ *and* $|b^\bullet| \geq 1$*, (b)* $\exists i \in \{1, \ldots, q\}$ *such that* $b^\bullet \subset T_i$*, (c)* $\forall p \in P_1 \cup \ldots \cup P_q$*:* $t, t' \in p^\bullet \Rightarrow Pre(b, t) = Pre(b, t')$*.*

The first two items of the previous definition state that a DSSP is composed by a set of SM's ($\mathcal{SM}_i, i = 1, \ldots, q$) and a set of buffers ($B$). By item *3.a*, buffers are neither source nor sink places. The output-private condition is expressed by condition *3.b*. Requirement *3.c* justifies the word "deterministic" in the name of the class: the marking of buffers does not disturb the decisions taken by a SM, i.e., choices in the SM's are free. This definition generalizes the class of DSSP's defined in [Sou93], where buffers are required to have not only a single output SM (output-private) but also a single input SM (input-private).

2.2 Stochastic interpretation

We consider net systems with *timed transitions*. Marking and time independent exponentially distributed random variables associated to the firing of transitions define their *service time*. The mean values of these variables are denoted s_i for each transition t_i of the net.

For the modelling of conflicts we use *immediate transitions* with the addition of (marking and time independent) *routing rates*. Consequently, routing is completely decoupled from duration of activities.

The *visit ratio* of transition t_i with respect to t_j, $v_i^{(j)}$, is the average number of times t_i is visited (fired) for each visit to (firing of) the reference transition t_j. For live and bounded DSSP's, the vector of visit ratios can be computed by solving a linear system of equations, using the incidence matrix and the routing rates at conflicts (see [TSCC95]).

3 The decomposition phase

In this section we present the technique for the structural decomposition of DSSP's. This technique is specially suited for DSSP's composed from large SM's. A cut is defined through buffers, classifying the SM's into several subsets. An aggregated subsystem is obtained per each of these subsets by reducing as much as possible all the *tangible reachability sets* (TRS) of all the other SM's. A method for the TRS reduction is presented such that the *functional behaviour* of the aggregated subsystems is an "*exact projection*" (reachable markings and firing sequences) of the original system on the preserved nodes. Moreover, the probabilistic routing among observable transitions of the reduced SM's is also preserved with the aggregation.

3.1 Reduction of the TRS of a SM

First, we present the reduction technique for the TRS of the SM's. Let $\mathcal{SM} = (P_{\mathcal{SM}}, T_{\mathcal{SM}}, Pre_{\mathcal{SM}}, Post_{\mathcal{SM}})$ be a strongly connected SM. Let $T_v \subseteq T_{\mathcal{SM}}$ be a subset of *observable transitions*. In the following paragraphs we assume a safe initial marking M_0 for $\mathcal{SM}$ that enables a transition in T_v. We are interested in constructing other safe and strongly connected SM which preserves the *branching probabilities* among observable transitions. The branching probability p_{ij} between observable transition t_i and observable transition t_j of a given SM is the probability that transition t_j is the next observable transition in that SM which is fired after the firing of t_i. Later, we will consider as observable transitions those defining the interface between the SM and the rest of the DSSP (i.e., those connected to buffers) and we will assume they are timed.

To preserve branching probabilities we follow a technique similar to that in [BI82]. This technique consists of several steps. Given a SM, a set of observable transitions and the routing probabilities between them, a non-observable transition is shorted at each step and the new routing probabilities are computed. Once all non-observable transitions have been shorted we obtain *the branching probabilities between observable transitions*. These probabilities are *maintained* in the reduced net.

