The 1985 International Conference on Parallel Processing. August, 1985. pp. 521-530

A SEMI-MARKOV MODEL FOR THE PERFORMANCE OF MULTIPLE-BUS SYSTEMS

by

T. N. Mudge and Humoud B. Al-Sadoun
Computing Research Lab
EECS Department
The University of Michigan
Ann Arbor, MI 48109.

Abstract

This paper presents a discrete time model of memory interference in multiprocessor systems employing multiple-bus interconnection networks. It differs from earlier models in its ability to model variable connection time and arbitrary interrequest time. The model describes each processing element's behavior by means of a semi-Markov process. It takes as input the number of processing elements, the number of memory modules, the number of buses, the mean think time of the processing elements, and the first and second moments of the connection time between processing elements and memories. The model produces as output the memory bandwidth, processing element utilization, memory module utilization, average queue length at a memory and average waiting time experienced by a processing element while waiting to access a memory. Using the model, it is possible to analyze the interaction of the input parameters on the system performance. This modeling capability is attained without having to employ a complex Markov chain. In fact, a four state semi-Markov process is sufficient regardless of the think and connection time distributions. The accuracy and capability of the model is illustrated.

Index terms--Multiple-bus system, multiprocessors, memory interference, memory bandwidth, performance evaluation, semi-Markov processes, Markov chains.

This work was supported in part by National Science Foundation under Grant MCS-8009315.

1. Introduction

There have been an interesting variety of proposals for interconnecting processors and memories in multiprocessors [13]. This paper presents a discrete time model for one of these, the multiple-bus interconnection network. The model allows the user to quantify the effects of changing various system design parameters on such performance measures as memory bandwidth, processor utilization, memory queue length and waiting time. A number of discrete time models for multiple-bus systems have been presented in [7, 16, 3, 10, 12, 5]. The model in this paper is the first to include variable connection time and arbitrary interrequest time. The introduction of semi-Markov processes to model the processor behavior allows this increased generality without model complexity. A four state semi-Markov process is sufficient for the model. The use of semi-Markov processes also simplifies the derivation of the memory queue length and waiting time.

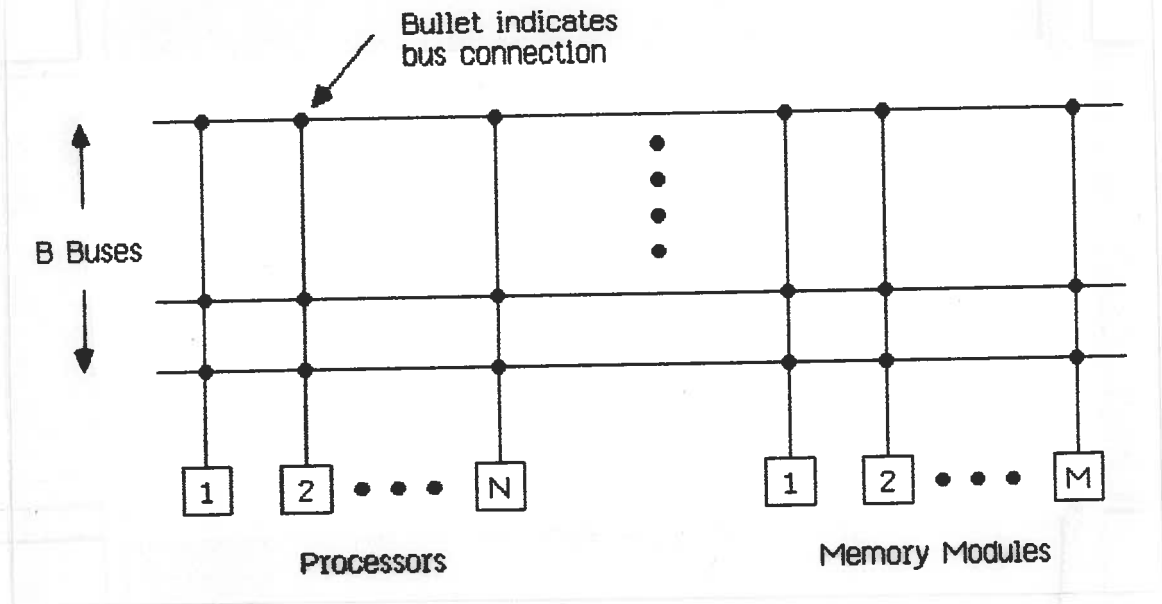

Figure 1. A multiple-bus system.

