ABSTRACT

Stepwise Reduction and Approximation Method for Performance Analysis of Generalized Stochostic Petri Nets

by
Jinming Ma

This thesis delves into the performance analysis of generalized stochastic Petri net (GSPN) model by using an approximation method: the Stepwise Reduction and Approximation (SRA) Method. The key point is that we are able to analyze a subnet in isolation by keeping its token flow direction and its sub-throughput equivalent with all the possible tokens entering into the subnet. The thesis first defines various kinds of potentially reducible subnets, subnet selection rules, approximation subnet construction rules, and reduction evaluation rules. Then corresponding to the possible subnets, the approximation method is used stepwisely until the interested measures are found with the global state space reduced. Two GSPN model examples from the literature are analyzed by using the proposed method. The approximation errors are given and discussed. Finally, the conclusions are drawn and future research is discussed.

STEPWISE REDUCTION AND APPROXIMATION METHOD FOR PERFORMANCE ANALYSIS OF GENERALIZED STOCHASTIC PETRI NETS

by
Jinming Ma

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science

Manufacturing Engineering Programs

January 1993

Blank Page

APPROVAL PAGE

Stepwise Reduction and Approximation Method for Performance Analysis of Generalized Stochostic Petri Nets

Jinming Ma

12/15/92

Dr. MengChu Zhou, Thesis Adviser
Assistant Professor of Electrical and Computer Engineering, and
Manufacturing Engineering Programs, NJIT

12/15/92

Dr. Raj S.Sodhi, Committee Member
Director of Manufacturing Engineering Programs, and
Associate Professor of Mechanical and Industrial Engineering, NJIT

Dec. 15, 92

Dr. Nouri Levy, Committee Member
Associate Professor of Mechanical and Industrial Engineering, NJIT

BIOGRAPHICAL SKETCH

Author: Jinming Ma

Degree: Master of Science in Manufacturing Engineering

Date: January 1993

Undergraduate and Graduate Education:

- Master of Science in Manufacturing Engineering, New Jersey Institute of Technology, Newark, NJ, 1993
- Master of Science in Automatic Control, Beijing Institute of Technology, Beijing 100081, P.R.China, 1986
- Bachelor of Science in Electrical Engineering, North China University of Technology, Beijing 100041, P.R.China, 1983

Major: Manufacturing Engineering

Presentations and Publications:

Ma, Jinming, Hesheng Xu and Jindi Chen. "An Improvement on Multi-variable Minimum Variance Regulator." Journal of Beijing Institute of Technology, P.R.China, Volume 1989, No.4, 1989. And Presented in Zhangjiajie, Hunan, P.R.China, June, 1988, at the China Annual Conference on Automatic Control Theory and its Application.

Ma, Jinming. "A Method of Multivariable Self-tuning Control." Journal of Army (ACTA ARMAMENTARII), P.R.China, Volume 42, No.4, 1989.

Ma, Jinming. " A Self-tuning Control for Multivariable System with Different Time Delays." Journal of Graduate of Beijing Institute of Technology, Beijing 100081, P.R.China, No.1, 1986.

Ma, Jinming and MengChu Zhou. "Performance Evaluation of Discrete Event Systems Via Stepwise Reduction and Approximation of Stochastic Petri Nets." To appeared at the IEEE International Conference on Decision and Control, Tuscon, AZ, December 16-18, 1992.

Zhou, MengChu and Jinming Ma."Reduction and Approximation for Performance Evaluation of Stochastic Petri Nets with Multi-input and Multi-output Modules." Submitted to 12th IFAC World Congress (DES Control), Sydney, Australia, July, 1993.

Ma, Jinming and MengChu Zhou. "An Approximation Algorithm for Performance Analysis of Arbitrary Stochastic Petri Nets,"Presented in Newark, NJ, USA, April, 1992, at the 3rd MINI-TECH Conference by Society for the Advancement of Material and Process Engineering, NJ, 1992.

Ma, Jinming and MengChu Zhou. "Generalizing the Moment Generating Function Approach to Performance Analysis to Deterministic and Stochastic Petri Nets." Presented in Newark, NJ, USA, April, 1992, at the 3rd MINI-TECH Conference by Society for the Advancement of Material and Process Engineering, NJ, 1992.

Ma, Jinming and Nouri Levy. "Modeling and Performance Analysis of Flexible Manufacturing System by Using Petri Nets." Presented in Newark, NJ, USA, April, 1992, at the 3rd MINI-TECH Conference by Society for the Advancement of Material and Process Engineering, NJ, 1992.

This thesis is dedicated to
my parents, my wife and my son.

ACKNOWLEDGEMENT

The author would like to express the sincere gratitude to thesis advisor, Dr. MengChu Zhou, for his ingenious guidance, friendship and moral support throughout the course of this work.

The author wishes to acknowledge Dr. Raj S. Sodhi and Dr. Nouri Levy for their financial supports. Without all those supports, this work could not have been finished.

Special thanks to Dr. Raj S. Sodhi and Dr. Nouri Levy for serving as the members of the thesis committee, and for their professional guidance.

The author is thankful to his wife and other family members for their supporting and encouraging him to complete the work successfully.

A thank you to, Professor Zhifang Zhang, Vice President of Institute of Management of China Academy of Science, Professor Chunbo Feng, Director of automation institute of South-East University of China, Professor Runjin Ma, Vice President of North China University of Technology, and Professor Hesun Zhu, President of Beijing Institute of Technology, for their recommending and helping the author to the United States.

And finally, the author gratefully acknowledge the valuable discussions with all his friends.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Petri Net Theory

Petri nets (PN) [8] are a useful graphical tool for modeling and analyzing systems involving such features as concurrence, synchronization and mutual exclusion, and so on. The traditional PN model (called non-timed PN or general PN) has no means of expressing time and cannot be used to study system performance [19]. If only a single arc is allowed between a place and a transition or vice versa, an ordinary PN results. A Marked Graph results if each place in the PN has exactly one input and output arc, and a State Machine PN results if each transition in the PN has exactly one input and output arc. In the past, many attempts have been made to include time in a PN model, and we will refer these models as Timed Petri Nets (TPN), including Timed Transition PN (TTPN) and Timed Place PN (TPPN). In TTPN, the firing delay of a transition can be specified either deterministically or stochastically. In Deterministic Timed Petri Nets (DTPN), the delay is either specified by a constant or a finite interval. In Stochastic Timed Petri Nets (STPN) [11], the delay is a random time that is generated by a user-specified distribution, that is, the transitions are with arbitrarily distributed time delays, also called arbitrary stochastic PN [12]. Assuming the firing times with exponential distribution, we obtain Stochastic Petri Nets (SPN) [18]. If it also involves the immediate transitions, meaning no time delay, we call it Generalized Stochastic Petri Net (GSPN)[21]. The Extended Stochastic Petri Nets (ESPN) [14], which partition transitions into three classes - exclusive, competitive and concurrent - are developed to allow delays generally distributed, including the deterministic transition delays, and non-exponential transition delays, and for concurrent ones the memoryless property of exponential distribution is required for exact solutions. Deterministic and Stochastic PN (DSPN) contain both deterministic and stochastic transition firing time delays [13].

In addition, there are many kinds of extensions applicable to both timed and non-timed PN - such as inhibitor arcs, probabilistic arcs (or random switches), priority function and so on. They lead to different classification of PNs. An inhibitor arc with multiplicity k from a place to a transition has a small circle rather than an arrowhead at the transition. The transition cannot be enabled unless the number of tokens in that place is less than k. Firing that transition does not affect the number of tokens in the inhibiting input place. When a transition fires, the tokens remove from the normal input places and deposit into the output places as usual, but the number of tokens in inhibiting input place remains unchanged. A probabilistic arc from a place called place probabilistic arc to a set of immediate transitions is used to resolve conflicts between two or more immediate transitions and is basically a discrete probability distribution. A probabilistic arc from a transition called transition probabilistic arc to a set of output places deposits a token in one and only one of the places in the set. The choice of which place receives the token is determined by the probability labeled on each branch of the arc. We also have counter arc, counter-alternative arc, and so on [14]. A priority function is defined for the marking in which both timed and immediate transitions are enabled. Usually, immediate transitions are given the higher priority. Inhibitor arcs and transition probabilistic arcs do not expand the modeling power of TPN, but in some cases, they allow a simpler description of the system operations, since the use of inhibitor arcs and transition probabilistic arcs can reduce the number of random switches to be defined in the TPN [14].

1.2 Performance Evaluation (PE)

Recently the Performance Evaluation (PE) for TPN has received much attention. For DTPN, each transition takes exactly r units of time to complete its execution. The maximum cycle time can be computed for processing a task. This cycle time is regarded as a performance measure. Basically it can deal with only decision-free PN or those that can be converted to them.

For STPN, performance measures are average production rate, average in-processing inventory, average resource utilization and average waiting time. Molloy [18] established the connection between SPN and discrete space Markov process and formed the basis for PE using SPN. The PE method based on Markovian analysis models and numerical solution of the equilibrium equations is called Numerical Method.

1. Software tools for PE, in which the steps are involved in going from the PN model to reachability tree and then to the Markov Chain have all been automated. They can be found in several software packages.

Chiola [15] has developed Great SPN for the construction and analysis of SPN and DSPN models. This software accepts deterministic delays or exponentially distributed firing rates. It also computes the transient and steady state solutions to the Markov Chains.

Dugan et al. [16] have developed the Duke extended SPN evaluation package (DEEP) for the PE of SPN models. This led to a new version: Stochastic Petri Net Package (SPNP) [1], which is available in ITC computer laboratory, NJIT, and can deal with GSPN, which also permits the use of inhibitor arcs, priority functions, place probabilistic arcs, marking dependent firing rates, and throughput subnets such as Erlang subnet. All those additional modeling capabilities do not destroy their equivalence to Markov Chain.

Holiday and Vernon [17] have developed the GTPN analyzer for PE of the Generalized timed PN models.

All NM methods mainly used the steady state probabilities obtained from the Markov chain to compute the average (expected) tokens in a place, average firing rate (throughput) of a transition, and the probability that a place is not empty and a transition is enabled .

2. Moment Generating Function Approach (MGF), which offers the closed form analytical solutions, is another way to conduct PE for a class of PN by using Moment generating function. Theoretically, it can deal with SPN, GSPN, ESPN [4] and DSPN [23].

In addition, Viswanadham and Narahari [19] have given a brief review of other existing PE techniques. First is the discrete event simulation, which enables us to run through the detailed operation of the system using the computer program but at the expense of greater programming time to create the model, greater input time to generate data, and increased computer time for running the model. Second is the queueing networks, which capture the dynamics, interactions, and uncertainties in the system in an aggregate way. Third is the perturbation analysis, which enables parameter sensitivities to be computed on-line, in real time and can handle detailed features of the systems, but cannot predict accurately the effects of large changes in decisions. Using NM by solving the equivalent Markov chain involves the solution of a set of linear algebraic equation. In this case, a theoretical solution is only available since closed-form solutions are difficult, if not impossible, to find. Particularly, for the large state cases, we cannot use the NM and MGF methods to conduct PE. Since ASPN or practical PN model often leads to a very large state space, either approximation (APPR) or the simulation (SIM) methods are needed.

From the above discussions, here we give a relationship graph (Figure 1.1) for outlining the state-of-art.

1.3 Basic idea of the approximation method

It is impossible to analyze a GSPN with the state space explosion problem by using the conventional techniques such as Numerical Method (NM) and Moment Generating Function (MGF) approach. Because many real PN suffer from the state explosion problem, approximation methods are needed. For a class of GSPN, it is possible to analyze it in isolation by using the approximation subnets which can be equivalent to Generalized Stochastic PN (GSPN) subnets. After obtaining the section results, we can construct the

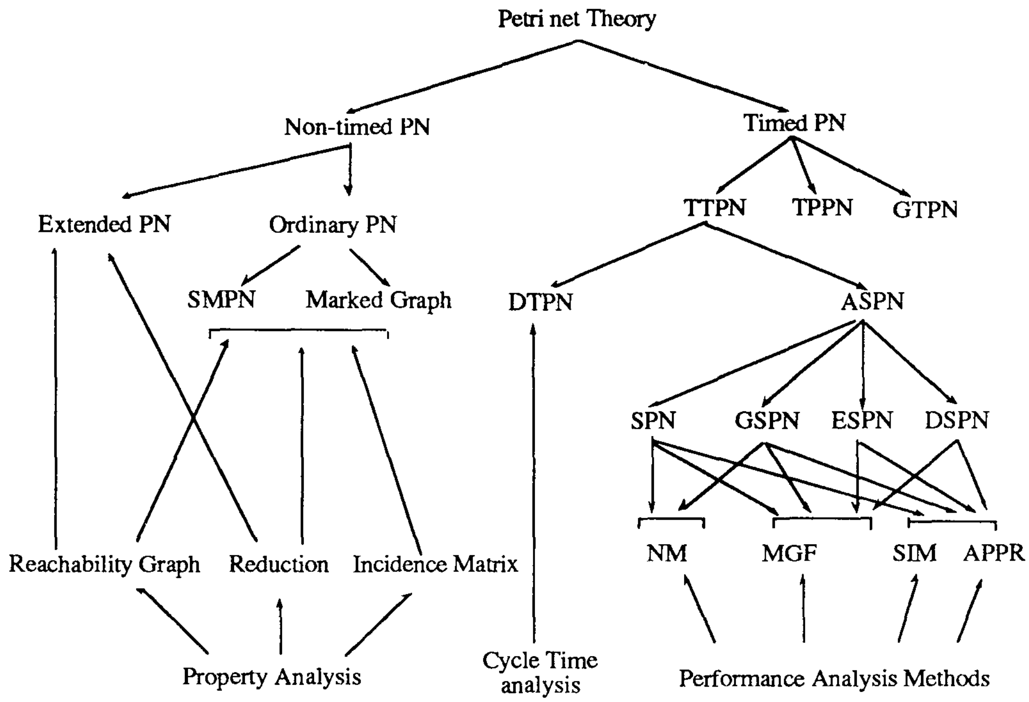


Figure 1.1 The relationship graph of Petri Net theory

approximation subnets with reduced state space and substitute them stepwisely into the original PN model which would be computationally intractable with conventional methods.