To explain this calculus we need some notation. Let $T_v = \{t_1, \ldots, t_r\}$ be the set of observable transitions and $T_{\mathcal{SM}} \setminus T_v = \{1, \ldots, s\}$ the set of non-observable transitions of a given state machine $\mathcal{SM}$. We denote by $\mathcal{P}$ the $(r+s) \times (r+s)$ matrix of routing probabilities of $\mathcal{SM}$ (the routing matrix in queuing networks terminology). We are going to compute from $\mathcal{P}$ the reduced SM routing $(r \times r)$ matrix $\mathcal{P}'$. If we short transition k on step k $(k = 1, \ldots, s)$, the new routing probabilities are:

$$\begin{aligned} p_{ij}^{(k+1)} &= p_{ij}^{(k)} + p_{ik}^{(k)} \left(\sum_{u=0}^{\infty} \left(p_{kk}^{(k)} \right)^u \right) p_{kj}^{(k)} \\ &= p_{ij}^{(k)} + p_{ik}^{(k)} \left(1 - p_{kk}^{(k)} \right)^{-1} p_{kj}^{(k)} \end{aligned} \qquad (1)$$

$$i, j = t_1, \ldots, t_r, k+1, \ldots, s; k = 1, \ldots, s$$

Then, matrix $\mathcal{P}'$ is $\mathcal{P}' = \left(p_{ij}^{(s+1)} \right)$. We will denote the elements $p_{ij}^{(s+1)}$ by p'_{ij}.

The reduced state machine $(\mathcal{SM}_{T_v}, M_0^{T_v})$ is easily constructed as follows: For each observable transition t_i, add an input p_i^{in} and an output place p_i^{out}; for each $p'_{ij} = 1$, make the fusion of places p_i^{out} and p_j^{in}; and for the rest of $p'_{ij} \neq 0$, add an immediate transition $t_{i,j}$ from $\{p_i^{out}\}$ to $\{p_j^{in}\}$, with routing rate equal to p'_{ij}.

Although the number of immediate transitions of the reduced SM can be (in the worst case) exponential in the number of observable transitions, the number of tangible markings is *never* increased by the reduction technique (it is the number of observable transitions). Thus, the efficiency of the CTMC solution algorithms is improved.

With regard to the initial marking of the reduced SM, since M_0 enables an observable transition t in $\mathcal{SM}$, we define $M_0^{T_v}$ as the marking with only one token in ${}^\bullet t$, thus t is also enabled in $\mathcal{SM}_{T_v}$. From this reduction technique, the next property and corollary easily follow.

Property 3.1 *Let $(\mathcal{SM}, M_0)$ be a safe strongly connected SM and $T_v \subseteq T_{\mathcal{SM}}$ a subset of observable transitions. Let $(\mathcal{SM}_{T_v}, M_0^{T_v})$ be the SM reduced by previous technique. Then, the branching probabilities among all observable transitions are preserved after the reduction.*

Corollary 3.1 *Let $(\mathcal{SM}, M_0)$ be a safe strongly connected SM and $T_v \subseteq T_{\mathcal{SM}}$ a subset of observable transitions. Let $(\mathcal{SM}_{T_v}, M_0^{T_v})$ be the reduced SM by previous technique. Then:*
i) The visit ratios of observable transitions in $(\mathcal{SM}, M_0)$ and $(\mathcal{SM}_{T_v}, M_0^{T_v})$ are the same.
ii) $L(\mathcal{SM}, M_0)|_{T_v} = L\left(\mathcal{SM}_{T_v}, M_0^{T_v}\right)\Big|_{T_v}$
iii) $R(\mathcal{SM}, M_0)|_{^\bullet T_v} = R\left(\mathcal{SM}_{T_v}, M_0^{T_v}\right)\Big|_{^\bullet T_v}$

Now, consider a live and bounded DSSP $(\mathcal{N}, M_0)$. Without loss of generality, in the rest of the paper we

