

STATE UNIVERSITY OF NEW YORK AT
STONY BROOK

CEAS TECHNICAL REPORT 598

WHY MOST STOCHASTIC PERTI NETS
ARE NON-PRODUCT FORM NETWORKS

T.G. Robertazzi

March 8, 1991

Why Most Stochastic Petri Nets Are Non-Product Form Networks

Prof. Thomas G. Robertazzi

Dept. of Electrical Engineering,
SUNY at Stony Brook,
Stony Brook N.Y. 11794

Abstract

It is shown that stochastic Petri net models of the fundamental paradigms of concurrent resource sharing and synchronization among concurrent processes generally do not have a product form solution for the equilibrium state probabilities. This occurs since the state transition diagrams of such models are not decomposable into the usual consistent set of integral building blocks associated with product form networks. Thus it appears that, except for special cases, there is generally no closed form solution for stochastic Petri nets. Such nets must be solved by such techniques as Markov chain solvers or simulation.

I. Introduction

Petri networks were introduced in 1962 by C.A. Petri [6]. In early work on the subject the timing in Petri nets was assumed to be deterministic. That is, a transition would fire a constant amount of time after being enabled. Stochastic timing for Petri nets was introduced independently by Florin and Natkin [1] and Molloy [4]. To date the usual assumption for such stochastic Petri nets is that a transition fires an exponentially distributed amount of time after being enabled. Thus a stochastic Petri net is a Markovian system and can be associated with a Markov chain. A further elaboration, Generalized Stochastic Petri Nets [3], has the additional feature of immediate transitions which fire immediately after being enabled.

Petri nets are useful for modeling concurrency, serializability, synchronization and resource sharing [5]. It is the purpose of this paper to show that basic Petri net models of concurrent resource sharing and synchronization among concurrent processes do not admit a (closed form) product form solution, except in special cases.

Previous work by the author and co-authors [2,7,8] has characterized the conditions necessary for a product form solution for *safe* stochastic Petri nets consisting of a concurrent number of linear sequences of places and transitions known as "task sequences". The characterization holds that a product form solution exists if, whenever a token is allowed to proceed in a task sequence, there is a non-zero probability that the task sequence can return to its original state without a need

for state changes in other task sequences to allow this progress.

This paper is organized as follows. In section II, the case of non-safe stochastic Petri nets modeling resource sharing is examined. In section III the case of non-safe Petri nets modeling synchronization is considered. The interested reader can consult [5,7] for the details of stochastic Petri net operation. In this paper all transitions are assumed to fire a negative exponential distributed amount of time after being enabled.

II. Non-Safe Resource Sharing Models

First consider the canonical non-safe resource sharing model of Fig. 1. It consists of a linear sequence of three places indicating the basic functions IDLE, REQUESTING RESOURCE and ACCESSING RESOURCE. There is also a RESOURCE place. After being enabled transitions fire at rates of, respectively, q^0 , q^1 , and q^2 . Such models, developed for multiprocessor modeling, appear in [10].

Fig. 2 shows the state transition diagram of this model where the number of users making making resource accesses is the vertical coordinate and the number of resource requests is the horizontal coordinate. Transitions have the rates indicated in the legend. The state transition diagram has an upper boundary due to the constraint that the number of concurrent resource accesses must be less than the number of resources. There is also a diagonal boundary at the right due to the constraint that the sum of the number of concurrent resource requests and the number of concurrent resource accesses must be less than the number of users.

In [7] it is pointed out that the probability flux of networks with a product form solution has a distinctive circulatory structure. Specifically, the aggregate circulation is composed of an aggregation of smaller, closed circulations ("isolated circulations"). The state transition diagram of Fig. 2, almost, but not completely, has the requisite structure to produce a product form solution. The flaw is the presence of transitions such as those labeled "A" and "B". These disrupt the flow of probability flux so that isolated circulations and a product form solution do not result.

In fact, if these transitions were removed from the state transition diagram a product form solution would exist. The corresponding Petri net schematic is shown in Fig. 3. It is like the one of Fig. 1 except for the presence of the dotted arc. The presence of this arc means that a necessary condition for the q^0 transition to fire is that there must be at least one resource in the resource place. Note that the dotted arc also indicates that resources are not removed from the resource place when the q^0 transition fires.

Essentially this means that a user can only move from being idle to requesting a resource if a resource is presently free. In the model of Fig. 1 a user can go from being idle to requesting a resource and then be "stuck" there if no resources are available. This can be thought of as a form of blocking. It is well known that blocking queueing networks generally do not have a product form solution.

Another way of looking at the model of Fig. 3 is to note that requests are immediately cleared from the system if the necessary resources are not available. By way of contrast, in the model of Fig. 1 users requesting resources are allowed to wait in the USERS REQUESTING RESOURCES place until they become available.

The product form solution for the model of Fig. 3 is:

$$p(n_r, n_a) = \left(\frac{q^0}{q^1} \right)^{n_r} \left(\frac{q^0}{q^2} \right)^{n_a} p(0,0) \quad (2.1)$$

where n_r is the number of users requesting resources and n_a is the number of users accessing resources.

An alternate way of modeling resource sharing appears in Fig. 4. Here each user has a separate Petri subnet to represent the states it can assume. These subnets are safe. There is, again, a single place for holding multiple idle resources.

The state transition diagram for this model when there are two users and a single resource is shown in Fig. 5. Here the coordinates for a user indicates if it is idle ("0"), requesting a resource ("1") or accessing a resource ("2"). Work on safe Petri nets [2,8] with product form solutions indicates that this model would have a product form solution if it wasn't for the presence of transitions "A" and "B". Once again, the problem is that a user can move from being idle to requesting a resource and then be "stuck" there if the resource is being accessed by the other user. This form of blocking precludes the existence of the product form solution.

Increasing the dimensionality, Fig. 6 shows the state transition diagram for three users and two resources. Again, the presence of transitions "A", "B" and "C" precludes the existence of the product form solution. As the dimensionality is increased further, the same problem of blocking remains as long as there are less resources than users.

To show that a state transition diagram like those of Fig. 5 and Fig. 6, without the A,B,C... transitions has a product form solution one can consider the drawing of Fig. 7. It shows one of the integral building blocks [7] that should have an isolated circulation if the product form solution exists. Specifically it is a building block of the k th task sequence (i.e. of user k), with the other user states assumed to be held fixed. We can assure a product form solution of the form:

$$p(0) = \frac{q^{10}}{q^{1i}} \frac{q^{20}}{q^{2j}} \dots \frac{q^{N0}}{q^{Nz}} \quad (2.2)$$

$$p(1) = \frac{q^{k0}}{q^{k1}} p(0) \quad (2.3)$$

$$p(2) = \frac{q^{k0}}{q^{k2}} p(0) \quad (2.4)$$

and show that it satisfies the local balance equations of this building block. Here (i,j,k...z) indicates the states of the task sequences. Similar equations satisfy each of the possible building blocks of the state transition diagram and so the product form solution exists.

The overall solution is

$$P(k_1, k_2, \dots, k_N) = \prod_{l=1}^N \frac{q^{l0}}{l k_l} p(0, 0, \dots, 0) \quad (2.5)$$

where k_l is the k_l th sub-task (place) of the l th linear task sequence of places [2,7,8].