For some practical cases, modeling a system with PN can lead to a large number of places, transitions and arcs. For general PN, it is possible to make a conclusion about the token flow and structural properties of the original PN by studying and analyzing the reduced net [20], [21]. In those cases, the number and flow direction of tokens into and out of the original PN (or subnet) are conserved. Thus from an input and output point of view, the flow of tokens is indistinguishable, and the nets are equivalent and still keep the properties such as boundedness, liveness etc., of the original PN. The related results were also reported in [10].

For the STPN, based on the similar idea, we proposed a method which tries to keep both the token and flow direction, and throughput (expected firing time) equivalent to the original one by replacing the reducible subnet which can be analyzed in isolation. For the cases exactly meeting the two element requirements, we can have the equivalent reduced PN model. For some other cases, which frequently exist, we must loosen some conditions in order to further conduct PE, especially for a large state space case. The approximation methods must be used in order to avaid the time-consuming simulation.

1.4 Main Work

The limitation of STPN is that the graphical PN model for a system rapidly becomes more difficult as system size and complexity increase. Therefore, the number of states of associated Markov chain grows very fast as the dimension of the PN graph increases, or as the initial markings are of large number even if this PN may not be so complicated.

This thesis proposes a Stepwise Reduction Approximation method (SRA) [25], [28] to approximate the GSPN model with approximation subnet and then to reduce its state space. Based on the reduced model, we conduct the performance analysis by using SPNP software [1] to get the numerical results. Two GSPN examples [3] [19] are used to show how the approximation method works. It shows that the approximation method is one of the reasonable and efficient methods to deal with the practical PN model. Further research works, such as combining the approximation method with the MGF approach to get the closed form results of performance analysis, and the approximation method dealing with the ASPN or so, are under study [24],[27].

CHAPTER 2

THE STEPWISE REDUCTION AND APPROXIMATION METHOD - DEFINITIONS AND RULES

2.1 Fundamental Theory

For a fundamental knowledge of Petri net theory, a reader is referred to [7],[8]. To be consistent, we introduce the following definition and notation [29].

A Petri net Z=(P, T, I, O), where

(1) $P=\{p_1, p_2, ..., p_n\}$, n>0;

(2) $T=\{t_1, t_2, ..., t_s\}$, s>0 with $P \cup T \neq \emptyset$ and $P \cap T = \emptyset$;

(3) I: $P\times T \rightarrow \{0, 1\}$; and

(4) O: $P\times T \rightarrow \{0, 1\}$.

In this definition, p_i ($1 \leq i \leq n$) is called a place, t_i ($1 \leq i \leq s$) a transition, I an input function defining the set of directed arcs from P to T, and O an output function defining the set of directed arcs from T to P. (P, T, I, O, m_0) is a marked Petri net where m_0 is an initial marking whose i^{th} component represents the number of tokens in place p_i.

The *preset* of p is the set of all input transitions to the place p, i.e., $\bullet p=\{t: t\in T$ and $O(p, t)\neq 0\}$. The *postset* of p is the set of all output transitions from the place p, i.e., $p\bullet=\{t: t\in T$ and $I(p, t)\neq 0\}$. Similarly, $\bullet t=\{p \mid O(p,t) \neq 0\}$ and $t\bullet =\{p \mid I(p,t) \neq 0\}$.

2.1.1 Definitions

The subsystem can be thought of as a black box whose behavior is characterized by the expected amount of time delay that takes under the conditions of putting all the possible arriving tokens to the subsystem, and the path that takes through the system with the time that takes to leave it. In general, these expected delays will depend on the present state of the subsystem.

Definition 1 *[Subnet Z']*

Petri nets is defined as Z = (P, T, I, O) [12]. Let Z' = (P', T', I',O'). Z' is a subnet of Z, $Z \supseteq Z'$, iff $P \supseteq P'$, $T \supseteq T'$, and I'(p,t) = I (p,t), $\forall$ p $\in$ P',t $\in$ T' ; O'(p,t) = O(p,t), $\forall$ p $\in$ P',t $\in$ T'.

Definition 2 *[Input (Output) Place P^{in} (P^{out}) (for Transition: T^{in} ,T^{out})]*

A place p $\in$ P' is a p^{in} (p^{out}) place iff there exists t $\in$ •p (t $\in$ p•) such that t $\notin$ T'. A set of p^{in} (p^{out}) are denoted as P^{in} (P^{out}) . So, P^{in} = { p_1^{in} ... p_k^{in} } , P^{out} = { p_1^{out} ... p_l^{out} }. A transition t $\in$ T' is a t^{in} (t^{out}) transition iff there exists p $\in$ •t (p $\in$ t•) such that p $\notin$ P'. A set of t^{in} (t^{out}) is denoted as T^{in} (T^{out}). T^{in} = { t_1^{in} ... t_m^{in} } , T^{out} = { t_1^{out} ... t_n^{out} }. P^{in} and T^{in} are called Input of subnet, and P^{out}, T^{out} are called Output of subnet. If K > 1, we call it multi-input; if l >1, multioutput results.

Definition 3 *[Place Subnet Z^p]*

A place Subnet Z^p is a subnet of Z, Z^p = (P^p, T^p , I^p ,O^p), if P^p = (P^{in} , P^{out}) $\cup$ P^f where P^{in} and P^{out} are two sets of distinct places, and P^f is a set of places except P^{in} and P^{out} in Z^p

Definition 4 *[Transition Subnet Z^t]*

A transition Subnet Z^t is a subnet of Z, Z^t= (P^t , T^t , I^t ,O^t), if T^t = (T^{in} , T^{out}) $\cup$ T^f where, T^{in} and T^{out} are two sets of distinct transitions. T^f is a set of transitions except T^{in} and T^{out} in Z^t .

Definition 5 *[Complete Subnet Z^c]*

A complete Subnet Z^c is a subnet of Z, Z^c= (P^c, T^c , I^c ,O^c), if P^c = $P^{in} \cup P^f$ and T^c = $T^{out} \cup T^f$, or if P^c = $P^{out} \cup P^f$ and T^c = $T^{in} \cup T^f$.

Definition 6 *[General associated Petri net Z^a]*

A general associated Petri net Z^a of a subnet Z^p, Z^t, or Z^c is a PN which is made up of the subnet and appropriate arcs to link the Input and Output of the subnet as follows:

the dumb transitions T^d, for the Z^p subnet,

the dumb places P^d, for the Z^t subnet,

the dumb arcs , for the Z^c subnet,

and maintain the agreeable (same) configuration in Z. For single input and output case, we call it associated Petri net.

Definition 7 *[Equivalent Throughput Subnet (ETS) Z^e]*

The equivalent throughput (approximation) subnet for each above subnet is a GSPN which

(1) has the agreeable (same) configuration as Z^p, Z^t and Z^c in Z but a reduced state space,

(2) has of the equivalent expected time delay entering the subnet through setting all possible initial tokens with its general associated PN of the subnet to keep the input and output dynamic properties.

REMARK: It is necessary to mention that the equivalent average time, or the equivalent throughput, is subnet initial marking dependent. From the following definition on interactive subnet, one will see that this concept is the key for the ETS construction in order to keep the approximation accurate. It is called marking dependent ETS.

2.1.2 Subnet Selection Rules

For using the approximation method, we must select the reducible subnets. They should satisfy the following conditions in order to keep the approximation accuracy:

1. $P^f \cap (T - T') = \varnothing$, $T^f \cap (P - P') = \varnothing$
2. $\forall\ t \in\ T^{out}\ \ ({}^{\bullet}t\ \cap\ P) \in\ P'$

$\forall\ p \in P^{out}\quad (\bullet p \cap T) \in T'$

3. $\forall\ t \in T^{in}\quad (t\bullet \cap P) \in P'$

$\forall\ p \in P^{in}\quad (p\bullet \cap T) \in T'$

Physical meaning:

Subnet selection rule 1 guarantees that the system is calculated in isolation, because the subnet is independent on the rest of Z both in the structure and parameters, and that a token which enters the subnet eventually leaves it, and that no tokens are created or absorbed by a firing sequence within the subnet. Rule 2 and Rule 3 guarantee that no tokens will be deposited in the place given by {P-P'} and that the recycling the subnet only depends on the marking of P'. This guarantees that the subnet is self-contained.

2.1.3 ETS Subnet Construction Rules

Definition 8 *[K-order Closed Subnet Z^{cl}]*

Let ET_K be the subnet equivalent throughput at initial marking $m(p_0) = K$. If $ET_{K-1} \neq ET_K = ET_{K+1}$, it is called K-order closed subnet. Otherwise, it is called open (loop) subnet.

For example, subnet Z1 is 3-order closed subnet, since the subnet throughput is same when $m(p_0) = m(p) \geq 3$. Similarly, Z2, Z3 and Z4 all are 2-order subnets.

Definition 9

If $|ET_{K-1} - ET_K| > \varepsilon \geq |ET_K - ET_{K+1}|$, it is called ε–K-order subnet where ε is a given small positive number.

This definition is used in a open subnet in order to save cost.

So the definition also can be generalized to multiinput and multioutput modules as follows:

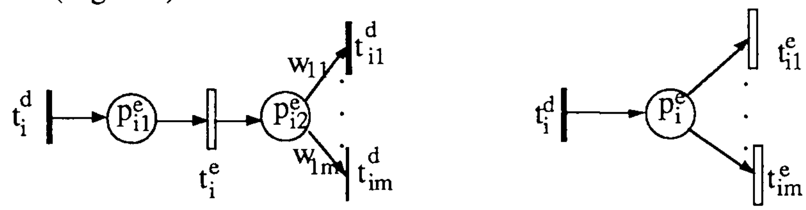
Definition 10

Let ET_K be the subnet equivalent throughput at initial marking vector $M(p_0) = \{k_1,k_2,...k_p\}$, Let ET_{K-i} be the subnet equivalent throughput at initial marking vector $M(p_0) = \{k_1\text{-}i, k_2\text{-}i,... k_p\text{-}i\}$, If $ET_{K-1} \neq ET_K = ET_{K+1}$, it is called K-vector-order subnet.

Similarly, we can define the ε–K-vector-order subnet.

For explanation, see example 2 with selection of subnet Z'-1 in Chapter 3.

According to Definition 7, we formalize its behavior as following general structure of ETS for subnet Z'(Fig. 2.1)



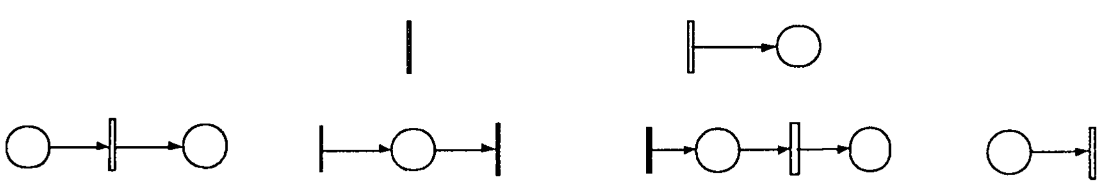
(a) (b)

i = 1,2 .. n (Multisubnet input dimension)

Figure 2.1 General structure modules of the ETS for subnet Z'

In Fig. 2.1, the token enters through the immediate transition t_i^d and is deposited in place p_{i1}^e. The timed transition t_i^e with the firing rate λ_i^e models the equivalent time delay of tokens in the system. p_{i2}^e forms the probabilistic switch with immediate transitions t_{ij}^d. The probabilistic arcs are defined by w_{ij}. Here $i = 1,2.. l$ and $j = 1,2.. m$ are the dimension of Input and Output of the subnet, respectively.

Figure 2.2 and 2.3 show the ETS structure for the single input subnets.



(a) (b) (c) (d)

Figure 2.2 The single input and single output ETS structure

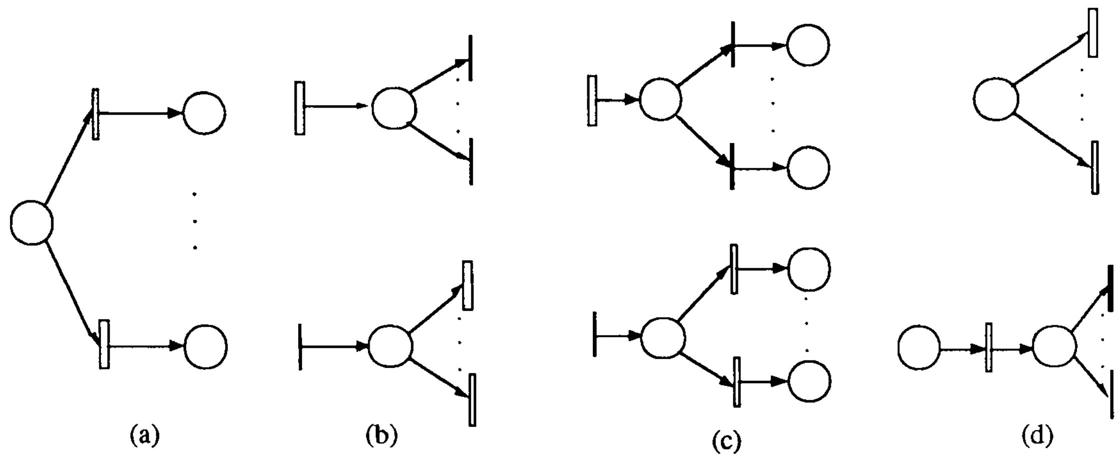


Figure 2.3 The single input and multiple output ETS structure

Definition 11 *[ETS subnet behavior function]*

$$f: (\text{Input}, \text{Output}) \rightarrow W \in R^{l \times m}$$

Where $\sum_{j=1}^{m} w_{ij} = 1, \quad i = 1,2...l$

and there is one and only one nonzero entry per column.

l and m are the input and output dimension of a subnet

The matrix $W = (w_{ij})$ defines the probabilistic arcs.

For example, if a subnet has one input and two output, then $W = [\ w_{11}\ w_{12}\ ]$, and $w_{11} + w_{12} = 1.0$.

If a subnet has two inputs and two outputs, then $w_{11}=1.0$, $w_{22}= 1.0$, $w_{12}=0$, $w_{21}= 0$.

Definition 12 *[Throughput algebra]*

The mathematical operation on throughput is called throughput algebra [27], based on the property of throughput conservation among all transitions in a STPN. This property is the basis of calculating equivalent throughput.

This idea is shown by the example in Figure 2.7 in Section 2.2.

If the subnet is a SPN, we have the throughputs for its every transition. So it is not necessary to do throughput algebra. Let $F^{in} = [f_1{}^{in}, f2^{in} \ldots f_l{}^{in}]$ and $F^{out} = [f_1{}^{out}, \ldots fm^{out}]$ be the throughput vector of $T^{in}(T^{fi})$ and $T^{out}(T^{fo})$ respectively. If the subnet is a GSPN, since only the throughput of the timed transitions in the subnet are available, we must use throughput algebra to calculate the equivalent throughputs.