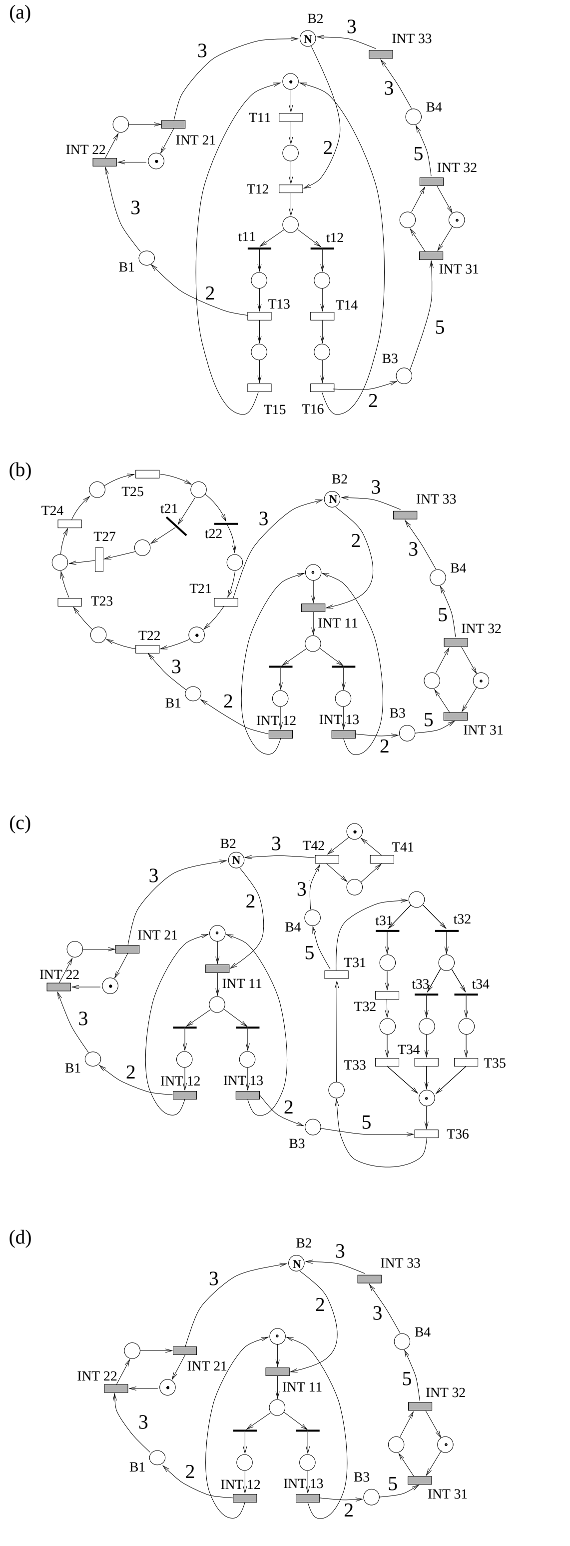

Figure 2: Aggregated subsystems (a) $\mathcal{AS}_1$, (b) $\mathcal{AS}_2$, (c) $\mathcal{AS}_3$, and (d) basic skeleton of the net model in Fig. 1.b (shadow transitions are the observable transitions summarizing the response time of SM's).

will assume that in M_0 all tokens are placed in input places of observable transitions (otherwise such a marking can be reached in linear time on $|T|$ and considered as initial). From the definition of DSSP's, it is clear that provided liveness the only effect of buffers on a single state machine is a possible additional delay for the firing of transitions ($L(\mathcal{N}, M_0)|_{T_{\mathcal{SM}}} = L\left(\mathcal{SM}, M_0^{\mathcal{SM}}\right)$). Therefore, the functional properties of the reduced state machine are also preserved if it is considered within the whole system, as stated in the next property.

Property 3.2 *Let $(\mathcal{N}, M_0)$ be a live and bounded DSSP, $\mathcal{SM}$ a SM of $\mathcal{N}$ and T_v the subset of observable transitions of $\mathcal{SM}$. Let $(\mathcal{N}', M_0')$ be the net system obtained by reducing $\mathcal{SM}$. Then we have:*
i) The visit ratios of observable transitions of $\mathcal{SM}$ are preserved after the reduction.
ii) $L(\mathcal{N}, M_0)|_{TAS} = L\left(\mathcal{N}', M_0'\right)\big|_{TAS}$
iii) $R(\mathcal{N}, M_0)|_{PAS} = R\left(\mathcal{N}', M_0'\right)\big|_{PAS}$
Where the set $TAS = (T_{\mathcal{N}} \setminus T_{\mathcal{SM}}) \cup T_v$ *and the set* $PAS = (P_{\mathcal{N}} \setminus P_{\mathcal{SM}}) \cup {}^\bullet T_v$.