Figure 1 shows a typical multiprocessor system in which B buses are used to interconnect N processing elements with M memory modules (where $B \leq \min(N,M)$). The multiprocessor of Fig. 1 will be referred to as an $N \times M \times B$ system. The multiple-bus network has a number of desirable features. First, it is compatible with the prevailing "bus-centered" philosophy implicit in most microprocessor families of components. Second, the multiple-bus system is modular allowing easy incremental increase in the number of the processing elements, the number of memory modules and the number of buses. Finally, the multiple-bus system is fault-tolerant. For instance, if a bus fails the system still works (provided $B > 0$), albeit with degraded performance.

The model presented in this paper is an extension of the technique developed in [11] where semi-Markov processes were first used to model memory interference. We adopt the terminology of [11] and refer to our model as a semi-Markov interference (SMI) model. In an $N \times M \times B$ system three types of memory interference, or memory conflict, can occur. A type one conflict arises when several processing elements attempt to access an idle memory module simultaneously. A type two conflict arises when a processing element attempts to access a busy memory module. A type three conflict arises when one or more processing elements attempt to access an idle memory module when no buses are available. We follow the two-stage arbitration scheme proposed by Lang et al. [7] to resolve access conflicts. In the first stage, the conflict due to the memory modules is resolved by M arbiters of the N-users 1-server type. Each of these arbiters selects equibrobably one of the processing elements which have outstanding requests to the arbiter's associated memory module. In the second stage, the conflict due to the buses is resolved by one arbiter of the M-users B-servers type. This arbiter assigns the memory requests selected in the first stage to the available buses. The arbiter makes the assignment in a cyclic fashion, i.e., on a round robin basis.

The paper is organized as follows: Sec. 2 describes the assumptions that characterize the operation of the multiprocessor system; Sec. 3 develops the SMI model under the assumptions of Sec. 2; Sec. 4 concludes the paper by evaluating the SMI model against simulations.

2. The System Operation Assumptions

The multiprocessor system of Fig. 1 is assumed to be synchronous with a basic time unit of a bus cycle. A processing element (PE) may be in any of three states: *thinking*, when it is working on an internal task with no memory request outstanding; *accessing*, when it is connected to a memory module; and *waiting* or *blocked*, when it is waiting in the queue of a memory module for that memory to become available. The memory module (MM) can be in any of two states: *busy*, when a processing element is connected to it; and *idle*, when there is no processing element connected to it. The following notation will be used throughout this paper: a discrete random variable will be denoted by its name with a ~ above it, e.g., the discrete random variable Z will be denoted by $\tilde{Z}$; the probability mass function (pmf) of $\tilde{Z}$, will be denoted by $z(x)$, i.e., $z(x) = Pr[\tilde{Z} = x]$; the mean value of $\tilde{Z}$ will be denoted by $\overline{Z}$; and the n^{th} moment of $\tilde{Z}$ will be denoted by $\overline{Z^n}$.

System operation will be characterized by the following assumptions:

I. The behavior of the PE's can be modeled as identical stochastic processes.

II. The PE's think for an integer number of bus cycles. The thinking period of any PE is characterized by a discrete independent random variable, $\tilde{T}$.

III. Each PE will submit a memory request after its thinking period, i.e., the thinking time is the interrequest time. Requests originating from the same PE are independent of each other. The destination of the request originated from any PE will be uniformly distributed between the M memory modules.

IV. The system uses a two-stage arbitration scheme following that described in [7]. In the first stage the conflict due to the MM's (first conflict type) will be resolved by M arbiters of the N-users 1-server type. In the second stage the conflict due to the buses (third conflict type) will be resolved by 1 arbiter of the M-users B-servers type. The blocked PE's will try again to the same module in the next cycle.

V. When the second type of memory conflict occurs, i.e., the MM is busy when requested by a PE, the blocked PE waits until the connection is completed and then it resubmits its request to the same module.