Without loss of generality, we simply use F^{in} and F^{out} to classify the throughputs. From the conservation of the token flow and throughput for a subnet,we have the following construction rules, according to the above analyses.

1. Initial values with the Input P^{in} (P^d), and T^d at associated PN are P^{in} (P^d) = all possible tokens from 1 to k, and dumb transition firing rate $\lambda^d = \infty$. Here k corresponds to the k-order subnet and k > 0.

2. The general structures of ETS are in Figure 2.1. For any given subnet, one can construct its ETS, based on the general structure.

3. The corresponding associated PN is analyzed in isolation. According to the conservation of the token from the P^{in} (P^d) to P^{out} with the throughput flow from F^{in} and F^{out} we have:

$$\sum_{i=1}^{l} f_i{}^{in} = \sum_{j=1}^{m} f_j{}^{out} \qquad f_i{}^{in} = \sum_{j=n+1}^{n+r} f_j{}^{out} \quad (1 \le i \le l,\ 1 \le j \le m)$$

4. Equivalent parameters: $\forall f_i{}^{in} \in F^{in}$

let (1) $\lambda_i{}^{e} = f_i{}^{in}$ (2) $w_{ij} = \dfrac{f_j{}^{out}}{f_i{}^{in}}$ $(1 \le i \le l,\ n+1 \le j \le n+r)$

Or we can use throughput algebra to find the parameters.

From Definition 8, for K-order subnet, we should have K sets of equivalent parameters.

Figure 2.4 is an example of two-input and three output subnet and its ETS structure. Figure 2.5 is its associate PN.

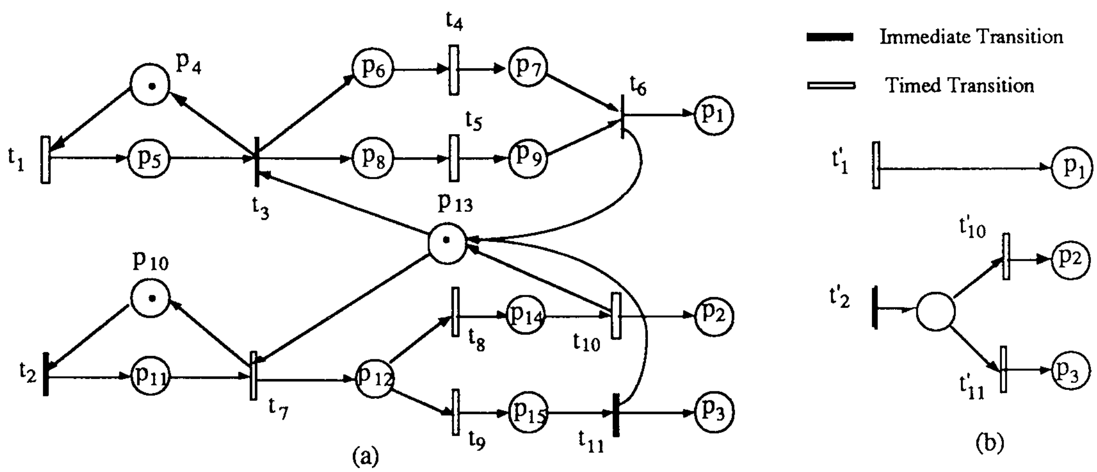


Figure 2.4. An example of the ETS for a multiple-input-multiple-output subnet

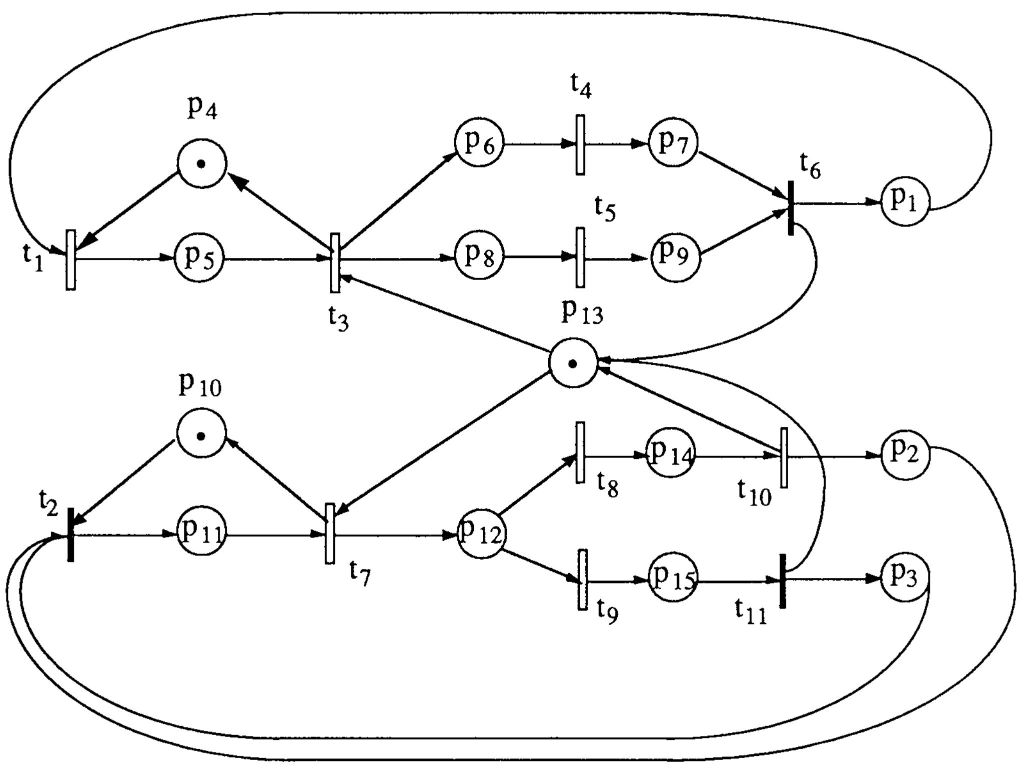


Figure 2.5 The corresponding associated PN in Fig. 2.4

From the above discussion, we can see that for different equivalent structures, we can use different construction rules to derive its parameters.

2.1.4 Reduction and Approximation Evaluation

According to the definitions of subnets, we can select one kind of subnets within the given GSPN. In order to avoid the subnet state explosion problem and reach the best reduction degree, we define the reduction evaluation rules as follows:

Definition 13 *[N step approximation Petri net]*

After using the stepwise reduction and approximation method with one subnet to be approximated, the substituted PN is called one step approximation PN. The subnet is called one step subnet. Similarly, one can have N step's. If one have had the N step approximation PN and do not go further, the N step approximated PN is called Final Approximation PN.

Definition 14 *[Global reduction degree]*

Global reduction degree is defined to compare the state numbers of all subnets with that of the Final Approximation PN. If the state number of Final approximation PN is less than one of those in the subnets, the global reduction degree is defined as negative; otherwise, positive.

This implies that the approximation is required to make the reduction and approximation globally effective.

Definition 15 *[Reduction ratio]*

The reduction ratio refers to the ratio of the original PN state number to the maximum state number we have involved in N step approximation PN. It is defined by:

$$Rr = \frac{No}{Nf} \quad \text{If the global reduction degree is positive}$$

$$Rr = \frac{No}{Nmax} \quad \text{otherwise}$$

Where, Nf means the state number of Final approximation PN

No means the state number of original PN

Nmax means the maximum state number of N step approximation PN

For the state explosion case, we simply define the reduction ratio as infinite, because the state number of the original PN, No is very large. By using computer, it will be overflow.

This comparison is not enough when we have a series of subnets during the evaluation process since they may result in a much larger space than that of the final one, i.e. the global reduction degree is negative. An ideal situation is that the number of states in each subnet is equal to or less than that in the final net. In other words, we need to compare the number of states between the original one and those of the subnets and the final net.

It is true that the reduction ratio is the initial marking dependent in the PN. We denoted it as $R_r(P^{in} = K)$, simply, $R_r(K)$. Here $P^{in} = \{p^{i1}, p^{i2}, \ldots p^{in}\}$, $K = \{k^1, k^2, \ldots k^n\}$, n is the dimension of Input of multisubnet.

Definition 16 *[Performance analysis error]*

Performance analysis error is the relative error in throughput between the original PN and the final approximation PN, which is

$$\text{Error}(\%) = \frac{|\text{exact throughput - approximate throughput}|}{\text{exact throughput}} * 100$$

Definition 17 *[Stepwise Reduction Approximation (SRA)]*

There are three kinds of SRA methods, when N>1. If the N step approximation PN is based on the N independent subnets, the method is called Serial SRA (SSRA). If the N step approximation PN is based on the N dependent subnets, and for any i-1 step (1<i<N) subnet , it also is a subnet of i-step subnet, the method is called Parallel SRA (PSRA). If both SSRA and PSRA are used , Hybrid SRA (HSRA) results.

Using above discussions, we have **reduction and approximation evaluation** as follows:

1. Keep the global reduction degree positive.

2. Decide the step N when global reduction degree changes from positive to negative, if N>1; Otherwise, N=1.

3. Make the reduction ratio as large as possible.

4. Make performance analysis error acceptable to meet the engineering need.

Therefore, the evaluation should make the error small enough and the reduction ratio as large as possible.

2.2 Illustrative Example for Selection and Construction of Subnets

In Figure 2.6, it is a computer system modeled by GSPN [3]. Let us discuss subnets below:

In Figure 2.7 the subnet Z'-1with the place p_{01} forms its associated PN. Where p_{01} is the dumb place. It is easy to verify that it meets the subnet Z^t definition where T^{in} = {t4} and $T^{out} = \{t_{21}, t_{22}\}$. First, we find the throughputs of transitions t_{14} and t_{20}, i.e., $f(t_{14})$ and $f(t_{20})$. Then the parameters in Figure 2.7(a) are obtained as follows according to the throughput algebra.

$$f(t^e{}_{11}) = Pr(t_{21})f(t_{14}) \text{ and } f(t^e{}_{12}) = Pr(t_{22})f(t_{20})$$

and the parameters in Fig. 2.7(b) are

$$f(t^e{}_1) = f(t_9) = f(t_{14}) - f(t_{15}) = f(t_{14}) - Pr(t_{15})f(t_{20})$$

$$Pr(t^d{}_{11}) = \frac{Pr(t_{21})}{Pr(t_{21}) + Pr(t_{22})Pr(t_{17})}$$

$$\text{and } Pr(t^d{}_{12}) = \frac{Pr(t_{22})Pr(t_{17})}{Pr(t_{21}) + Pr(t_{22})Pr(t_{17})}$$

The probability of the immediate transition $t^d{}_1$ remains the same. Note that $Pr(t_{21})$ and $Pr(t_{22})$ are probability of transitions t_{21} and t_{22} in the associated net.