3.2 Cutting the system

The decomposition of a live and bounded DSSP (like that in Fig. 1.b) is based on the splitting in k pieces by a cut defined on buffers. Once the cut and the pieces are selected we construct k *aggregated subsystems* ($\mathcal{AS}_1, \ldots, \mathcal{AS}_k$; see Figs. 2.a, 2.b, and 2.c) and a *basic skeleton system* ($\mathcal{BS}$; see Fig. 2.d). First, we formally define the cut.

Definition 3.1 *Let* $(\mathcal{N}, M_0) = (P_1 \cup P_2 \cup \cdots \cup P_q \cup B, T_1 \cup T_2 \cup \cdots \cup T_q, Pre, Post, M_0)$ *be a live and bounded DSSP. A subset $Q \subseteq B$ of buffers is said k-cut of $\mathcal{N}$ ($k \geq 2$) iff there exist k subnets* $\mathcal{N}_i = (P_{\mathcal{N}_i}, T_{\mathcal{N}_i}, Pre_{\mathcal{N}_i}, Post_{\mathcal{N}_i}), i = 1, \ldots, k$*, of $\mathcal{N}$ verifying:*
i) $\bigcup_{i=1}^k T_{\mathcal{N}_i} = T$, $T_{\mathcal{N}_i} \cap T_{\mathcal{N}_j} = \emptyset$, $\forall i \neq j$, $i,j = 1, \ldots, k$.
ii) $\forall i \in \{1, \ldots, q\}$ $\exists j \in \{1, \ldots, k\}$ *such that* $T_i \subseteq T_{\mathcal{N}_j}$
iii) $P_{\mathcal{N}_i} = {}^\bullet T_{\mathcal{N}_i} \cup T_{\mathcal{N}_i}{}^\bullet$, $i = 1, \ldots, k$.
iv) $\bigcup_{i=1}^k P_{\mathcal{N}_i} = P$ *and* $\bigcup_{i \neq j} \left(P_{\mathcal{N}_i} \cap P_{\mathcal{N}_j}\right) = Q$, $i,j \in \{1, \ldots, k\}$.
v) $Pre_{\mathcal{N}_i} = Pre|_{P_{\mathcal{N}_i} \times T_{\mathcal{N}_i}}$, $Post_{\mathcal{N}_i} = Post|_{P_{\mathcal{N}_i} \times T_{\mathcal{N}_i}}$, $i \in \{1, \ldots, k\}$.

Now we define the aggregated subsystems and the basic skeleton obtained from a k-cut.

Definition 3.2 *Let $(\mathcal{N}, M_0)$ be a live and bounded DSSP, $Q \subseteq B$ a k-cut of $\mathcal{N}$ and $\mathcal{N}_i, i = 1, \ldots, k$, the k subnets defined by Q. For each $i = 1, \ldots, k$, the aggregated subsystem $\mathcal{AS}_i = (\mathcal{AN}_i, M_0^{\mathcal{AN}_i})$ is the system obtained from $(\mathcal{N}, M_0)$ by reducing every SM not included in $\mathcal{N}_i$. All buffers are maintained. The basic skeleton $\mathcal{BS} = (\mathcal{BN}, M_0^{\mathcal{BN}})$ is the system obtained from $(\mathcal{N}, M_0)$ by reducing all the SM's.*

The next theorem includes the main result of this section. It relates the behaviour of a DSSP with those of the aggregated subsystems and the basic skeleton. Previously, we introduce some additional notation. For each $i = 1, \ldots, k$, let TV_i be the set of observable transitions of $\mathcal{N}_i$ and PV_i the set of input places of transitions of TV_i belonging to a SM.