In summary, we have delineated two classes of resource sharing models. For one, users may proceed from being idle to requesting a resource and then be blocked from proceeding if resources are not available. This class is a non-product form network. For the other class, requests for resources that are not immediately available are immediately cleared from the system and a product form solution exists. It seems, to this researcher, that the former class would be more prevalent in applications. It should be noted that the preceding information is of interest for situations where the canonical models of this paper are sub-nets of larger, more complex nets.

III. Synchronization Models

Besides resource sharing, another common paradigm that can be modeled by Petri nets is that of synchronization. In this paper the view is taken that synchronization can be modeled as the need for tokens to arrive into each of a number of places that are incident to the same transition before the transition can fire. This may represent, for instance, parallel fragments of a process that must be completed before the next, serial fragment can be executed. In this section a number of increasingly elaborate synchronization models will be examined.

The first synchronization model appears in Fig. 8. It, and all the models of this section, are not safe. There must be at least one token in each of the upper places before the λ transition can fire. For this simple minded model the synchronization problem is trivial since the μ transition releases tokens into each of the upper places simultaneously. Letting the number of tokens in the lower place be the state description, the state description of Fig. 9 results. It can be seen to be equivalent to the state transition diagram of a single queue with finite buffer size and has the characteristic one dimensional product form solution:

$$p(n) = \left(\frac{\lambda}{\mu} \right)^n p(0) \quad n=1,2\dots M \quad (3.1)$$

A model with a more substantial stochastic component appears in Fig. 10. Here there must be tokens in each of the l_i places before the λ_1 transition can fire, in each of the m_i places before the λ_2 transition can fire and in each of the n_i places before the μ transition can fire. The stochastic synchronization part of this model has to do with the n_1 and n_2 places and μ transition. Tokens in one of these places may have to wait if the other place is empty. The waiting time is distributed exponentially with the rate parameter of the appropriate λ transition.

The state transition diagram of the model of Fig. 10 appears in Fig. 11. The state variables are the number of tokens in the n_1 and n_2 places. The transition rates are shown in the legend. Expressions for the extent of the right and upper boundaries are also listed. These boundaries are not necessarily equal in size.

If one makes the change of variables

$$n'_{1} = n_{1} \quad (3.2)$$

$$n'_{2} = M_{2} - n_{2} \quad (3.3)$$

the state transition diagram can be seen to be equivalent to that for two tandem queues with finite buffers. It is well known that such a model has no analytical solution (if one buffer is of size 1 or 2, recursive solutions are possible [9]).

The next model appears in Fig. 12. The state description is now three dimensional (n_1, n_2, n_3) and the state transition diagram appears in Fig. 13. The transitions oriented in rectilinear directions correspond to arrivals to the n_1, n_2, n_3 places and the diagonal transitions correspond to the firing of the μ transition. The overall structure is rectangular in 3 dimensions.

The state transition diagram is composed of elements like that of Fig. 14 with adjacent elements sharing element transitions. These elements are not "building blocks" in the product form sense [7,8].

To see if the state transition diagram of Fig. 13 can support a product form solution one must be able to find a decomposition of the state transition diagram into integral building blocks with isolated circulations. This does not seem possible. For instance, one might consider a decomposition into tetrahedral shaped building blocks. There are six such potential decompositions. However after attempting to make each such transition there are transitions left over that do not belong to integral building blocks. This indicates a lack of product form solution.

After examining the models of Fig. 8 and Fig. 10 one might conjecture that the state transition diagram of Fig. 13 is equivalent to that of three tandem queues with finite buffers. In fact this is not true. There are key topological differences between the state transition diagram of Fig. 13 and that of the finite buffer three tandem queue network:

- The state transition diagram origin of the three queue tandem network has one incoming transition and one outgoing transition. None of the corners of Fig. 13 (or Fig. 14) have this topology.
- In the tandem model state transition diagram a "diagonal" transition increments a single state variable (i.e. queue length) and decrements another. In the state transition diagram of Fig. 13 a diagonal transition decrements three state variables simultaneously.

Experience with such synchronization state transition diagrams for 1,2 and 3 dimensions would appear to indicate that one can conjecture that models of higher dimensionality do not have a product form solution.

IV. Conclusion

The consequence of these new results, as well as previous results [2,8], is to indicate that generally stochastic Petri nets modeling concurrent resource sharing and synchronization among concurrent processes do not have a product form solution for the equilibrium state probabilities. In the context of resource sharing the situation where a product form solution exists is when requests for resources are cleared when the necessary resources are not immediately available. It would seem that the cases where a product form solution exists would be outnumbered in practice by those where it does not exist. This observation confirms the use of Markov chain solvers and simulation for analyzing such nets.

Acknowledgement

The research in this paper was supported by the National Science Foundation under grant no. NCR-8703689 and by the SDIO/IST and managed by the Office of Naval Research under grant no. N00014-85-K0610.

References

- [1] G. Florin and S. Natkin, "Evaluation des Performances d'un Protocole de Communication a l'aide des Reseaux de Petri et des Processus Stochastiques", *J. AFCET Multiprocesseurs et Multiordinateurs en Temp Re'el*, 1978, pp. E0-E26.
- [2] A.A. Lazar and T.G. Robertazzi, "Markovian Petri Net Protocols with Product Form Solution", *Performance Evaluation, Vol. 12, 1991*, pp. 67-77.
- [3] M. Marsan, G. Balbo and G. Conte, "A Class of Generalized Stochastic Petri Nets for the Performance Evaluation of Multiprocessor Systems", *ACM Transactions on Computer Systems*, May 1984.
- [4] M.K. Molloy, "Performance Analysis using Stochastic Petri Nets", *IEEE Transactions on Computers*, Vol. C-31, No. 9, 1982, pp. 913-917.
- [5] J.L. Peterson, *Petri Net Theory and the Modeling of Systems*, Prentice Hall, Englewood Cliffs N.J. 1981.
- [6] C.A. Petri, *Kommunikation mit Automaten*, Ph.D Thesis, University of Bonn, Federal Republic of Germany, 1962.
- [7] T.G. Robertazzi, *Computer Networks and Systems: Queueing Theory and Performance Evaluation*, Springer-Verlag, N.Y. 1990.
- [8] I.Y. Wang and T.G. Robertazzi, "Service Stage Petri Net Models with Product Form Solution", *Queueing Systems*, 1990, pp. 355-374.
- [9] I.Y. Wang and T.G. Robertazzi, "Recursive Computation of Steady-State Probabilities of Nonproduct Form Queueing Networks Associated with Computer Network Models", *IEEE Transactions on Communications*, Vol. 38,

No. 1, Jan. 1990, pp. 115-117.

- [10] Marsan, M., Balbo, G., Conte, G. and Gregoretti, F., "Modeling Bus Contention and Memory Interference in a Multiprocessor System", *IEEE Transactions on Computers*, Vol. C-32, No. 1, 1983, pp. 60-72.