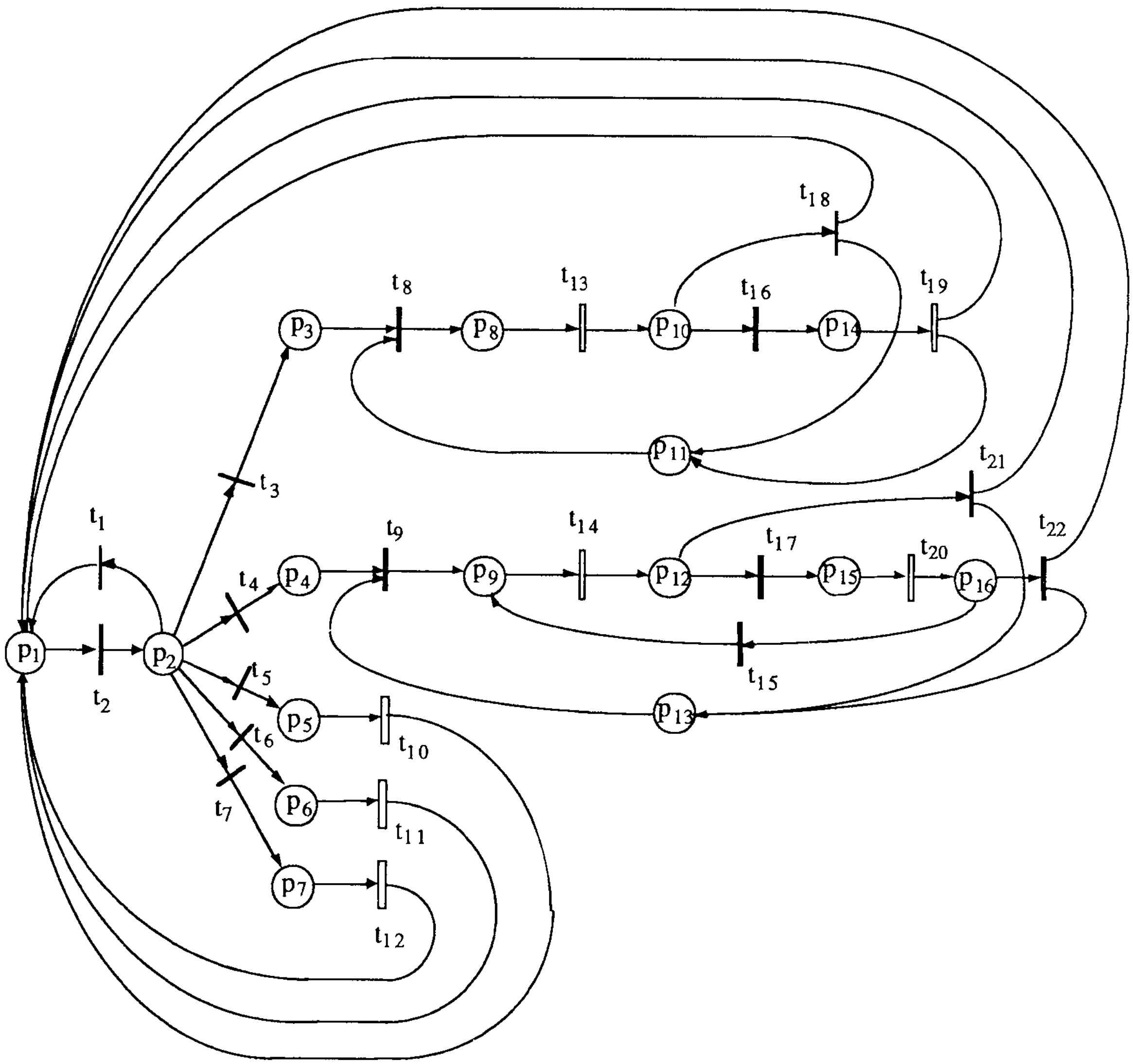


Figure 2.6 A computer system modeled by GSPN

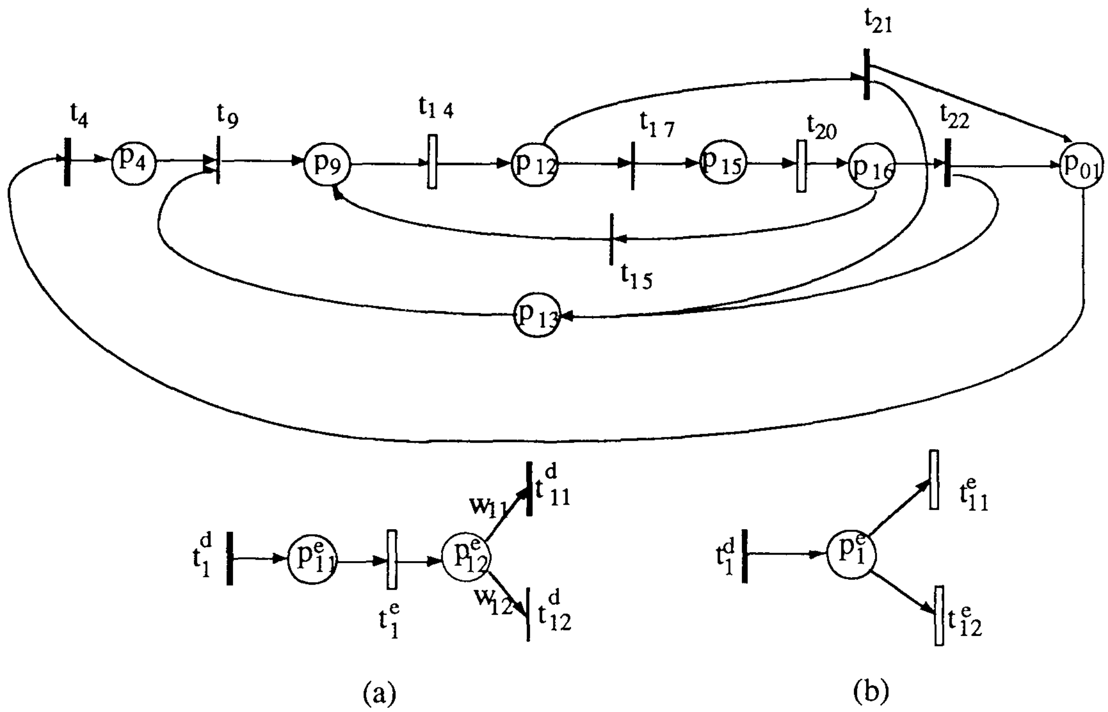


Figure 2.7 The associated PN of transition subnet Z'-1 and its ETS-1

Here we also can construct an ETS-1, based on Z'-1, by assuming the subnet is an GSPN. In this example, $l = 1$, m = 2, then n=0, r=2.

$$f_1^{in} = f_1^{out} + f_2^{out}$$

$$w_{11} = f_1^{out} / f_1^{in}$$

$$w_{12} = f_2^{out} / f_1^{in}$$

Let $\lambda_{14} = \lambda_{20} = 1.0$, by using SPNP [30]:

$f1^{in} = 1/2$ and $f1^{out} = 1/3$, $f_2^{out} = 1/6$

Then we find : $w_{11} = 2/3$, $w_{12} = 1/3$

From the above calculation, we can get the ETS of the subnet Z'-1, based on the general ETS structure as shown in Figure 2.7.

NOTE: If we select the subnet not including t_4 and p_4, it is a subnet Z^t and also 1-order closed subnet. If the selected subnet does not include t_4 , we know it is a complete subnet Z^c; for its associated PN only the dumb arcs are required. Its ETS will be the subnet of Figure 2.7. The immediate transition t_1 in it will be cancelled. And it is also 1-order closed subnet. If the place $p_{01} \in P'$ the subnet will be a place one and its ETS structure will have only p_{11}^e, t_1^e and p_{12}^e. This shows that the subnet selection is flexible.

CHAPTER 3

ALGORITHM AND EXAMPLES

3.1 Stepwise Reduction and Approximation Algorithm

Given a discrete event system, a GSPN Z is modeled and its initial marking is determined. A procedure to derive the results is formulated as follows:

1. If Z can be evaluated with the software packages available, it is done; otherwise,

2. According to the subnet selection rule, identify a subnet Z' while keeping those transitions or places in Z unchanged if they are of special interests.

3. Construct the ETS for this subnet and derive the parameters for ETS based on the throughputs in Z':

a) Find the maximum numbers of tokens possible in the related places,

b) Find the throughput by starting from 1's in the places to the maximum numbers or the numbers whose increase will not change the throughputs of the subnet, and

c) Calculate the parameters in ETS.

If S(Z') cannot be evaluated with software packages, either re-select a subnet or select a sub-subnet in S(Z') and continue this procedure.

4. Let Z" be the net which is the reduced net of Z by replacing Z' with its ETS. Let Z=Z", go to Step 1.

It is clearly that we need to keep the right size of the subnet since a big subnet itself will be difficult to evaluate even though the final net may have a few number of states. The net which satisfies the conditions may not exist. Then we must loosen the conditions at the expense of approximation accuracy.

All discussion on the single input case can be generalized to the subnet with MIMO modules [28].

3.2 Examples

3.2.1 Example 1

Consider a GSPN model in Figure 2.6. In Section 2, we have discussed the subnet Z'-1. We get the equivalent subnet ETS-1 in Figure 2.7. Similarly, we can analyze that the subnet Z'-2 also meets the ETS conditions and has its equivalent subnet ETS-2 in Figure 3.1. We also have a similar discussion (see NOTE in section 2).

For the original PN model, we assume that all the timed transitions are exponentially distributed with firing rates:

$$\lambda_2 = \lambda_{10} = \lambda_{11} = \lambda_{12} = \lambda_{13} = \lambda_{14} = \lambda_{19} = \lambda_{20} = 1.0$$

In the following, we use the GSPN model with different assumptions to show the different cases:

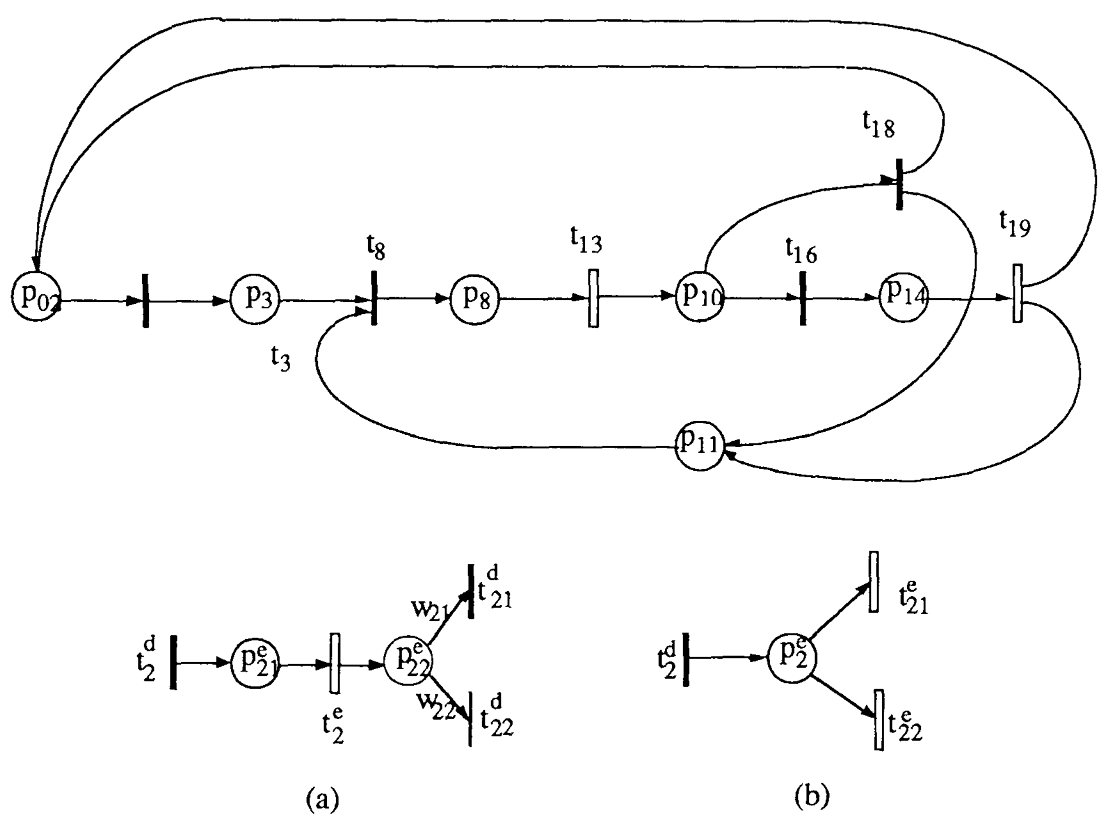


Figure 3.1 The transition subnet Z'-2 and its ETS-2 in Fig. 2.6

CASE 1: SPN model with subnet selection (1)

To use this GSPN model to compare the performance results for a SPN model, we assume the immediate transitions as the timed ones with very large firing rate value ω (that implies no vanishing states in the PN) and define ω to have the following properties;

1. $0 < \lambda << \omega < \infty$
2. If $\omega_1 < \omega_2$, then the transition with firing rate ω_2 will fire before one with ω_1.
3. In a given PN model, ω is set fixed once for all .

Here, λ is the general exponential firing rates, and ω_1, ω_2 are very big positive number.

If the probabilities of probabilistic arcs are w_i , which corresponds to transition t_i, then we can use the firing rate with wi * ω to represent the immediate transition with probability wi. For example, in Figure 2.7, we found the probabilities $w_{11} = 2/3$, and $w_{12} = 1/3$, then the immediate transition t'_{14} and t'_{16} in Figure 3.4, representing t_{11} and t_{12} in the ETS -1, have the firing rate, $\lambda'_{14} = 2/3 * \omega$, $\lambda'_{16} = 1/3 * \omega$

It is easy to verify that $w_{11} = \lambda'_{14} / (\lambda'_{14} + \lambda'_{16})$

By using SPNP we got the performance analysis based on the two PN models, the original one in Figure 2.6 and the approximation PN in Figure 3.2 ~ 3.6. The error is very small [30]. Here we summarize some results as Tables in Appendix 1, corresponding to the approximation ones.

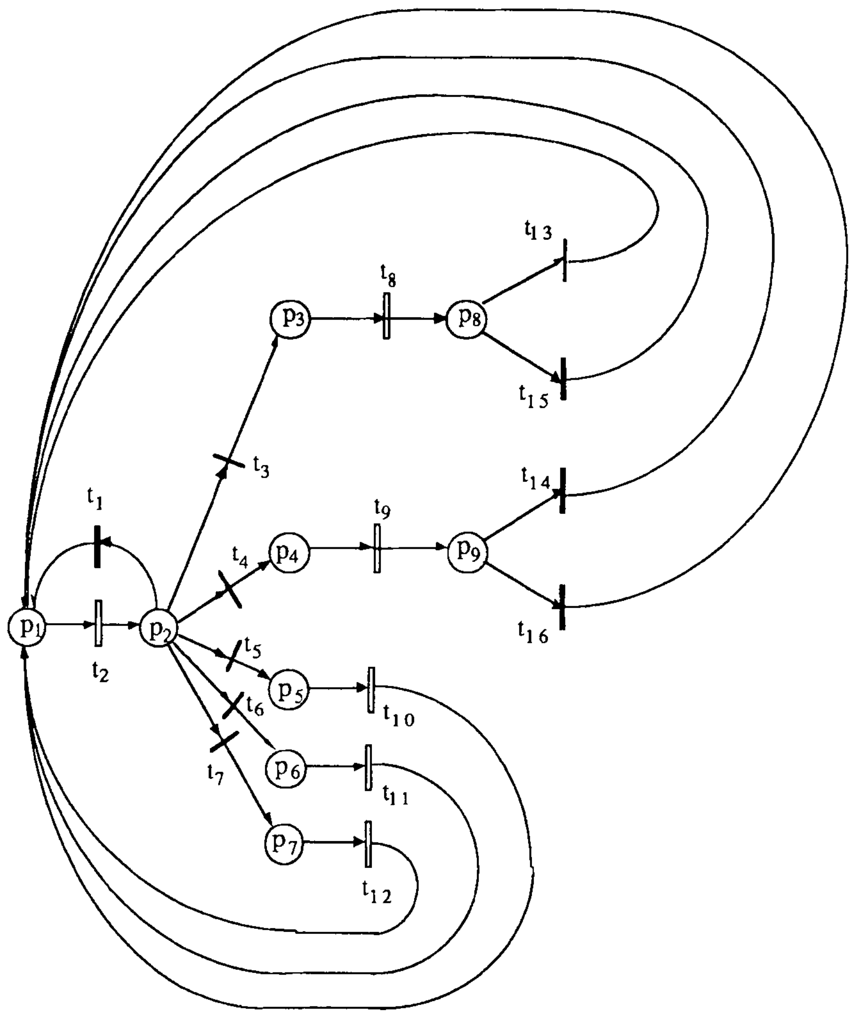


Figure 3.2 The approximation PN -I by substitute of two ETS subnets

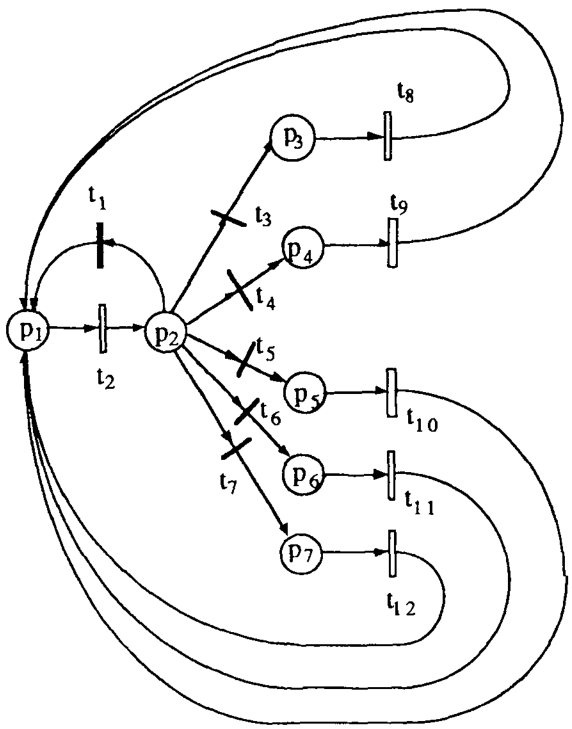


Figure 3.3 The approximation PN - II (2-step)

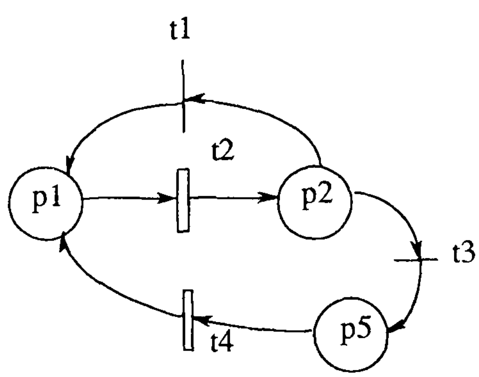


Figure 3.4 The approximation PN model - III (3-step) based on Fig. 3.3

CASE 2: SPN model with subnet selection (2)

In this case, all assumptions are the same as case 1, the difference is the subnet selection method. Here we select subnet Z"-3 and Z"-4 by another way in Figure 3.7 and 3.8. Then we can construct the final approximation F-III PN model in Figure 3.9.