VI. The connection time between a PE and any MM is characterized by a discrete independent random variable, $\tilde{C}$, measured in units of bus cycles.

Empirical evidence reported in [1, 2, 6], supports the assumptions in the case where $\tilde{C}$ is a deterministic random variable with a value of one. Further work reported in [8] supports the assumptions in the more general case where $\tilde{C}$ is a discrete random variable with arbitrary distribution.

In order to obtain numerical information from the SMI model developed later, the values of M, N, $\overline{T}$, $\overline{C}$ and $\overline{C^2}$, must be obtained through measurements or by hypothesis. These quantities can be regarded as input parameters of the SMI model; knowledge of the full distributions of $\tilde{T}$ and $\tilde{C}$ is not necessary for solving the SMI model. A number of performance measures can be derived from the analytical model. These are: memory bandwidth, BW; processing element utilization, PU; memory module utilization, MU; utilization of a bus, BU; average queue length at a MM, L; and average waiting time experienced by a PE, W.

3. The Semi-Markov Memory Interference (SMI) Model

A Markov chain which models a multiprocessor system according to the assumptions outlined in Section 2 has an unmanageably large state space, see [1] and [15]. To simplify this we first adopt a technique presented in [9]. In that work separate identical Markov chains are used to describe the behavior of each PE, and the coupling between the N chains appears in the transition probabilities between the states in each chain. Solving the model requires only one of the chains to be considered which dramatically reduces the solution complexity. Moreover, because the chains are coupled, independence of PE's does not have to be assumed, resulting in a more realistic model (assumption I does not imply independence). The number of states in the model of [9] can still grow large, in some cases, because it depends on the number of discrete values $\tilde{T}$ and $\tilde{C}$ can take on. This can be avoided, resulting in a further simplification, by replacing the Markov chains by semi-Markov processes. These have only four states regardless of the distributions for $\tilde{T}$ and $\tilde{C}$. In addition, the semi-Markov processes simplify the computation of the average queue length at each MM and the average waiting time experienced by a PE.

A detailed discussion of semi-Markov processes can be found in [14]. Briefly, a semi-Markov process (SMP) is a stochastic process which can be in any one of K states $1, 2, \ldots, K$. Each time it enters state i it remains there for a random amount of time (the sojourn time) having mean η_i and then makes a transition into state j with probability p_{ij}. As a special case, a discrete time Markov chain is an SMP with a deterministic sojourn time of value one. If the SMP has an irreducible embedded Markov chain that consists of ergodic states, then the limiting probability of being in state i, denoted by P_i, can be expressed as,

$$P_i = \frac{\pi_i \eta_i}{\sum_{j=1}^{K} \pi_j \eta_j} \tag{1}$$

where π_i is the limiting probability of state i in the embedded Markov chain. All the SMP's that appear in this paper have irreducible embedded Markov chains with ergodic states, therefore, equation (1) will always be applicable. The rate of leaving state i, λ_i, is defined as the reciprocal

of the average time elapsed between two consecutive departures from state i. The rate can be obtained using the following equation,

$$\lambda_i = \frac{P_i}{\eta_i} = \frac{\pi_i}{\sum_{j=1}^{K} \pi_j \eta_j} \tag{2}$$

Since the average sojourn time in any one of the states of the SMP's that appear in this paper is at least one system cycle, then λ_i falls in the range $[0,1]$ and it is possible to view λ_i as the probability of leaving state i at the beginning of a bus cycle.