Idle
Users

Users
Requesting
Resources

Users
Accessing
Resources

Resources

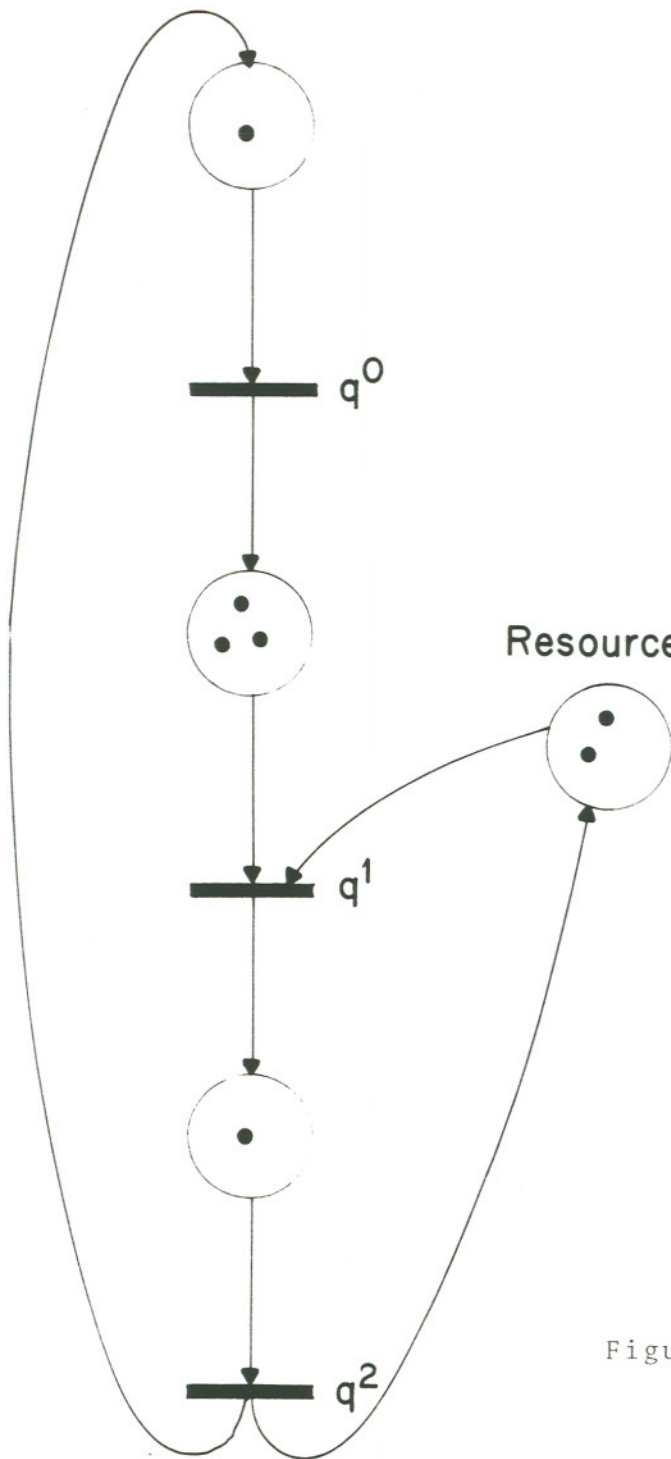
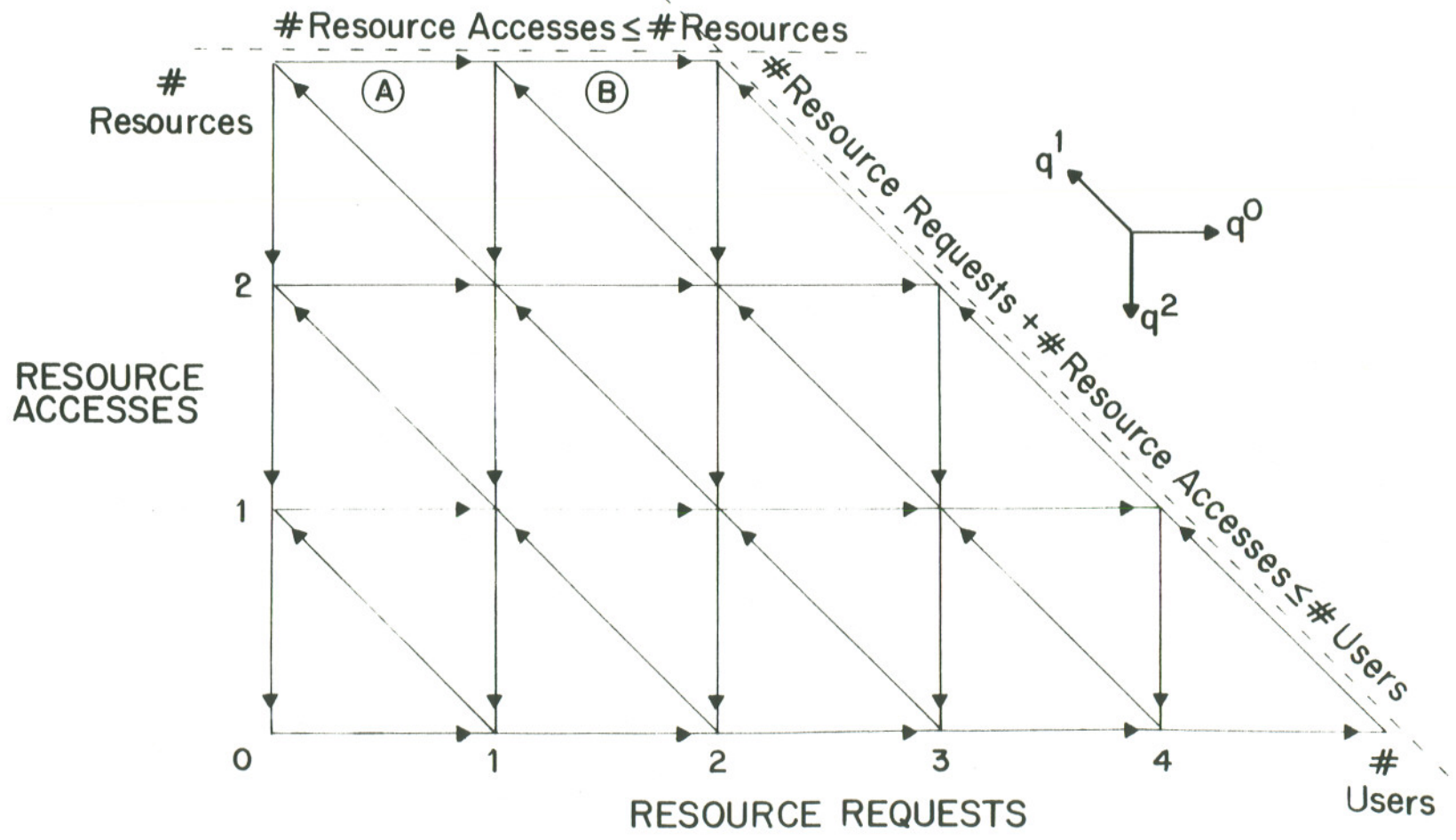