Theorem 3.1 *Let $(\mathcal{N}, M_0)$ be a live and bounded DSSP, $Q \subseteq B$ a k-cut of $\mathcal{N}$, $\mathcal{AS}_i = (\mathcal{AN}_i, M_0^{\mathcal{AN}_i}), i = 1, \ldots, k$, $\mathcal{BS} = (\mathcal{BN}, M_0^{\mathcal{BN}})$ the aggregated subsystems and the basic skeleton derived from Q. Then:*
i) $L(\mathcal{N}, M_0)|_{TAS_i} = L\left(\mathcal{AN}_i, M_0^{\mathcal{AN}_i}\right)\Big|_{TAS_i}$, *for* $i = 1, \ldots, k$.
ii) $R(\mathcal{N}, M_0)|_{PAS_i} = R\left(\mathcal{AN}_i, M_0^{\mathcal{AN}_i}\right)\Big|_{PAS_i}$, *for* $i = 1, \ldots, k$.
iii) $L(\mathcal{N}, M_0)|_{TBN} = L\left(\mathcal{BN}, M_0^{\mathcal{BN}}\right)\big|_{TBN}$
iv) $R(\mathcal{N}, M_0)|_{PBN} = R\left(\mathcal{BN}, M_0^{\mathcal{BN}}\right)\big|_{PBN}$
Where the sets $TAS_i = \left(\bigcup_{j \neq i} TV_j\right) \cup T_{\mathcal{N}_i}$, $PAS_i = \left(\bigcup_{j \neq i} PV_j\right) \cup P_{\mathcal{N}_i} \cup B$, $TBN = \bigcup_{i=1}^k TV_i$, *and* $PBN = \left(\bigcup_{i=1}^k PV_i\right) \cup B$.

Proof: An aggregated subsystem $\mathcal{AS}_i$ can be computed from the original net by reducing one by one the SM's not included in $\mathcal{N}_i$. Applying at each step property 3.2, results *(i)* and *(ii)* follow. The same argument is valid for *(iii)* and *(iv)*. Q.E.D.

We remark that the previous result assures an "exact functional aggregation" of the original DSSP, in the sense of state space projection. In this way, good performance approximations should be expected.

4 Iterative approximation phase

The technique for an approximate computation of the throughput that we present now is, basically, a *response time approximation* method [ABS84, CCJS94]. The observable transitions of $\mathcal{N}_j$ in $\mathcal{AS}_i$ ($j \neq i$) approximate the response time of all the subsystem $\mathcal{N}_j$. A direct (non-iterative) method to compute the constant service rates of observable transitions in order to represent the aggregation of the subnet gives, in general, low accuracy. Therefore, we are forced to define a *fixed-point search iterative process.* The proposed algorithm is the following:

select a k-cut Q;
derive $\mathcal{AS}_i$, $i = 1, \ldots, k$, and $\mathcal{BS}$;
give an initial service rate μ_t^0 for each $t \in TBN$;
$j := 0$; {counter for iteration steps}
repeat
  $j := j + 1$;
  for $i := 1, \ldots, k$
    solve the aggregated subsystem $\mathcal{AS}_i$ with
      input: μ_t^j for each $t \in TV_l$ $(l = 1, \ldots, i-1)$
        μ_t^{j-1} for each $t \in TV_l$ $(l = i+1, \ldots, k)$
      output: $\mathcal{X}_i^j$
    solve the basic skeleton system $\mathcal{BS}$ with
      input: μ_t^j for each $t \in TV_l$ $(l = 1, \ldots, i-1)$
        μ_t^{j-1} for each $t \in TV_l$ $(l = i+1, \ldots, k)$
        ratios among μ_t^0 of $t \in TV_i$ and $\mathcal{X}_i^j$
      output: scale factor of μ_t^j of $t \in TV_i$
  end for;
until convergence of $\mathcal{X}_1^j, \ldots, \mathcal{X}_k^j$;