The SMI model uses an SMP to approximate the behavior of a *PE* which functions according to the system operation assumptions given in Sec. 2. Hence, N identical SMP's will approximate the behavior of the multiprocessor system. The SMP in this case is depicted in Fig. 2. The states of the SMP denote the different states of any *PE*. The first state is the thinking state, 0. The process enters state 0 and remains there for a duration of time equivalent to the thinking time of the *PE*. The mean sojourn time in this state is η_0. A memory request is modeled by the SMP leaving state 0. The destination state depends on the state of the requested *MM* and also on whether the memory request is passed by the two levels of the arbitration logic. The second state is the accessing state, 1. The process enters state 1 if the memory module is idle and the memory request is successfully passed by the two levels of the arbitration. The process remains in state 1 for a duration equivalent to the connection time between the *PE* and any *MM*. The mean value of the sojourn time in state 1 is η_1. From state 1 the process returns to state 0, i.e., the *PE* resumes thinking after it has completed its memory access. The third state is the full waiting state, 2. The process enters state 2 when the *PE* requests an idle *MM* simultaneously with at least one other request, and one of these requests obtains access to the *MM* by passing the two levels of arbitration. In this case the *PE* has to wait for the full duration of the connection time between the *MM* and the selected *PE*. The full connection time has a mean value of η_2. A blocked *PE* will try again to access the *MM* at which it is blocked as soon as that *MM* is released. If it succeeds, the process enters state 1; if other *PE*'s requested the same idle memory module simultaneously and one of these *PE*'s obtains the connection with the *MM*, the process

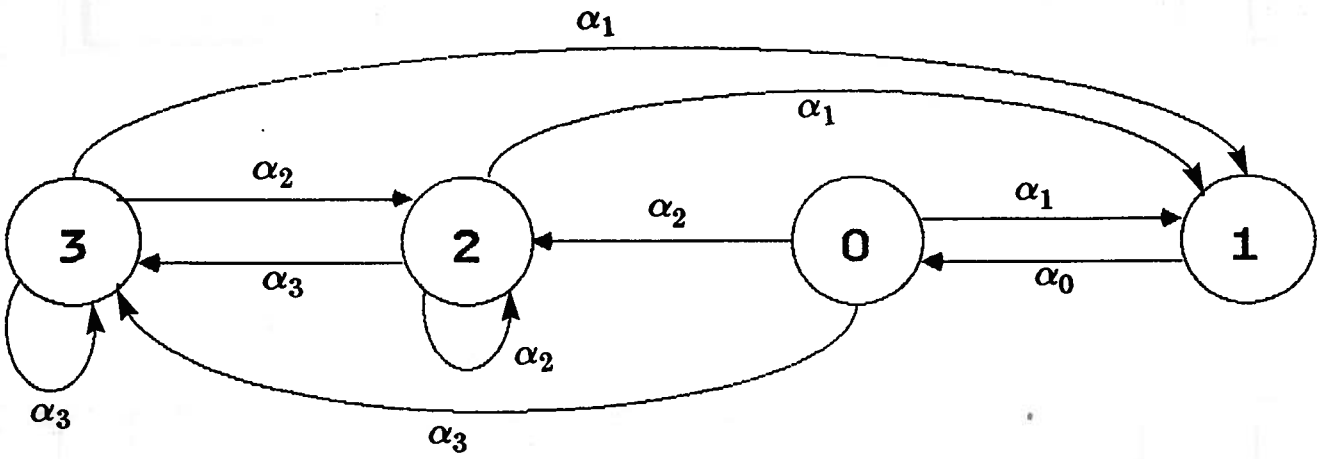

Figure 2 The SMP that describes PE behavior in the multiple-bus system.

reenters state 2; otherwise the process enters state 3. The fourth state is the residual waiting state, 3. The process enters state 3 when the *PE* requests a busy *MM*, or when, due to bus contention, access is blocked to an *MM* even though it is idle. The *PE* has to wait for the residual connection time before retrying to access that particular *MM*. The mean value for the sojourn time in state 3 is η_3. The process then enters state 1 if the *PE* succeeds in accessing the *MM*; or it enters state 2 if the *PE* requests an idle *MM* simultaneously with other *PE*'s and one of these *PE*'s manages to obtain the connection with that *MM*; otherwise it reenters state 3. Clearly, the SMP description does not include which module the *PE* is accessing or which module the *PE* is waiting to access. This does not represent an approximation of the *PE*'s behavior because of the symmetry in this case. The underlying approximation of the SMI model is in describing any *PE* behavior independently from the other *PE*'s while compensating for the coupling between the *PE*'s behaviors in the transition probabilities between the states of the SMP (the coupling results

from the *PE*'s sharing the *MM*'s). The effects of this approximation are explored in detail in[12] for the unit connection time case.