Figure 1

Figure 2



Idle
Users

Users
Requesting
Resources

Users
Accessing
Resources

Resources

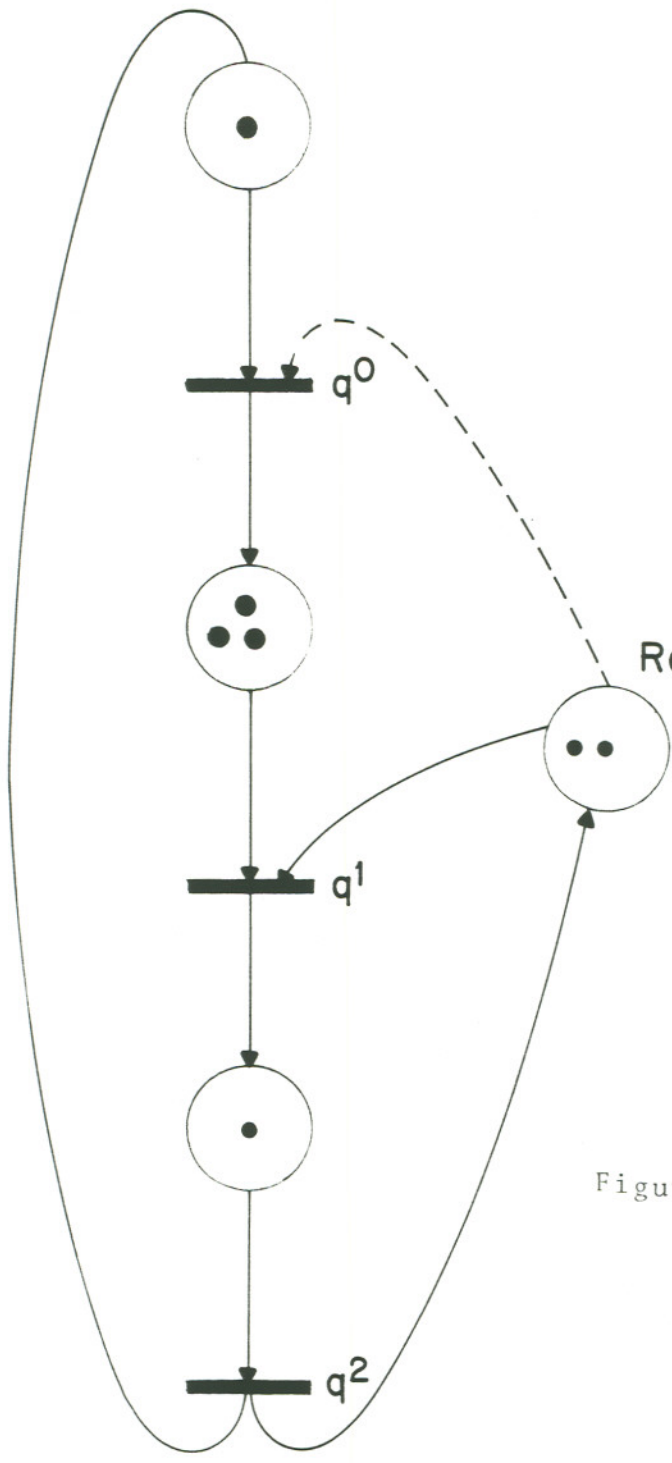