In the above procedure, once a k-cut has been selected and given some initial values μ_t^0 for service rates of all observable transitions except those in $\mathcal{N}_1$, the underlying CTMC of aggregated subsystem $\mathcal{AS}_1$ is solved (the computation of appropriate initial values is explained in [PJCS95]). From the solution of that CTMC, the first estimation $\mathcal{X}_1^1$ of the throughput of $\mathcal{AS}_1$ can be computed. Then, the initial estimated values of service rates of observable transitions TV_1 must be derived. To do that, we take the initial values μ_t^0 for service rates of transitions in TV_1 and we search in the basic skeleton a *scale factor* for all these rates such that the throughput of the basic skeleton and the throughput of $\mathcal{AS}_1$, computed before, are equal. The same procedure is executed for each aggregated subsystem in a cyclic way. Each time we solve the aggregated subsystem $\mathcal{AS}_i$ we obtain in the basic skeleton a new estimation of the observable transitions rates of TV_i.

The computation of the scale factor in the basic skeleton can be implemented with a linear search. Now the net system (the basic skeleton) has considerably fewer states than the original one. In each iteration of this linear search, the underlying CTMC of the basic skeleton is solved. Note that only in the first iteration the CTMC is completely derived. For later iterations only some values must be changed. In [BLS95], it is proved that in stochastic DSSP's (with time and marking independent rates) the throughput of any transition is a monotonic function of the rate of any other transition. In other words, if we increase (decrease) the rate of any transition, the throughput of any other transition will not be decreased (increased). This fact assures that the linear search in the basic skeleton (being a DSSP) converges.

The convergence of the solution of the entire method can be proved by similar arguments to those in [MT95]. With regard to the accuracy of the results, no formal proof gives positive answers to the question, but an extensive battery of numerical experiments has shown us that the error is less than 3% in the worst cases.

5 Numerical results

Let us present some numerical values obtained for the manufacturing system in Fig. 1.a. We study its net model (Fig. 1.b) for N (tokens in $B2$, i.e. the total number of pallets) ranging from 10 to 25.

In Table 1 we present the iteration steps of the method for the case $N = 17$. The exact quantity of product A manufactured by the system per time unit (throughput of transition $T21$) is 0.29444 and of product B (throughput of transition $T41$) 0.09814. The underlying CTMC has 242448 states. Columns $\mathcal{X}(Ti)$ are the estimated values for throughput of transition Ti at each iteration step. Columns 'scale f.' are the scale factors modifying the previous estimated service rates, computed with the basic skeleton. Convergence of the method is usually obtained from the fourth iteration step. The error for this example was -0.142%. The following additional fact must be remarked: the number of states of the underlying CTMC's of $\mathcal{AS}_1$, $\mathcal{AS}_2$, $\mathcal{AS}_3$, and $\mathcal{BS}$ (see Fig. 2) are 12348, 14049, 46594, and 5849 respectively. Note that the TRS of $\mathcal{AS}_3$ could be reduced by splitting its two SM's.

Table 1: Iteration results for the case $N = 17$

Service and routing rates as given in Fig. 1.b					
$\mathcal{AS}_1$		$\mathcal{AS}_2$		$\mathcal{AS}_3$	
$\mathcal{X}(T11)$	scale f.	$\mathcal{X}(T21)$	scale f.	$\mathcal{X}(T41)$	scale f.
0.57576	1.02706	0.29489	1.08452	0.09845	2.07529
0.58783	1.02029	0.29404	1.08633	0.09801	2.11383
0.58804	1.02017	0.29402	1.08636	0.09800	2.11249
0.58804	1.02017	0.29402	1.08636	0.09800	2.11249
Exact $\mathcal{X}(T21) = 0.29444$				Error: -0.142%	

Table 2: Accuracy for several values of N

	Thousands of tangible markings					$\mathcal{X}(T21)$	% Error
N	Orig.	$\mathcal{AS}_1$	$\mathcal{AS}_2$	$\mathcal{AS}_3$	$\mathcal{BS}$	Prod. A	Approx.
10	40	2	2	9	1	0.247962	-1.116
14	126	6	7	25	3	0.285181	-0.293
17	242	12	14	47	6	0.294444	-0.142
21	488	25	30	91	12	0.300156	-0.076
25	863	45	56	158	22	0.302767	-0.045

In Table 2 we present the accuracy of the method for several values of the parameter N. Always four iterations are needed, in this case, to reach convergence.