In order to derive numerical information from the SMI model, the values of N, M, B, the first moment of $\tilde{T}$, and the first two moments of $\tilde{C}$ must be obtained. These quantities can be regarded as the input parameters to the model. They are defined as follows:

$$N \triangleq \textit{the number of PE's}$$
$$M \triangleq \textit{the number of MM's}$$
$$B \triangleq \textit{the number of buses}$$
$$\overline{T} \triangleq \textit{the first moment of } \tilde{T}$$
$$\overline{C} \triangleq \textit{the first moment of } \tilde{C}$$
$$\overline{C^2} \triangleq \textit{the second moment of } \tilde{C}$$

The average sojourn times, of the different states of the SMP, can be obtained from the parameters of the model as follows:

$$\eta_j = \begin{cases} \overline{T} & j = 0 \\ \overline{C} & j = 1 \\ \overline{C} & j = 2 \\ \dfrac{\overline{C^2} - \overline{C}}{2(\overline{C} - 1)} & j = 3 \end{cases} \tag{3}$$

The average sojourn times in the states 0, 1 and 2 arises directly from the definition of these states. The average sojourn time in state 3, i.e., the average residual waiting time, is taken from [11].

It is convenient to introduce some terms that will be used in formulating the model. These terms are: R, *BUSY*, *WIN* 1, and *WIN* 2. The term R is defined as the probability that a *PE* makes a request to access a particular *MM* at the beginning of a bus cycle. From our earlier definitions this is the probability of leaving states 0, 2 or 3 to access a particular *MM*. Therefore, R is given by:

$$R = \frac{1}{M}\left(\lambda_0 + \lambda_2 + \lambda_3\right) \tag{4}$$

The term *BUSY* is defined as the probability that a *PE* finds a particular *MM* busy at the beginning of a bus cycle (type 2 conflict). In other words, one of the other $(N-1)$ PE's is accessing that *MM* and is not on the point of releasing it. Hence, *BUSY* is the probability that one of $(N-1)$ PE's is accessing a particular *MM* and the accessing *PE* is not on the point of releasing the *MM*. By definition, the probability that a *PE* is accessing a *MM* is P_1. Thus, the probability that it is accessing and will not leave state 1 (release the *MM*) in the next bus cycle is $P_1 - \lambda_1$. Therefore, *BUSY* can be expressed as,

$$BUSY = \frac{N-1}{M}\left(P_1 - \lambda_1\right) = \frac{N-1}{M}\left(\overline{C} - 1\right)\lambda_1 \tag{5}$$

The term *WIN* 1 is the probability that the memory request initiated by a *PE* passed the first level of arbitration, i.e., one of the M arbiters of the N-users 1-server type selected it. The term *WIN* 1 is derived by the following argument. The probability that a *PE* will not request a particular *MM* is $1 - R$; the probability that none of the N *PE*'s request that *MM* is $\left(1 - R\right)^N$; and therefore the probability that a particular *MM* is requested by at least one of the *PE*'s is $\left[1 - \left(1 - R\right)^N\right]$. One of these requests will pass the first level of arbitration, therefore, the probability that a request from any *PE* passes the first level of arbitration, p, is given by,

$$p = 1 - \left(1 - R\right)^N$$

The expected number of *PE*'s which requested that *MM* at the beginning of a bus cycle is NR. Therefore, *WIN* 1 can be defined as follows:

$$WIN\,1 = \frac{p}{NR} \tag{6}$$

Finally, The term *WIN* 2 is the probability that the memory request initiated by a *PE* will pass the second level of arbitration given that it passed the first level of arbitration, i.e., the arbiter of the M-users B-server type selected the *PE* after it had been passed by one of the M arbiters of the N-users 1-server type. The term *WIN* 2 is calculated by conditioning on the

number of free buses. We need to calculate two quantities. The first, $X(k)$, is the probability that the request will pass the second level of arbitration given there are k free buses and that the request has already passed the first level of arbitration. The quantity $X(k)$ can be expressed as,

$$X(k) = \sum_{i=1}^{M} \frac{min(k,i)}{i} \binom{M-1}{i-1} p^{i-1} (1-p)^{M-i}$$