Figure 3

Figure 4

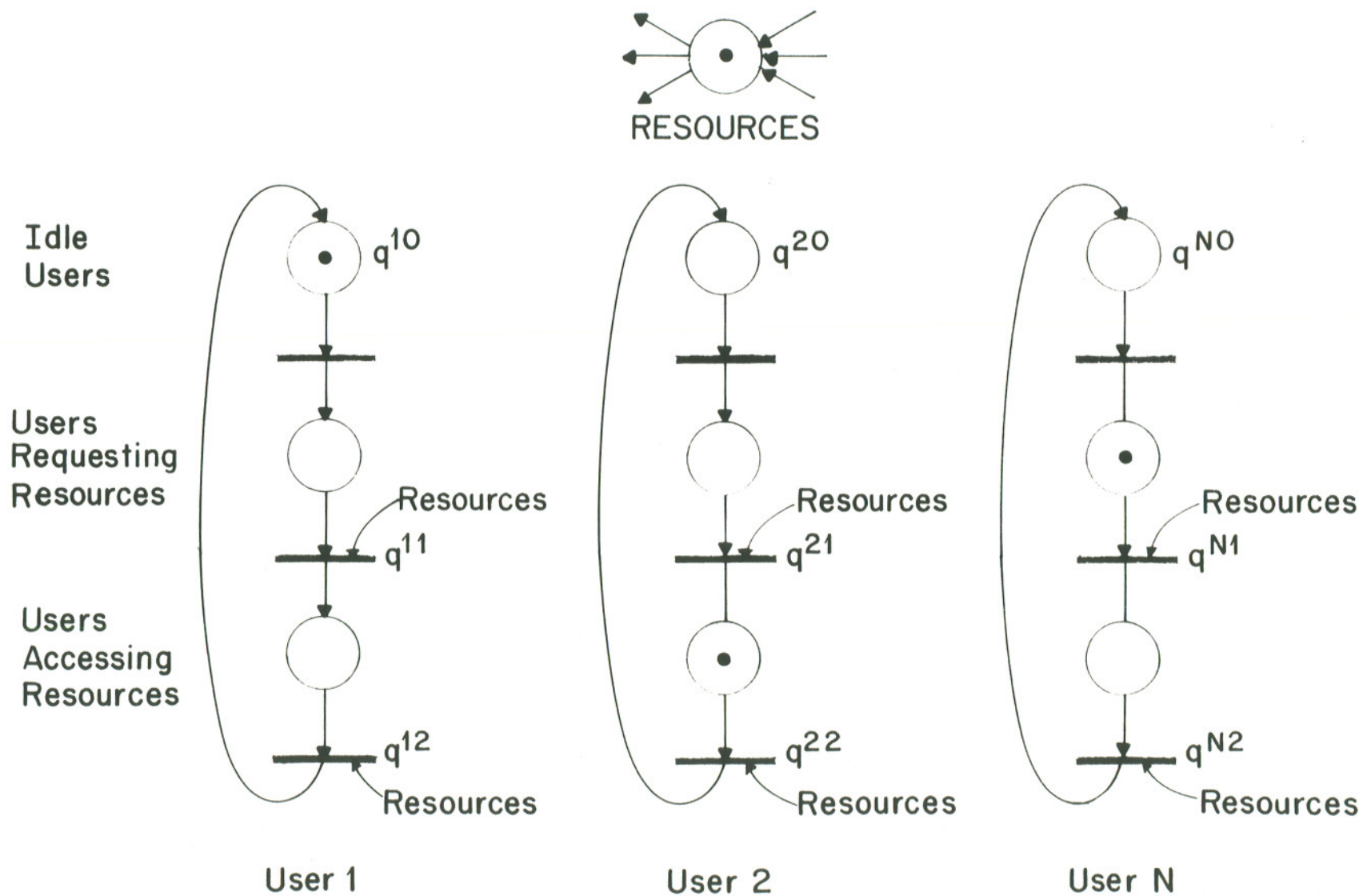
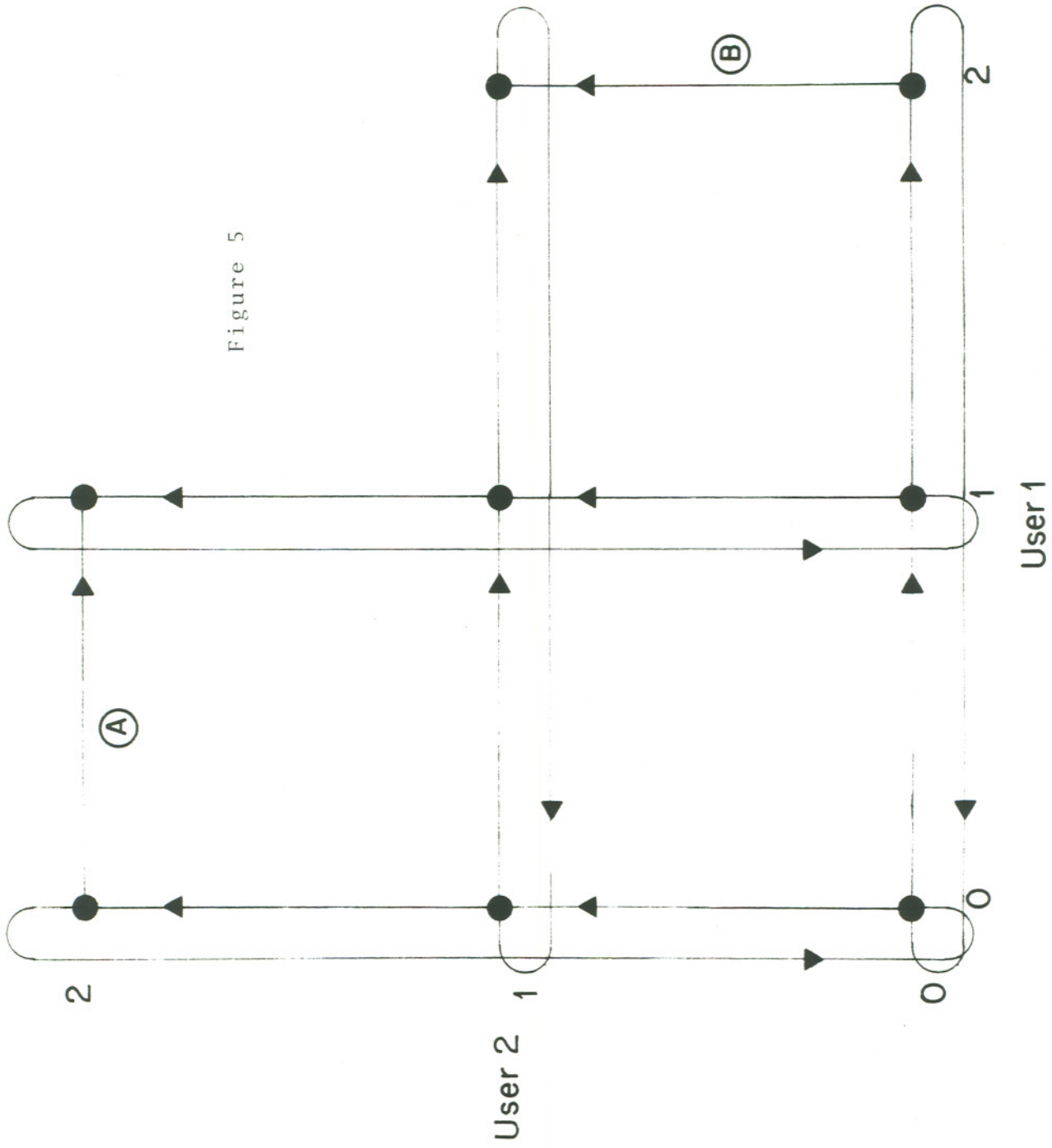