We remark that the accuracy of the method improves when the number of tokens is increased (additionally, monotonicity of $\mathcal{X}$ with respect to the resources can be observed).

In the extensive numerical experiments, convergence was reached after no more than five iteration steps and the error was no more than 3%.

6 Conclusions

An aggregation technique is presented as a basis for system decomposition. The technique is based on a reduction of state machines, achieved by preserving the interface transitions, called observable. The reduction preserves the projection of firing sequences on the observable transitions and, even more, the branching probabilities among these transitions. The original system is decomposed with a cut defined through buffers. Then, each aggregated subsystem is obtained by reducing a different subset of state machines. Both aggregations and decomposition can be done in polynomial time on the net size.

Even if the technique is applied here for DSSP's, it can be used for general models to reduce safe state machine subnets whose internal decisions are completely independent of the outside behaviour.

After the decomposition phase, a fixed-point search iterative process is used to approximate the throughput. Extensive numerical examples have shown that the iterative method converges in three to five steps. The state spaces are usually reduced in more than one order of magnitude and with very little error (less than 3% in all cases that we tried).

Finally, a realistic example taken from the manufacturing domain was selected to illustrate the method.

References

[ABS84] S. C. Agrawal, J. P. Buzen, and A. W. Shum. Response time preservation: A general technique for developing approximate algorithms for queueing networks. In *Proc. 1984 ACM Sigmetrics Conf. on Measurement and Mod. of Comp. Syst.*, pp. 63–77, Cambridge, MA, Aug. 1984.

[BI82] S. Balsamo and G. Iazeolla. An extension of Norton's theorem for queueing networks. *IEEE Trans. on Soft. Eng.*, 8(4):298–305, Jul. 1982.

[BLS95] F. Baccelli, Z. Liu, and M. Silva. Local and global monotonicities of stochastic Petri nets. Research report, 1995. In preparation.

[CCJS94] J. Campos, J. M. Colom, H. Jungnitz, and M. Silva. Approximate throughput computation of stochastic marked graphs. *IEEE Trans. on Soft. Eng.*, 20(7):526–535, Jul. 1994.

[MT95] V. Mainkar and K. Trivedi. Fixed point iteration using stochatic reward nets. In *Proc. PNPM'95*, pp. 21–30, Durham, N. Carolina, USA, Oct. 1995. IEEE-Computer Society Press.

[Mur89] T. Murata. Petri nets: Properties, analysis, and applications. *Proc. IEEE*, 77(4):541–580, Apr. 1989.

[PJCS95] C. J. Pérez-Jiménez, J. Campos, and M. Silva. State machine reduction for the approximate performance evaluation of manufacturing systems modelled with cooperating sequential processes. Research Report RR-95-07, Dep. de Inf. e Ing. de Sist., CPS, U. Zaragoza, Spain, Sep. 1995. Extended version.

[Rei82] W. Reisig. Deterministic buffer synchronization of sequential processes. *Acta Informatica*, 18:117–134, 1982.

[RTS95] L. Recalde, E. Teruel, and M. Silva. On well-formedness analysis: The case of DSSP's. In J. Desel, ed., *Proc. STRICT '95*. Springer Verlag, 1995. (To appear).

[Sil93] M. Silva. Introducing Petri nets. In *Practice of Petri Nets in Manufacturing*, chapter 1. Chapman & Hall, 1993.

[Sou93] Y. Souissi. DSSP's: A class of structured Petri nets. In G. Rozenberg, ed., *Advances in Petri Nets 1993*, vol. 674 of *LNCS*, pp. 406–426. Springer-Verlag, Berlin, 1993.

[TSCC95] E. Teruel, M. Silva, J. M. Colom, and J. Campos. Functional and performance analysis of cooperating sequential processes. In F. Baccelli, A. Jean-Marie, and I. Mitrani, eds., *Quantitative Methods in Parallel Systems*, Esprit Basic Research Series, pp. 52–65. Springer, 1995.