The factor $min(k,i)/i$ is the probability that if i requests pass the first level then $min(i,k)$ will obtain buses (pass the second level). The factor $\binom{M-1}{i-1} p^{i-1} (1-p)^{M-i}$ is the probability that $(i-1)$ additional requests pass the first level given that one request has with certainty. The second quantity, $Y(k)$, is the probability that there are k free buses. The quantity $Y(k)$ can be derived as follows. The probability that k out of B buses are free is $\binom{B}{k} q^{B-k} (1-q)^{k}$, where q is the probability that a bus is busy. The term q can be found through an argument similar to that used to derive the term $BUSY$, and can be expressed as $(N-1)\frac{(P_1 - \lambda_1)}{B}$. The term $WIN2$ can now be obtained from,

$$WIN2 = \sum_{k=1}^{B} X(k)\, Y(k) \tag{7}$$

The transition probabilities between the states of the SMP can be expressed as the following functions of $BUSY$, $WIN1$ and $WIN2$:

$$\alpha_j = \begin{cases} 1 & j = 0 \\ (1 - BUSY)\, WIN1\, WIN2 & j = 1 \\ (1 - BUSY)\,(1 - WIN1)\, WIN2 & j = 2 \\ BUSY + (1 - BUSY)\,(1 - WIN2) & j = 3 \end{cases} \tag{8}$$

Their derivation proceeds as follows. When the process, shown in Fig. 2, leaves any of the three states 0, 2 or 3 it enters the accessing state (state 1) with probability α_1 if the requested MM is idle and the PE's request successfully passes both levels of arbitration; or the process enters the full waiting state (state 2) with probability α_2 if the requested MM is idle and the PE's request fails to be selected in the first level of arbitration and another request for the same MM passed the second level of arbitration; or the process enters the residual waiting state (state 3) with

probability α_3 if the requested *MM* is busy or the requested *MM* is idle but none of the buses are free. The process always enters the thinking state after it leaves the accessing state ($\alpha_0 = 1$).

The embedded Markov chain can be solved and the π's can be represented as functions of the transition probabilities, i.e., of *BUSY*, *WIN* 1 and *WIN* 2. The SMP limiting probabilities can be derived by substituting the limiting probabilities of the embedded Markov chain (π's) into equation (1). Therefore, the SMP limiting probabilities can be expressed as functions of R and the transition probabilities as follows:

$$P_j = \begin{cases} \eta_0 \alpha_1 M R & j = 0 \\ \eta_1 \alpha_1 M R & j = 1 \\ \eta_2 \alpha_2 M R & j = 2 \\ \eta_3 \alpha_3 M R & j = 3 \end{cases} \tag{9}$$

It can be seen from the above equations that we have to solve a set of simultaneous non-linear equations to solve the SMP of Fig. 2. The non-linearity is introduced because the transition probabilities are defined as functions of the SMP's limiting probabilities, while the SMP's limiting probabilities are defined as functions of the transition probabilities. An iterative algorithm can be used to solve for R and λ_1 from which the performance measures discussed earlier can be derived. The algorithm breaks down as follows:

1. Calculate the average sojourn times of the states using equation (3).
2. Choose an initial value for R in the range $0 < R < 1$ (we used $R = 1/M$), and an initial value for λ_1 (we used $\lambda_1 = 0$).
3. Calculate the terms *BUSY*, *WIN* 1 and *WIN* 2 using equations (5), (6) and (7) respectively.
4. Calculate the transition probabilities using equation (8).
5. Calculate an improved estimate for R by first summing the four equations of equation (9) to one and then calculating R from

$$R = \frac{1}{\left(\eta_0 \alpha_1 + \eta_1 \alpha_1 + \eta_2 \alpha_2 + \eta_3 \alpha_3 \right)}$$

6. Calculate an improved estimate for λ_1 from

$$\lambda_1 = \alpha_1 M R$$

7. Repeat steps 3 through 6 until R and λ_1 have the desired accuracy[1].

The solution for R and λ_1 may be used to calculate the limiting probabilities of the states using equation (9). These can in turn be used to calculate the performance measures from the following equations:

$$\begin{aligned} BW &= N P_1 \\ PU &= P_0 + P_1 \\ MU &= \frac{N}{M} P_1 \\ BU &= \frac{N}{B} P_1 \\ L &= \frac{N}{M} (P_2 + P_3) \\ W &= \frac{\eta_2 \alpha_2 + \eta_3 \alpha_3}{\alpha_1} \end{aligned}$$