Figure 5



0=Idle User
1=User Requesting Resource
2=User Assessing Resource

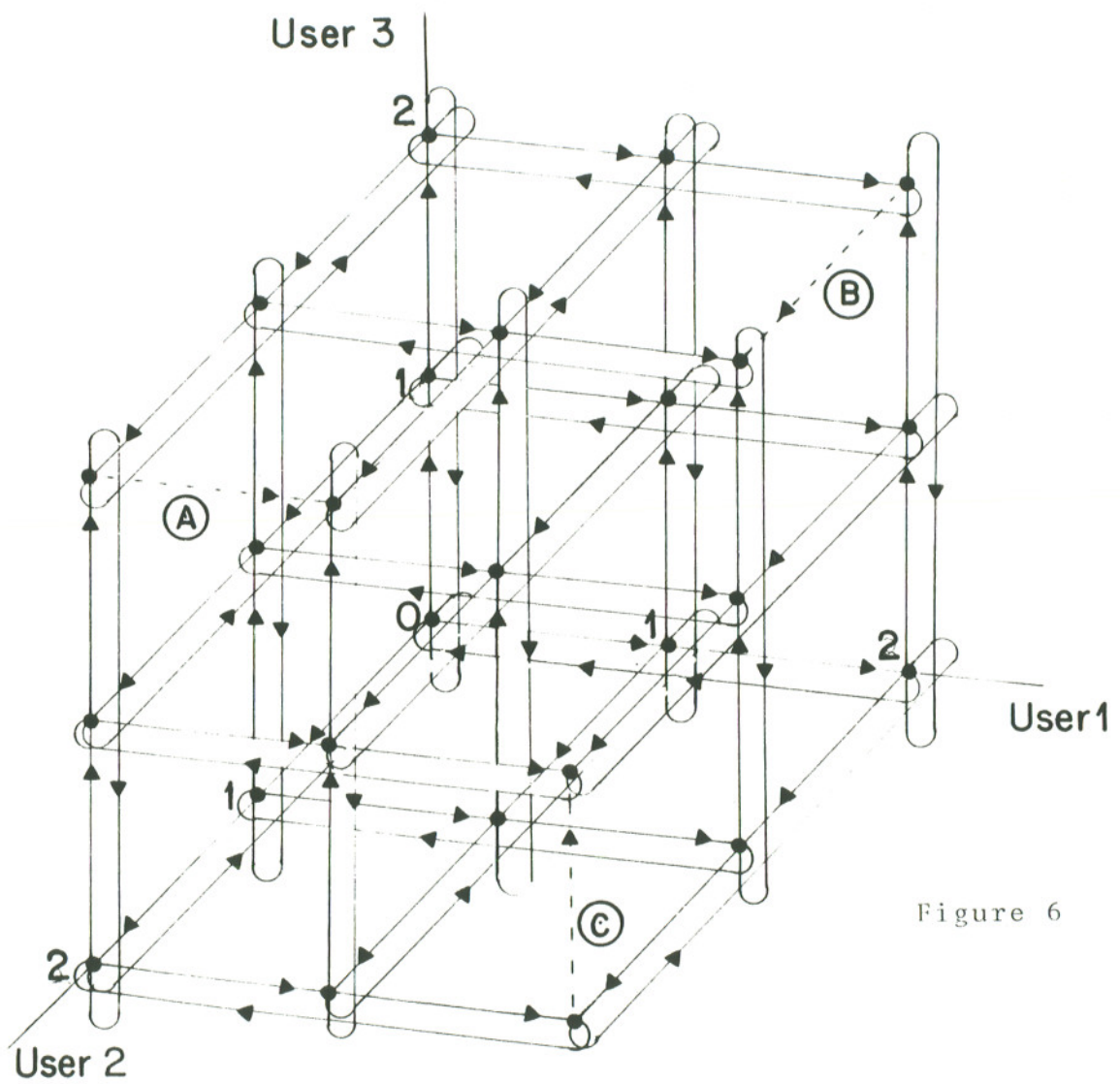


Figure 6

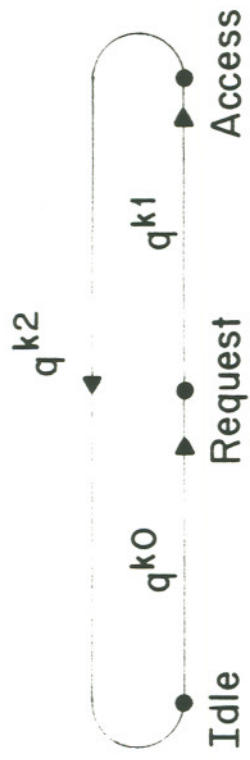


Figure 7

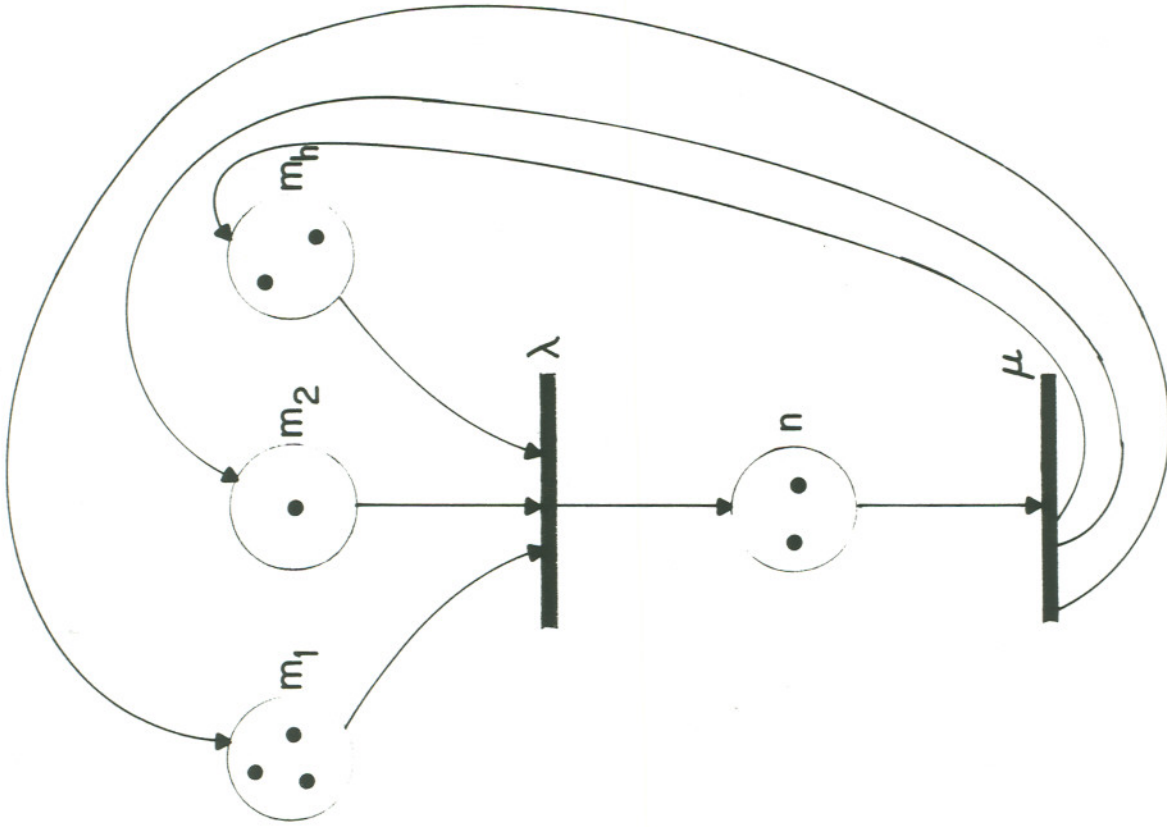


Figure 8