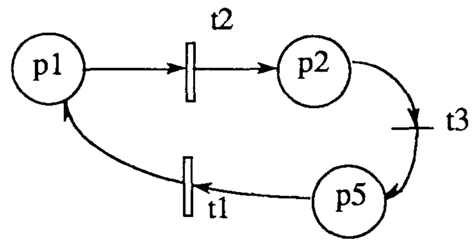


Figure 3.5 The approximation PN model -IV (3-step) based on Fig. 3.4

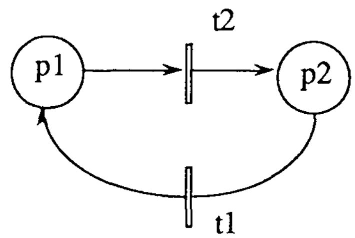


Figure 3.6 The approximation PN model - V (3-step) based on Fig. 3.5

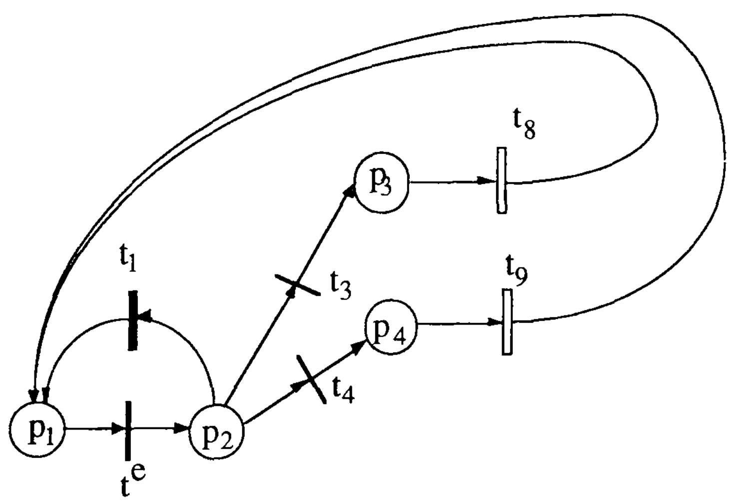


Figure 3.7 The subnet Z"-3 based on Fig. 3.3

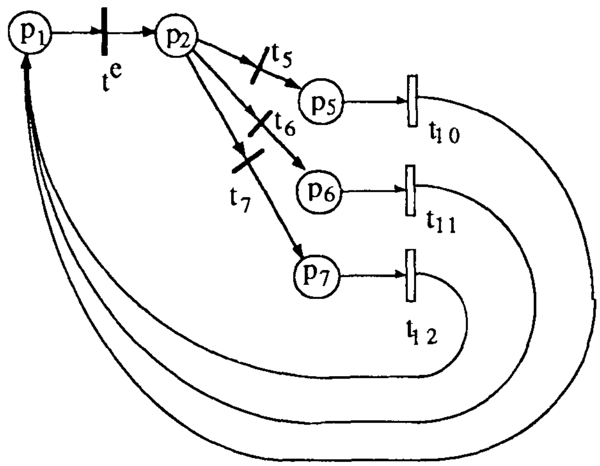


Figure 3.8 The subnet Z"-4 based on Fig. 3.3

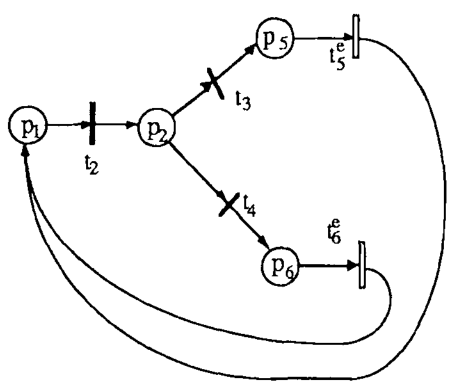


Figure 3.9 Approximation F-III PN based on subnet Z"-3 and Z"-4

CASE 3: GSPN model with subnet selection (1)

In this case, all parameters are the same as above, but we use the GSPN model. The throughput results are the same as before by ignoring the calculating error, but the state number is divided into tangible states and vanishing ones.

CASE 4: GSPN model with subnet selection(2) and different firing rate

In this case, we use the original parameters of transition firing rates in [3], which are as follows:

λ_2 =1/0.2, λ_{10} =1/0.056, λ_{11} =λ_{12} =1/0.036, λ_{13} =1/0.06, λ_{14} =1/0.081, λ_{19} =1/0.058, λ_{20} =1/0.121

And the subnets are selected as Z'-1, Z'-2, Z"-3, and Z"-4.

The performance results by Z'-1, Z'-2 are shown in Table 1-4, and the further results on state number are in Table 6-3.

CASE 5: Comparison of two general equivalent subnet structure

In this case, we use GSPN model to compare the results based on two equivalent subnet structures. Here we compare only that based on the approximation PN model-I in Table 2.2. Two GSPN ETS equivalent structure structures have the same performance results and tangible states, but the latter in Figure 3.1 has the fewer vanishing states.

CASE 6: Comparison of the GSPN and its revised one

In this case, we assume the GSPN model in Fig.2.6 as follows:

Transitions t_{19} and t_{21} are connected to place p_2, not p_1, which will show that the branches in the equivalent subnet structure must be decided [30].

3.2.2 Result Analysis 1

According to the assumptions on input values, we have the following results:

1. In Example 1, for transition subnet Z'-1 (and Z'-2), the approximation error can be nearly ignored, especially when m(p1) = 1 as shown in Table 1.1 and 1.2.

Because, for this case, the all reduction rules defined in Section 2, are well satisfied, we can simply say both, the original one and the reduced one, are equivalent.

2. Using the SRA method, we reach the following conclusions:

(1) With the initial tokens increasing, the reduction ratio will be fast decreased in Table 2. When it is beyond some value (in Table 2, m(p1) ≥ 8), the original will be overflow. But we can easily find excellent results by using SRA.

Generally speaking, if the PN consists of n sets of subnets Z'-1 or Z'-2, the method can be efficiently generalized to be used. We also can make sure the error is very small.

(2) By using the SRA method, we must keep the global reduction degree positive. Otherwise, the approximation may lose the power. In Example 1, although the Final approximation PN state number is reduced, by 3-step approximation PN, the subnet state number is higher than it in Table 6.1, meaning the global reduction degree is negative. So this step approximation is not necessary. For this example, only two step approximation meet the reduction rules, N=2.

We can also select different subnets: in CASE 2, we selected subnet Z"-3 and Z"-

4. The performance error is almost the same as that in CASE1, but the reduction ratio has significantly increased in Table 6.2. It is 23.3 at m(p1)=5, and when m(p1) = 15 - 30, the Original PN is overflowed. But we can also easily have the results by Approximation F - III, the state number of which is 816 - 5456.

(3) For dumb immediate transition t_i^d in ETS model, if it is also involved with other immediate ones, generally the probabilistic arcs must be defined. Otherwise the approximation error will be large.

In this Example, we set the same large value for immediate transitions, meaning their probabilities are same. After approximation, for example, the 3-step approximation PN model-III (using subnet Z'-3), the probabilities of immediate transitions t1 and t3 in ETS, corresponding to the t_1, and t'_3,t'_4,t_5,t_6,t_7 in the original, are 1/6 and 5/6 in Table 3.1. If one using 1/5, 4/5, the error will be large. in Table 3.2. Particularly if one using 1/2, 1/2, the error will be more than 20%.

(4) One important property of SRA is that if the subnet is K-order closed subnet, for example, subnet Z'-1, (if t_4 is not immediate, the Z'-1 is not k-order subnet), and $m(p_{01}) \geq K$, the equivalent throughput will not be constant. Here k = 1, for any initial marking in original PN, only one set of ETS is needed. In this situation, the approximation will be much simpler and the error will be very small. In other words, the global reduction degree will be always positive in Table 2. (for marking dependent, see Table 4).

We also have other approximation PN in Figure 3.5 and 3.6, to show the approximation method using subnet Z'-4 in Table 5.

3. The throughput of a transition will approach its initial firing rate with the increasing of initial token. In our example, when $m(p_1) > 10$, the throughput of t_2 will converge to the initial firing rate of transition. Thus one can use this property to inspect the results by using SPNP.

4. The paper [20], [21], [10] have stated the conditions for reducing the general PN. If the time is assumed to be involved, the advantages there will be fully taken by using SRA here.

5. For performance analysis, generally speaking, we must go further from the equivalent subnet to some approximation; otherwise, the PN may be intractable. So the approximation will be a must but the error will be larger than what we have here, to some extent.

3.2.3 Example 2

The FMS system considered in [14] is shown in Figure 4.1, which comprises a load / unload system (L/U), a machine center (M/C), a head changer (H/C), and a vertical turret lathe (VTL). The system produces two types of parts, each part being mounted on a fixture at L/U and carried between the machines by a conveyor system. The part routing and mean service times at each station are shown in Fig. 4.2. respectively. It can be seen that each part type has two alternative routing. For example, a part of type 1 is loaded into the system and transported to M/C, then to the VTL, and finally unloaded. The alternative routing is L/U, Conveyor, H/C, Conveyor, VTL, conveyor, and L/U. The routing probabilities are 0.15 for the first route and 0.85 for the second.

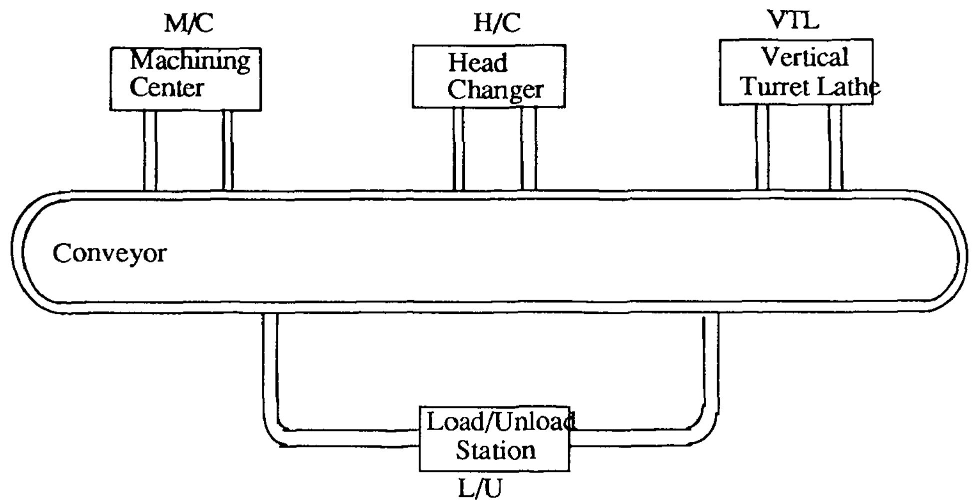


Figure 4.1 Layout of a flexible manufacturing system with three machines

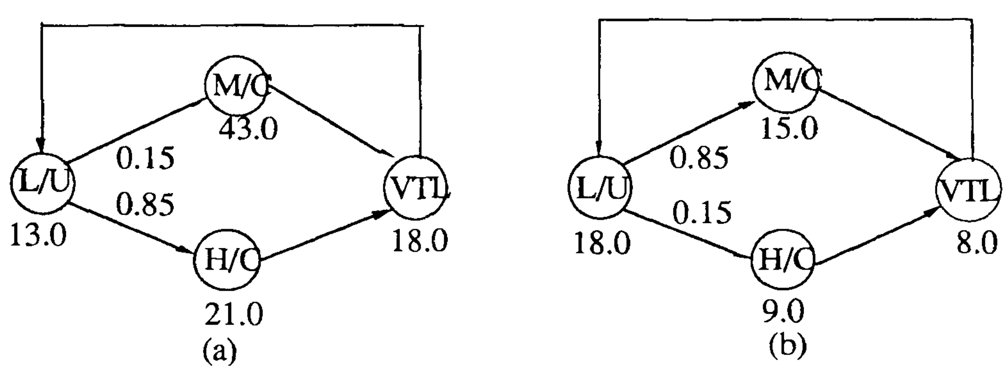


Figure 4.2 Flow sequence of parts in the FMS with routing probabilities and mean processing times: (a) part type 1, (b) part type 2.