$$M = \min(\max m_1, \max m_2, \dots, \max m_n)$$

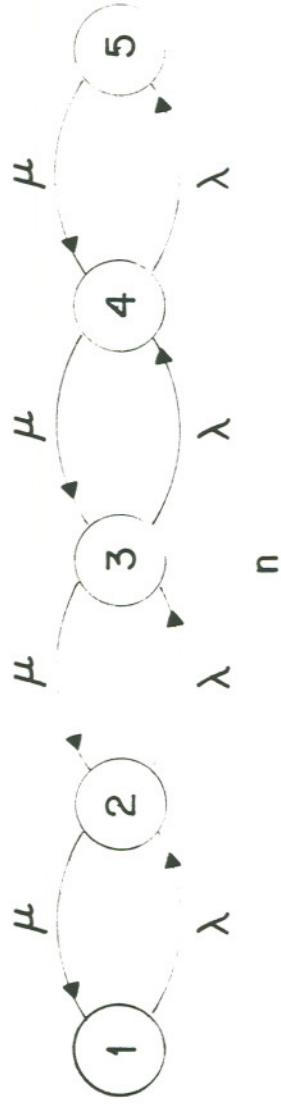


Figure 9

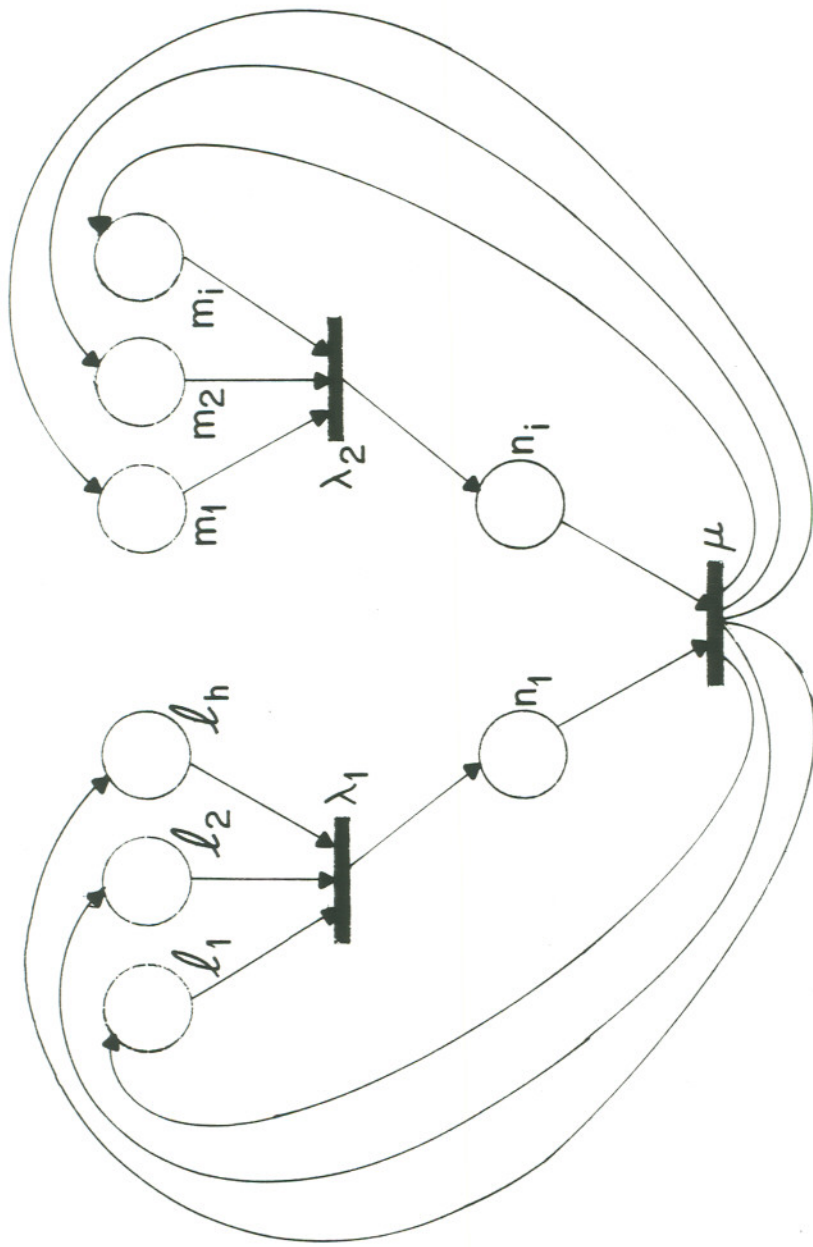


Figure 10

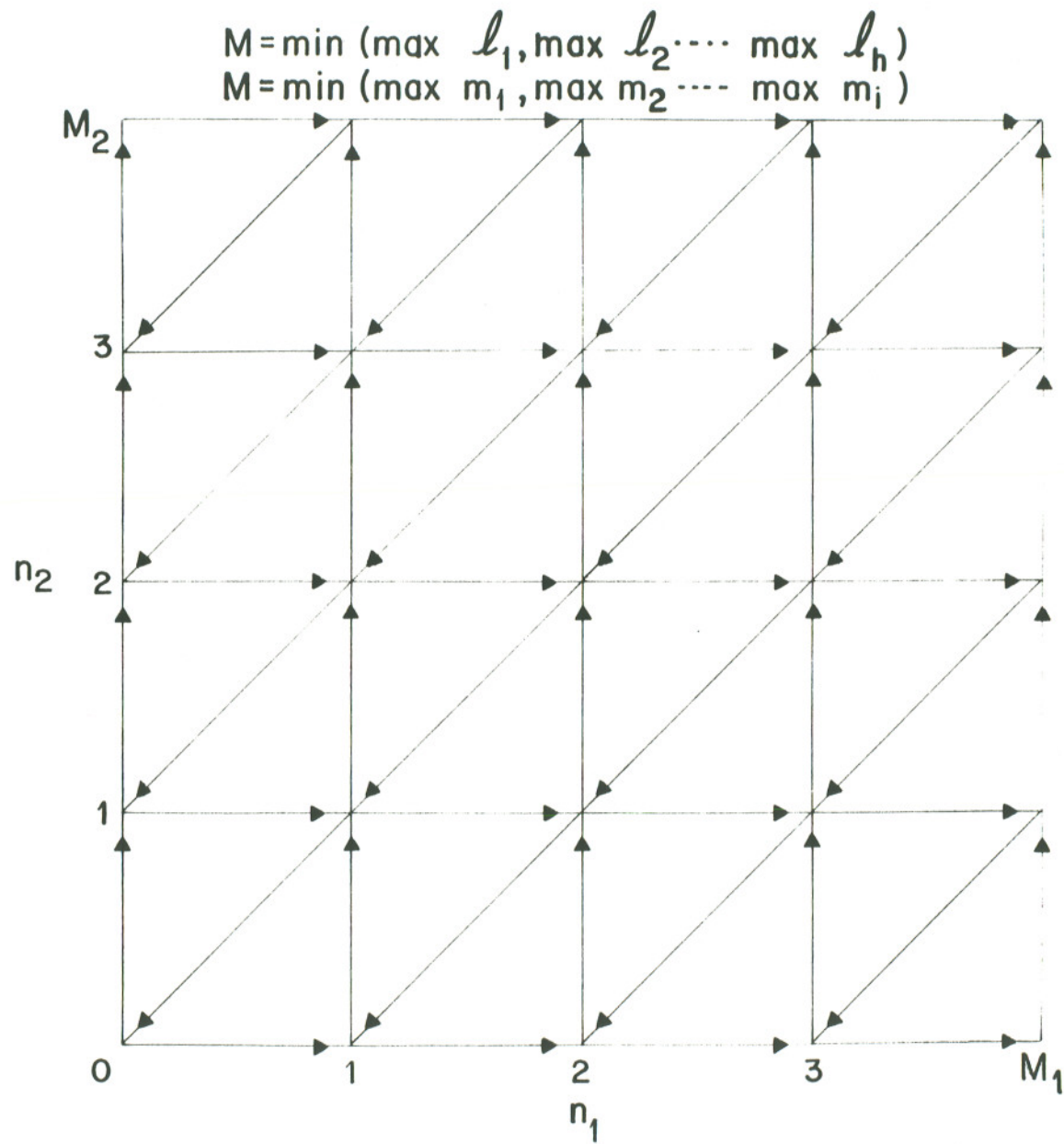
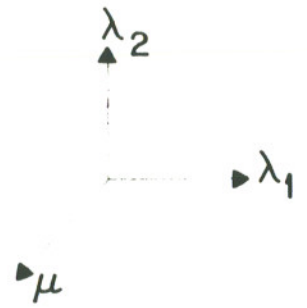


Figure 11



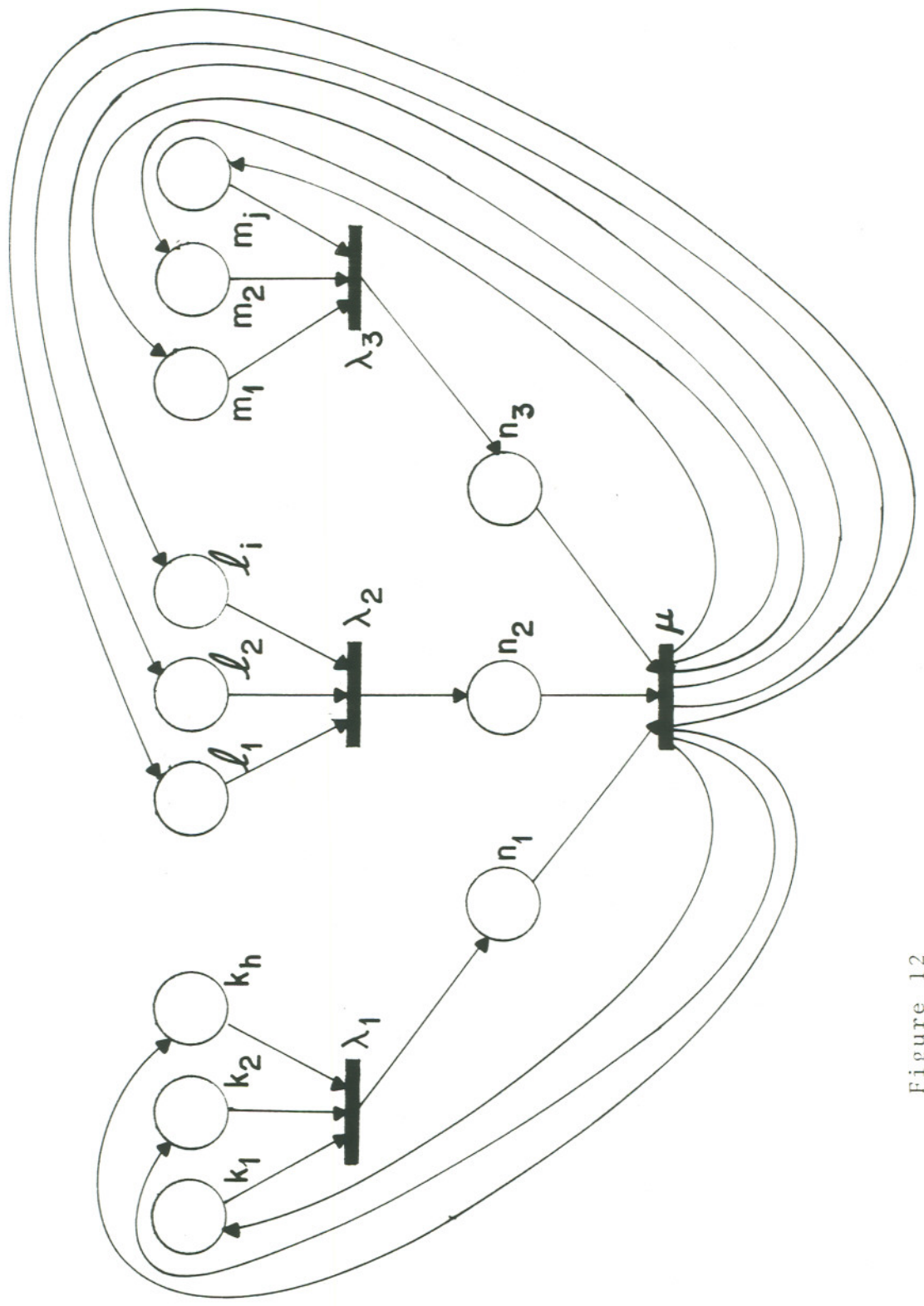


Figure 12

$$M_1 = \min (\max k_1 , \max k_2 \dots \max k_h)$$

$$M_2 = \min (\max l_1 , \max l_2 \dots \max l_i)$$

$$M_3 = \min (\max m_1 , \max m_2 \dots \max m_j)$$

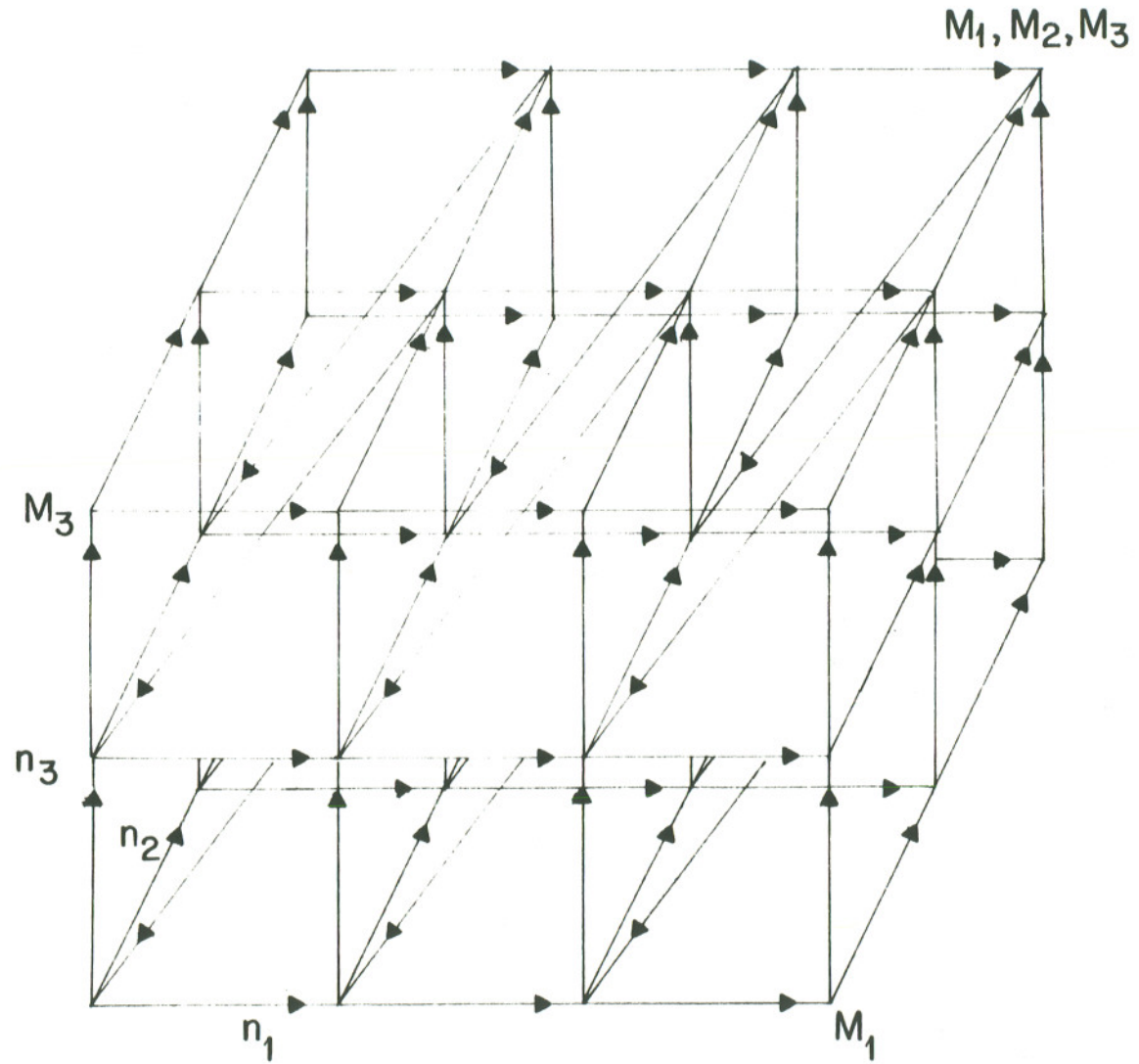


Figure 13

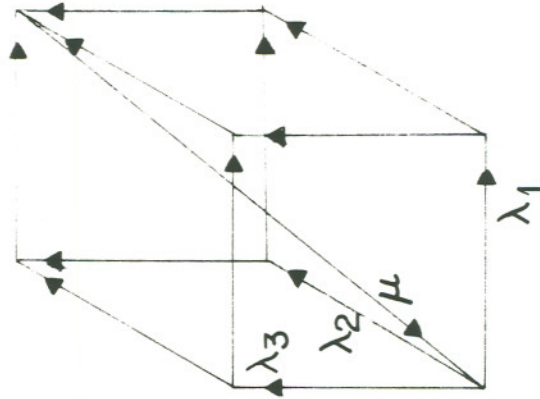
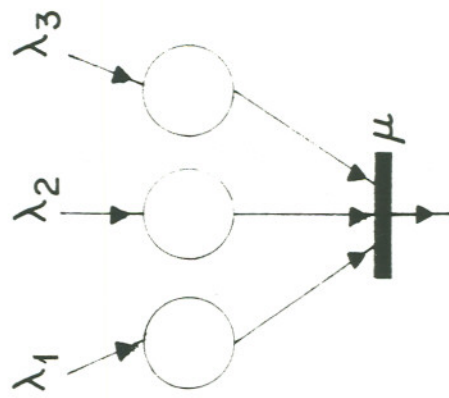


Figure 14