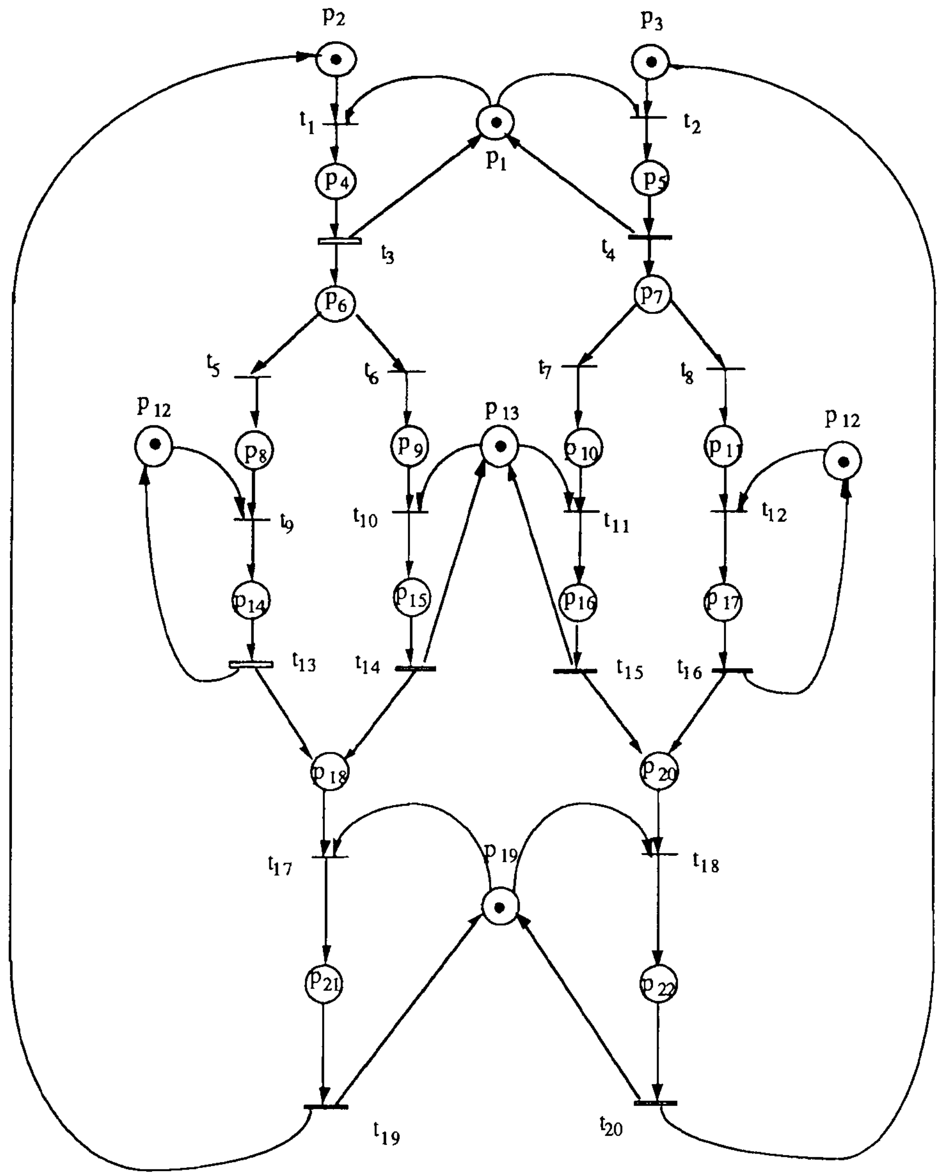


Figure. 4.3 GSPN model of the FMS example

Under the assumption that each machine service time is exponentially distributed with mean equal to the sum of the means of the machine and the conveyor time and there are a limited number of fixtures in the system but enough buffer space at each machine. Figure 4.3 shows a GSPN model for above system.

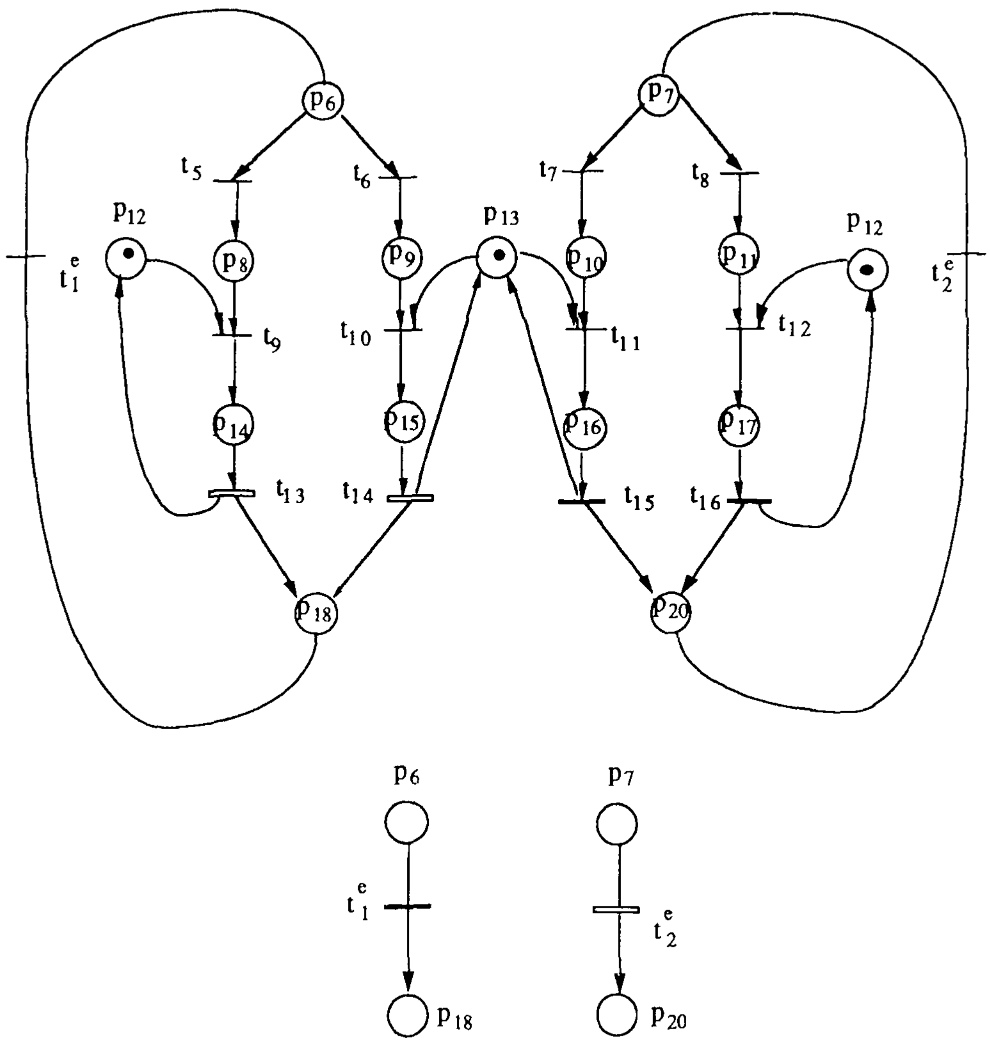


Figure. 4.4 The place subnet Z'-1 and its ETS-1 in Fig. 4.3

By using SPNP, the performance analysis results of GSPN for this system are shown in Table7, 8, and 9. For other probabilistic arcs from the places, the probabilities are assumed equal to each other.

3.2.4 Result Analysis 2

1. Performance analysis based on the GSPN model in Figure 4.3.

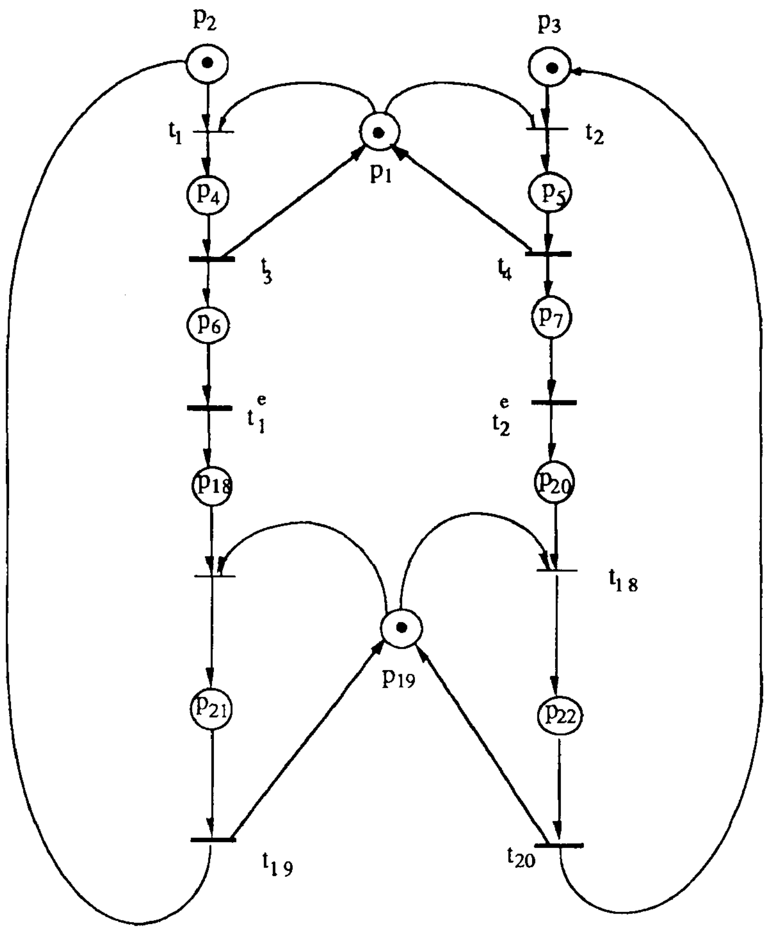


Figure 4.5 Approximation PN (1-step) model with subnet Z'-1

Table 7 shows the results for three different cases; Case1, there is one fixture for type 1 and type 2; Case2, two fixtures for type 1 and one fixture for type 2; Case3, two fixtures for type 2 and one fixture for type 1. It is to be noted that the GSPN is the same but only the initial marking is changed to initialize the number of fixtures, similarly to compute the performance measures for changing the other parameters such as machine speed and loading time.

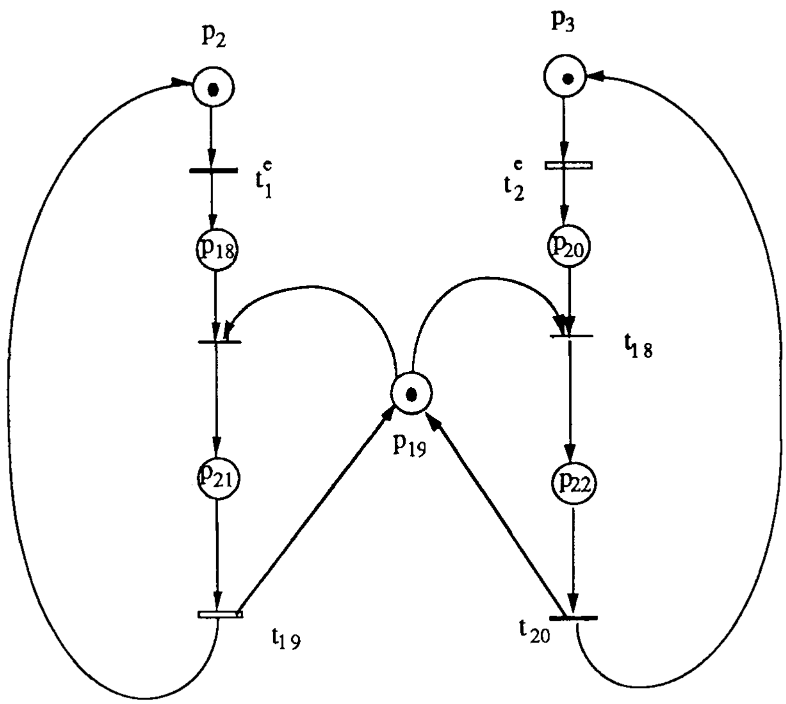


Figure 4.6 Approximation PN (2-step) with subnet Z'-2

Table 8 shows the results in [14] which are different from the throughputs by using SPNP shown in Table 7 but the others are almost the same.

Table 9 shows the results by using the SRA method to approximate the PN model. The error is small, and the state space is reduced. Figure 4.4 is a place subnet Z'-1 and its ETS-1, and Figure 4.5 is the corresponding 1-step approximation PN model. Meanwhile, as the subnet Z'-1 is 2 -order closed subnet, its equivalent throughput will be initial marking independent when the initial marking ≥ 2. If one uses the same throughput as that at initial marking 1, the results will have the bigger error. By using the SRA method based on 1-step approximation PN with Z'-1, we have the 2-step approximation PN with subnet Z'-2 in Figure 4.6.

If we use only 1-step to get the approximation PN in Figure 4.6 rather than 2-step one, the performance error is small. The global reduction degree, however, is very negative, which means the subnet is of large state space. According to the reduction evaluation rules, approximation is insignificant.

From now on, we simply use the SPN model to represent the GSPN for easily comparing the state number which is the sum of tangible ones and vanishing ones. The following results are by running GSPN model. The performance measures are the same as those before; the states consist of tangible and vanishing ones. In this case, the tangible state number is small.

Again, if 2-order closed subnet is considered here, the reduction ratio is upto 178.

According the reduction rules, the approximation error should be reasonably small. We can easily verify that the performance approximation error will be increased with the approximation step N. Generally speaking, for this case, the error will be large, i.e., the reduction rule 3 is weakly satisfied. The reason is that the subnet Z'-1 here is closely coupled one, but we still simply use the decoupled throughput equivalent subnet ETS-1 to approximation it. Therefore the error will be large.

Here one can find that in general the reachability set of a GSPN is a subnet of the reachability set of the non timed PN, because in GSPN, precedence rules introduced with immediate transitions do not allow some states to be reached. However, the reachability set of SPN is the same as for the non timed PN. Therefore, the reachability set of GSPN is divided into two disjoint subnets, one of which comprises markings that enable exponentially distributed transitions only, while the other comprises markings that enable immediate transitions. We called a state or marking of the former type tangible state and a state of the latter type vanishing one. Let SS indicate the state space which can be partitioned the tangible states, denoted as TS, and vanishing states, denoted as VS, then we have:

$$SS = TS \cup VS \text{ and } TS \cap VS = \varnothing$$

2. Performance analysis based on the following assumption

In Figure 4.3, we assume place p_{12} is the input place of transition t_9 and t_{10} , and the output place of transition t_{13} and t_{14}; place p_{13} is the input place of transition t_{11} and t_{12}, and the output place of transition t_{15} and t_{16}.

Then, the subnet Z'-1 in Figure 4.4 will be changed to the decoupled one. In this case, the performance analysis is good [30].

From the thesis, we claim that the approximation error can be monitored well by the following two ways:

1) If the subnet is loosely coupled or decoupled one, the error will be smaller and smaller. Particularly in decoupled case, the error is very small.

2) For a subnet, by calculating the variance for the corresponding subnet throughput, if the variance is small, the approximation error will be small [30].

By this way, one can forecast the error to meet the reduction evaluation rules. Anyway, for a real PN model, the k-order closed subnet often exists. Thus, the approximation method will be powerful in the case with the initial marking increasing.

CHAPTER 4

CONCLUSION REMARK AND FURTHER RESEARCH

The given examples show that using the method to approximate the large PN model can reduce the state space to make the analysis possible. The performance analysis errors are small. From this thesis, given a PN model, we can do follows by using the approximation methods:

1. At some conditions, one should try to simplify the PN model first in order to analyze it. For example, we can do some reduction on no-timed subnet, therefore, the vanishing state of the Petri net is reduced.

2. Based on the simplified PN model, one can use SPNP software to conduct performance analysis.

From this thesis, the conclusion is that the approximation error is dependent on the approximation degree i.e. approximating the original PN to what stage. The advantages of PN reduction method in [10],[20],[21] can be fully taken in this approach with the reduction error almost zero. For the other approximation case, the error will be dependent on the approximation degree.

3. Any present methods, including Numerical Method for a stochastic PNs and MGF method for an extended PNs, it was increasingly limited by the situation: for the former one, it may have the reachability graph explosion problem which makes the analysis more difficult or impossible, and it also cannot solve the one with non-exponential transition firing distribution; for the latter one, it is of limited use in an extended stochastic PN. In fact, it will be impossible even for a simple PN model with large reachability graph, or for the ASPN, it is only theoretically true to find a closed form performance analysis.

4. For the performance analysis of a given PN model, we first need to decide which methods are more efficient or possible. No matter what they are, we can use the

approximation method to conduct performance analysis theoretically or practically. The key is that the approximation method is required to meet the given conditions that make the error acceptable. The theoretical conditions are under study.

5. For some complicated system PN model, perhaps we cannot get the exact result by using the conventional methods. Thus studying the approximation method is a way to reach the purpose of performance analysis.

Summarizing, the approximation method has some key concepts involved:

1. Subnet selection: different subnet selection will lead to the different approximating process and the different approximating accuracy. - **Flexible**

2. Marking dependent: the equivalent subnet throughput is marking dependent. - **Dynamic**

3. K-order subnet property: the subnet has same throughput if some initial marking number is larger than a real number K. -**Switching**

4. Throughput algebra: in order to calculate the equivalent subnet throughput, some mathematical operation is needed, based on the throughput observation. - **Conservation**

5. Global reduction degree: based on flexible subnet selection, we should keep all subnets that are of smaller state space than final approximation PN. -**Efficiency**

6. Reduction, substitution and decomposition: for any original PN, it is necessary to do some reduction and decomposition to allow the method to meet the given conditions and to have less state space. Sometimes, the substitution of some subnet or transition by the equivalent one is needed. - **Equivalent**

Performance analysis plays important roles in using Petri net. Generally it is very difficult to conduct performance analysis for a timed Petri net and property analysis for an ordinary Petri net, if its state spaces is too large. Thus, the reduction and approximation may be the only cost-effactive solutions [25],[28]. A Stepwise Reduction and Approximation Method for GSPN is given under some conditions. Particularly, if we are interested only in some important performance measures and hope to have their closed form results, we can combine the approximation method and MGF approach to reach the aim [24]. From this paper, we know that for the cases that do not meet the given conditions, the error will be large and depends on approximation degree. Furthermore, if the subnet is independent from the point of structure but dependent on the parameters, the method may not work. Anyway, for those situations, the method is limited. Therefore, we are trying to do something along the direction, including the theoretical proof, to solve those problems.

In the future, we will also do some work to loose some conditions, such as subnet selection Rule 1, which means independent subnet. we can use some decouplling or decomposition method to equivalent and approximate the Petri nets, in order to reach the desired results. Another way is to use "throughput subnet" [2] thinking way through finding the variance of the subnet throughput. Based on that, we can reconstruct the equivalent throughput subnet to reduce the error.

APPENDEX. SPNP Program and Running Results

Table 1.1

Comparison of the Numerical Results by using SRA Algorithm (The initial token in place p1 m(p1) = 1)			
Throughput	Original PN	Approximation PN -I	Error %
t1	7.999999140446e-02	8.00000000048e-02	0.00001
t2	4.799999580464e-01	4.80000000288e-01	0.00002
t10	7.999999300773e-02	8.00000000048e-02	0.00001
t11	7.999999300773e-02	8.00000000048e-02	0.00001
t12	7.999999300773e-02	8.00000000048e-02	0.00001

Table 1.2

Comparison of the Numerical Results by using SRA Algorithm (The initial token in place p1 m(p1) = 5)			
Throughput	Original PN	Approximation PN -I	Error %
t1	1.633482547424e-01	1.63218720241e-01	0.079
t2	9.800895387395e-01	9.79312352733e-01	0.079
t10	1.633482560739e-01	1.63218718368e-01	0.079
t11	1.633482560739e-01	1.63218718368e-01	0.079
t12	1.633482560739e-01	1.63218718368e-01	0.079

Table 1.3

Subnets Z'-1 and Z'-2 Numerical Results for Throughput Equivalent Subnets			
Z'-1 , m(p01) >= 1 (throughput)		Z'-2 , m(p02) >= 1 (throughput)	
t4	0.5000	t3	0.6667
t21	0.3333	t18	0.3333
t22	0.1667	t19	0.3334

Table 1.4

Subnets Z'-1 and Z'-2 Throughputs (with CASE 4: GSPN)			
Z'-1 , m(p01) >= 1 (throughput)		Z'-2 , m(p02) >= 1 (throughput)	
t9=t14-t15	5.3	t8=t13	11.23
t14	7.07	t13	11.23
t20	3.53	t19	5.62

Table1.5

Comparison of Original and ApproI,II throughputs (with CASE 4: GSPN)			
m(p1)	(Original) throughput t2	(Approximation) throughput t2	Error(%)
5	4.99852	4.99866	0.003
7	4.9999576	4.99997	0.0004
10	-----	4.999999	-----

Table 2.1

Comparison on State Numbers at the different initial tokens						
Initial token m(p1)		1	2	3	5	7
Number of states	Original	14	89	364	2940	13728
	Sub.Z'-1	6	not needed, since it is 1-order closed			
	Sub.Z'-2	5	not needed, since it is 1-order closed			
	Appro.I	9	49	165	1287	6435
	Appro.II	7	28	84	462	1716
Reduction ratio		2.0	3.18	4.33	6.36	8.0

Table 2.2

Comparison of state number in Approximation PN I by using two ETS structures (tangible + vannishing states)					
ETs structure (a)	6+3	21+18	56+63	252+378	792+1386
ETs structure (b)	6+1	21+6	56+21	252+126	792+462

Table 3.1

Comparison of the Numerical Results by using SRA Algorithm (Throughput of t2 when different m(p1))			
m(p1)	Original PN	Approximation - III	Error %
1	4.799999e-01	4.79991e-01	0.0002
2	7.45515e-01	7.92033e-01	6.18
4	9.50442e-01	9.76236e-01	2.19
5	9.80090e-01	9.89311e-01	0.95
7	9.97119e-01	9.97808e-01	0.55

Table 3.2

(If the approximation PN with error on probabilistic arcs)

Comparison of the Numerical Results by using SRA Algorithm (Throughput of t2 when different m(p1))			
m(p1)	Original PN	Approximation - III	Error %
1	4.799999e-01	4.90186e-01	2.12
2	7.45515e-01	8.02626e-01	7.66
4	9.50442e-01	9.34157e-01	1.71
5	9.80090e-01	9.64143e-01	1.30
7	9.97119e-01	9.84544e-01	1.26

Table 4

Subnets Z'-3 and Z'-4 Numerical Results for Throughput Equivalent Subnets			
Z'-3		Z'-4	
m(p03)	Throughput	m(p04)	Throughput
1	0.7692	1	0.9231
2	1.2621	2	1.5146
3	1.5969	3	1.9163
4	1.83291	5	2.4041

Table 5.1

(If the approximation PN with immediate transition)

Comparison of the Numerical Results by using SRA Algorithm (Throughput of t2 when different m(p1))			
m(p1)	Original PN	Approximation - IV	Error %
1	4.799999e-01	4.80009e-01	0.0001
2	7.45515e-01	7.92033e-01	6.23
4	9.50442e-01	9.26170e-01	2.55
5	9.80090e-01	9.53519e-01	2.71
7	9.97119e-01	9.80719e-01	1.65

Table 5.2

(If the approximation PN by place subnet only)

Comparison of the Numerical Results by using SRA Algorithm (Throughput of t2 when different m(p1))			
m(p1)	Original PN	Approximation - V	Error %
1	4.799999e-01	4.79979e-01	0.004
2	7.45515e-01	7.9211e-01	6.25
5	9.80090e-01	9.9269e-01	1.30
7	9.97119e-01	9.9874e-01	0.17
10	overflow	9.9999e-01	No

Table 6.1

Comparison on State Numbers at the different initial tokens						
Initial token m(p1)		1	2	3	5	7
State Numbers	Original	14	89	364	2940	13728
	Sub.Z'-3	7	28	84	462	1716
	Sub.Z'-4	7	28	84	462	1716
	Appro.III IV	3	6	10	21	36
	Appro.V	2	3	4	6	8

Table 6.2

Comparison on State Numbers at the different initial tokens (CASE2)						
Initial token m(p1)		1	2	3	5	7
State Numbers	Original	14	89	364	2940	13728
	Sub.Z"-3	4	10	20	56	120
	Sub.Z"-4	5	15	35	126	330
	ApproF.III	5	10	20	56	120
Reduction ratio		2.8	5.93	10.5	23.3	42

Table 6.3

Comparison on State Numbers at the different initial tokens(CASE 4)							
Initial token m(p1)		1	2	3	5	7	10
State Numbers	Original	8+6	34+41	104+ 157	560 +1069	1968+ 4270	8294+ 20163
	Sub.Z'-1	2+4	N/A	N/A	N/A	N/A	N/A
	Sub.Z'-2	2+3	N/A	N/A	N/A	N/A	N/A
	Sub.Z"-3	2+2	3+7	4+16	6+50	8+112	N/A
	Sub.Z"-4	3+2	6+9	10+25	21+105	36+294	N/A

Table 7

(Here Ni denotes the number of fixtures of type i, i = 1, 2)

Performance measures for the FMS example for 3 situations: Ni, i=1, 2		N1 = 1 N2 = 1	N1 = 2 N2 = 1	N1 = 1 N2 = 2	N1=5 N2=5
Measure 1	Machine utilizations (By probabilities)				
L/UL	Prob(p4,1)+Prob(p5,1)	0.5283	0.5884	0.6972	N/A
M/C	Prob(p14,1)+Prob(p17,1)	0.3304	0.3468	0.4559	N/A
H/C	Prob(p15,1)+Prob(p16,1)	0.3055	0.4723	0.2850	N/A
VTL	Prob(p21,1)+Prob(p22,1)	0.4289	0.5717	0.4783	N/A
Measure 2	Fixture Utilizations (By probabilities)				
Type 1	1 - Prob (p2, 1)	0.9152	0.7976	0.8254	N/A
Type 2	1 - Prob(p3,1)	0.9121	0.8729	0.6107	N/A
Measure 3	Buffer occupancies				
L/UL	ET(P2) + ET(p3)	0.1726	0.3614	0.5640	N/A
M/C	ET(P8) + ET(p11)	0.0700	0.1191	0.2174	N/A
H/C	ET(P9) + ET(p10)	0.0219	0.1699	0.0357	N/A
VTL	ET(P18) + ET(p20)	0.1366	0.3701	0.2661	N/A
Measure 4	Throughput rates (Number of parts per hour)				
Type 1	TR(19)	0.9527	1.5224	0.8272	N/A
Type 2	TR(20)	1.0730	0.8622	1.7271	N/A

Table 8

(Here Ni denotes the number of fixtures of type i, i = 1, 2)

Performance measures for the FMS example for 3 situations: Ni, i=1, 2		N1 = 1 N2 = 1	N1 = 2 N2 = 1	N1 = 1 N2 = 2	(other)
Measure 1	Machine utilizations (By probabilities)				
L/UL	Prob(p4,1)+Prob(p5,1)	0.5284	0.5886	0.6975	N/A
M/C	Prob(p14,1)+Prob(p17,1)	0.3305	0.3472	0.4559	N/A
H/C	Prob(p15,1)+Prob(p16,1)	0.3071	0.4176	0.2845	N/A
VTL	Prob(p21,1)+Prob(p22,1)	0.4278	0.5717	0.4783	N/A
Measure 2	Fixture utilizations (By probabilities)				
Type 1	1 - Prob (p2, 1)	0.9152	0.7655	0.8252	N/A
Type 2	1 - Prob(p3,1)	0.9123	0.8727	0.6103	N/A
Measure 3	Buffer occupancies				
L/UL	ET(P2) + ET(p3)	0.1728	0.3617	0.1728	N/A
M/C	ET(P8) + ET(p11)	0.0702	0.1196	0.2177	N/A
H/C	ET(P9) + ET(p10)	0.0255	0.1695	0.0356	N/A
VTL	ET(P18) + ET(p20)	0.1366	0.3702	0.2662	N/A
Measure 4	Throughput rates (Number of parts per hour)				
Type 1	TR(19)	0.6484	0.8461	0.4541	N/A
Type 2	TR(20)	0.7907	0.5649	1.1029	N/A

Table 9

(Here Ni denotes the number of fixtures of type i, i = 1, 2)

Performance measures for the FMS example for 3 situations: Ni, i=1, 2		N1 = 1 N2 = 1	N1 = 2 N2 = 1	N1 = 1 N2 = 2	Error % (Max.)
Measure 1	Machine utilizations (By probabilities)				
L/UL	Prob(p4,1)+Prob(p5,1)	0.5161	0.5875	0.6931	0.43
M/C	Prob(p14,1)+Prob(p17,1)	N/A	N/A	N/A	N/A
H/C	Prob(p15,1)+Prob(p16,1)	N/A	N/A	N/A	N/A
VTL	Prob(p21,1)+Prob(p22,1)	0.4174	0.5806	0.4710	2.43
Measure 2	Fixture Utilizations (By probabilities)				
Type 1	1 - Prob (p2, 1)	0.9228	0.7962	0.8363	3.85
Type 2	1 - Prob(p3,1)	0.9283	0.8864	0.6674	8.56
Measure 3	Buffer occupancies				
L/UL	ET(P2) + ET(p3)	0.1489	0.3482	0.5293	13.73
M/C	ET(P8) + ET(p11)	N/A	N/A	N/A	N/A
H/C	ET(P9) + ET(p10)	N/A	N/A	N/A	N/A
VTL	ET(P18) + ET(p20)	N/A	N/A	N/A	N/A
Measure 4	Throughput rates (Number of parts per hour)				
Type 1	TR(19)	0.9236	1.5682	0.7887	4.65
Type 2	TR(20)	1.0531	0.8261	1.7577	4.19

```
#include "user.h"

/* 1-1-92 This is a SPNP file for my M.S. THESIS,
 (( MAIN PROGRAMME FOR USING SPNP with GSPN model))  From
N.Viswanadham*/


probability_type prb1 = 1.0;
probability_type prb2 = 0.15;
probability_type prb3 = 0.85;

int Z1,Z2;


parameters() {
      /*iopt(IOP_PR_FULL_MARK, VAL_YES);
          iopt(IOP_PR_MC,VAL_YES);
          iopt(IOP_PR_RGRAPH,VAL_YES);
          iopt(IOP_PR_PROB,VAL_YES);*/
        iopt(IOP_METHOD,VAL_SSSOR);
        iopt(IOP_PR_MARK_ORDER,VAL_CANONIC);
        iopt(IOP_PR_MC_ORDER,VAL_TOFROM);

        Z1 = input ("initial tokens of place p2 (from 1 to 5):");
        Z2 = input ("initial tokens of place p3 (from 1 to 5):");

}
net() {
        place("p1"); init("p1",1);
        place("p2"); init("p2",Z1);
        place("p3"); init("p3",Z2);
        place("p4");
        place("p5");
        place("p6");
        place("p7");
        place("p8");
        place("p9");
        place("p10");
        place("p11");
        place("p12"); init("p12",1);
        place("p13"); init("p13",1);
        place("p14");
        place("p15");
        place("p16");
        place("p17");
        place("p18");
```

```
        place("p19"); init("p19",1);
        place("p20");
        place("p21");
        place("p22");


          (void) trans("t1");   priority("t1",10);   probval
("t1",prb1);
          (void) trans("t2");   priority("t2",10);   probval
("t2",prb1);

          trans("t3");  priority("t3",1);    rateval("t3",1.0*60/
13.0);
          trans("t4");  priority("t4",1);    rateval("t4",1.0*60/
18.0);

        (void) trans("t5");    priority("t5",10);   probval
("t5",prb2);
          (void) trans("t6");    priority("t6",10);   probval
("t6",prb3);
          (void) trans("t7");    priority("t7",10);   probval
("t7",prb2);
          (void) trans("t8");    priority("t8",10);   probval
("t8",prb3);
          (void) trans("t9");    priority("t9",10);   probval
("t9",prb1);
          (void) trans("t10");   priority("t10",10);  probval
("t10",prb1);
          (void) trans("t11");   priority("t11",10);  probval
("t11",prb1);
          (void) trans("t12");   priority("t12",10);  probval
("t12",prb1);

       trans("t13");    priority("t13",1);   rateval("t13",1.0*60/
43.0);
          trans("t14");    priority("t14",1);
rateval("t14",1.0*60/21.0);
          trans("t15");    priority("t15",1);
rateval("t15",1.0*60/9.0);
          trans("t16");    priority("t16",1);
rateval("t16",1.0*60/15.0);

        (void) trans("t17");    priority("t17",10);   probval
("t17",prb1);
          (void) trans("t18");    priority("t18",10);   probval
("t18",prb1);
```

```
          trans("t19");     priority("t19",1);
rateval("t19",1.0*60/18.0);
          trans("t20");     priority("t20",1);
rateval("t20",1.0*60/8.0);


        iarc("t1","p1"); oarc("t1","p4");
        iarc("t1","p2");
        iarc("t2","p1"); oarc("t2","p5");
        iarc("t2","p3");
        iarc("t3","p4"); oarc("t3","p1");
                         oarc("t3","p6");
        iarc("t4","p5"); oarc("t4","p1");
                         oarc("t4","p7");
        iarc("t5","p6"); oarc("t5","p8");
        iarc("t6","p6"); oarc("t6","p9");
        iarc("t7","p7"); oarc("t7","p10");
        iarc("t8","p7"); oarc("t8","p11");
        iarc("t9","p12");  oarc("t9","p14");
        iarc("t9","p8");
        iarc("t10","p9");
        iarc("t10","p13"); oarc("t10","p15");
        iarc("t11","p10");
        iarc("t11","p13"); oarc("t11","p16");
        iarc("t12","p11"); oarc("t12","p17");
        iarc("t12","p12");
        iarc("t13","p14"); oarc("t13","p12");
                           oarc("t13","p18");
        iarc("t14","p15"); oarc("t14","p13");
                           oarc("t14","p18");
        iarc("t15","p16"); oarc("t15","p13");
                           oarc("t15","p20");
        iarc("t16","p17"); oarc("t16","p12");
                           oarc("t16","p20");
        iarc("t17","p18"); oarc("t17","p21");
        iarc("t17","p19");
        iarc("t18","p19");
        iarc("t18","p20");
                           oarc("t18","p22");
        iarc("t19","p21"); oarc("t19","p2");
                           oarc("t19","p19");
        iarc("t20","p22"); oarc("t20","p19");
                           oarc("t20","p3");
```

```
/*The  net is defined, and the analysis can be conducted */
}

/* the following three lines should appear in all programs */
assert() {return(RES_NOERR);}
ac_init() {}
ac_reach() {fprintf(stderr,"/nThe reachibility graph has been
generated/n/n");}

/* User-defined output functions */

reward_type ef2() {return(rate("t3"));}
/* throughput  of t3*/
reward_type ef3() {return(rate("t4"));}
/* throughput  of t4*/
reward_type ef12() {return(rate("t13"));}
/* throughput  of t13*/
reward_type ef13() {return(rate("t14"));}
/* throughput  of t14*/
reward_type ef14() {return(rate("t15"));}
/* throughput  of t15*/
reward_type ef15() {return(rate("t16"));}
/* throughput  of t16*/
reward_type ef18() {return(rate("t19"));}
/* throughput  of t19*/
reward_type ef19() {return(rate("t20"));}
/* throughput  of t20*/

reward_type ef20() {return(mark("p2") + mark("p3"));}
/* utilization of p2 and p3 is the buffer occupancies*/
reward_type ef21() {return(mark("p8") + mark("p11"));}
/* utilization of p8 and p11 is the buffer occupancies*/
reward_type ef22() {return(mark("p9") + mark("p10"));}
/* utilization of p9 and p10 is the buffer occupancies*/
reward_type ef23() {return(mark("p18") + mark("p20"));}
/* utilization of p18 and p20 is the buffer occupancies*/
reward_type ef24() {return(mark("p6"));}
/* utilization of p6*/
reward_type ef25() {return(mark("p7"));}
/* utilization of p7*/
/* ____________________________________________________________ */
reward_type ett19() {return(rate("t19")*rate("t19"));}
/* second_moment  of t19*/
reward_type ett20() {return(rate("t20")*rate("t20"));}
/* second_moment  of t20*/
```

```
/*Output results*/

ac_final() {

 pr_expected("throughput(t3) ",ef2);
 pr_expected("throughput(t4) ",ef3);
 pr_expected("throughput(t13) ",ef12);
 pr_expected("throughput(t14) ",ef13);
 pr_expected("throughput(t15) ",ef14);
 pr_expected("throughput(t16) ",ef15);
 pr_expected("throughput(t19) ",ef18);
 pr_expected("throughput(t20) ",ef19);

  pr_expected("utilization(p2+p3)  is the buffer occupancies
",ef20);
  pr_expected("utilization(p8+p11) is the buffer occupancies
",ef21);
  pr_expected("utilization(p9+p10) is the buffer occupancies
",ef22);
  pr_expected("utilization(p18+p20)is the buffer occupancies
",ef23);
     pr_expected("utilization(p6) ",ef24);
     pr_expected("utilization(p7) ",ef25);

 pr_expected("second_moment of (t19) ",ett19);
 pr_expected("second_moment of (t20) ",ett20);

             pr_std_average( );
            /* pr_std_average_der();*/
}
```

REFERENCES

1. Ciardo, G. 1990. Manual for SPN Package. Duke University.

2. Wastor III, J. F., and A. A. Desrochers. 1991. "Applying GSPN to Manufacturing Systems Containing Nonexponential Transition Functions." IEEE Transactions on System, Man and Cybernetics. Vol. 21. No. 5.

3. Balbo, G., S. C. Bruell, and S. Ghanta. 1986. "Combining Queueing Network and GSPN Models for the Analysis of Some Software Blocking Phenomena." IEEE Transactions on Software Engineering. Vol. 12. No. 4.

4. Guo, D. L., F. DiCesare, and M. C. Zhou. 1991. "Moment Generating Function Approach to Performance Analysis of Extended Stochastic Petri Nets." Proceedings of the IEEE International Conference on Robotic and Automation: 1309-1314.

5. Jungnitz, H., and A. Desrochers. 1991. "Flow Equivalent Nets for the Performance Analysis of Generalized Stochastic Petri Nets." Proceedings of the IEEE International Conference on Robotic and Automation.

6. Zhou, M. C. et al. 1991."Construction of Petri Net Based Mathematical Models of an FMS Cell." Proceedings of the IEEE International Conference on System, Man and Cybernetics.

7. Murata, T. 1989. "Petri Nets: Properties, Analysis and Applications." Proceedings of the IEEE. Vol. 77. No.4.

8. Peterson, J. L. 1981. Petri Net Theory and the Modeling of Systems. Englewood Cliffs. Prentice-Hall.

9. Marson, M. A. et al. 1989. "The Effect of Execution Polices on the Semantics and Analysis of an Stochastic Petri Nets." IEEE Transactions on Software Engineering. Vol. 15. No.7.

10. Valatte, R. 1979. "Analysis of Petri Nets by Stepwise Refinements." Journal of Computer and System Science. 18-35.

11. Chen, P. Z. et al. 1989. "Alternative Methods for Incorporating Non-exponential Distribution into Stochastic Timed Petri Nets." In Proc. Int. Workshop Petri Nets and Performance Models. Kyoto. Japan. IEEE: 187-197.

12. Zhou, M. C., D. Guo, and F. DiCesare. 1993. "Integration of Petri Nets and Moment Generating Function Approach for System Performance Evaluation." To Appear in Journal of Systems Integration.

13. Marson, M. A., and G. Chiola. 1987. "On Petri Net with Deterministic and Exponential Transition Firing Times." in LNCS. Vol. 266[24]: 132-145.

14. Dugan, J.B., K. S. Trivedi, R. M. Geist, and V. F. Nicola. 1984."Extended Stochastic Petri Nets: Applications and Analysis." PERFORMANCE' 84. in Proc. 10th Int. Symp. Paries: 507-519.

15. Chiola, G. 1987. "A Graphical Petri Net Tool for Performance Analysis." Proceedings of the 3rd International Workshop on Modeling Techniques and Performance Evaluation. AFCET.

16. Dugan, J. B., A. Bobbio, G. Ciardo, and K. Triverdi. 1985. "The Design of a Unified Package for the Solution of Stochastic Petri Net Models." Proceedings of the IEEE International Workshop on Timed Petri Nets: 6-13.

17. Holiday, M. A., and M. K. Vernon. 1985. "A Generalized Timed Petri Net Model for Performance Analysis." Proceedings of the IEEE International Workshop on Timed Petri Nets: 181-190.

18. Molloy, M. K. 1982. "Performance Analysis Using Stochastic Petri Nets." IEEE Transactions on Computer. Vol. c-31. No.9.

19. Viswanadham, N., and Y. Narahari. 1988."Stochastic Petri Net Models for Performance Evaluation of Automated Manufacturing Systems." Information and Decision Technologies 14: 125-142.

20. Lee-Hwang, H., J. Favrel, and P. Baptiste. 1987. "Generalized Petri Net Reduction Method." IEEE Transactions on System, Man and Cybernetics, Vol. SMC-17. No.2.

21. Lee-Hwang, H., and J. Favrel. 1985. "Hierarchical Reduction Method for Analysis and Decomposition of Petri Net." IEEE Transactions on System, Man and Cybernetics. Vol. SMC-15: 272-280.

22. Marson, M. A., G. Balbo, and G. Conte. 1984. "A Class of GSPN for the Performance Evaluation of Multiprocessor Systems." ACM Transactions on Computer Systems. Vol. 2. No. 2: 93-1.

23. Ma, J. M., and M. C. Zhou. 1992. "Generalizing the Moment Generating Function Approach to Performance Analysis to Deterministic and Stochastic Petri Nets (DSPN)." Society for the Advancement of Material and Process Engineering. 3rd MINI-TECH Conference. NJ.

24. Ma, J. M., and M. C. Zhou. 1992. "An Approximation Algorithm for Performance Analysis of Arbitrary Stochastic Petri Nets," Society for the Advancement of Material and Process Engineering. 3rd MINI-TECH Conference. NJ.

25. Ma, J. M., and M.C. Zhou. 1992."Performance Evaluation of Discrete Event Systems Via Stepwise Reduction and Approximation of Stochastic Petri Nets." To Appear in the IEEE International Conference on Decision and Control. Tuscon. AZ: December 16-18.

26. Ma, J. M., and N. Levy. 1992. "Modeling and Performance Analysis of Flexible Manufacturing System By Using Petri Nets." Society for the Advancement of Material and Process Engineering. 3rd MINI-TECH Conference. NJ.

27. Ma, J. M., and M. C. Zhou. 1993 "Performance Evaluation By Throughput Algebra on Timed Petri Nets," Working Paper. Manufacturing Engineering Programs. New Jersey Institute of Technology.

28. Zhou, M. C., and J. M. Ma. 1993."Reduction and Approximation for Performance Evaluation of Stochastic Petri Nets with Multi-input and Multi-output Modules." Submitted to 12th IFAC World Congress (DES Control). Sydney. Australia.

29. Ma, J. M., M. C. Zhou, Daniel Y. Chao, and N. Levy. 1993."Applying Petri Net Approximation to Performance Evaluation of Flexible Manufacturing System with Nonexponential Firing Functions " Submitted to 14th International Conference on Application and Theory of Petri Nets,USA.

30. Ma, J. M. 1992."Reduction and Approximation Method Studies for Performance Evaluation of Timed Petri Nets." Working Paper. New Jersey Institute of Technology.