The last equation is the only one that does not follow directly from the definition of the states of Fig. 2. It can be derived by calculating the expected value of $\tilde{W}$ in the usual way from the pmf of $\tilde{W}$. The pmf of $\tilde{W}$ can be expressed as follows:

$$Pr[\tilde{W} = (i-j)\eta_2 + j\eta_3] = \binom{i}{j} \alpha_1 \alpha_2^{i-j} \alpha_3^{j}$$

The derivation of the above equations proceeds as follows. The probability that the process moves from state 0 to state 1 after making $(i-j)$ consecutive visits to state 2 followed by j consecutive visits to state 3 is $\alpha_2^{i-j} \alpha_3^{j} \alpha_1$. Since there are $\binom{i}{j}$ combinations of these i visits to states 2 and 3 (not necessarily consecutive visits) then the probability that the process moves from state 0 to state 1 after making $(i-j)$ visits to state 2 and j visits to state 3 is $\binom{i}{j} \alpha_1 \alpha_2^{i-j} \alpha_3^{j}$. The average value of the waiting time in the queue, W, is calculated from the pmf of the waiting time

[1] This is a fixed-point iteration scheme. The Steffensen iteration algorithm was used to accelerate the convergence, see [4]. No more than four iterations were needed in our experiments.

described above. Therefore, W can be expressed as follows:

$$W = \sum_{i=0}^{\infty} \sum_{j=0}^{i} \binom{i}{j} \alpha_1 \alpha_2^{i-j} \alpha_3^{j} \left[(i-j)\,\eta_2 + j\,\eta_3 \right]$$

$$= \frac{\eta_2 \alpha_2 + \eta_3 \alpha_3}{\alpha_1}$$

4. Evaluation of the SMI Model

The empirical evidence reported in [1, 2, 6, 8] led to the assumptions of Sec. 2 as a phenomenological basis for the behavior of a large class of multiprocessors. However, in Sec. 3 an approximation was introduced to allow the construction of a manageable model. This approximation was the assumption that the SMP's for each of the PE's are independent. In this section we examine the consequences of this approximation by comparing the SMI model with simulations based on the assumptions of Sec. 2. The results are all for multiprocessor systems with 8 PE's and 8 MM's, and the effects of varying the other input parameters ($\overline{T}$, $\overline{C}$, $\overline{C^2}$, and B) on the performance measures (BW, PU, L, and W) are shown. The dependence on $\overline{C^2}$ is shown indirectly through dependence on the coefficient of variation, C_v, where $C_v = \sqrt{\frac{\overline{C^2}}{(\overline{C})^2} - 1}$.

In the first case the connection time between a PE and a MM lasts for one cycle and the average think time for a PE is zero cycles. Figure 3 shows the simulation results of BW, PU, L, and W as functions of B. Figure 4 shows the relative percentage error, $\%Error$, between the simulation's results and the model's results. The term $\%Error$ is defined as follows:

$$\%Error = \frac{Results\ from\ the\ model\ -\ Results\ from\ the\ simulation}{Results\ from\ the\ simulation} \times 100$$

Figure 4 shows close agreement between the SMI model's results and the simulation's results. The utilization measures (BW and PU) were within 7% of the simulation. The queue measures (L and W) were within 15% of the simulation.

In the second case the connection time between a PE and a MM lasts for one cycle and the average think time for a PE is one cycle. Figure 5 shows the simulation results of BW, PU, L, and W as functions of B. Figure 6 shows the relative percentage error, $\%Error$, between the simulation's results and the model's results. Again there is close agreement between the SMI model's results and the simulation's results. In this case the model shows similar accuracy to the previous case. By comparing Figs. 3 and 5 we can deduce the effect of the average think time $\overline{T}$ on the system's performance. The memory bandwidth, BW, decreased as $\overline{T}$ increased. The processor utilization, PU, increased as $\overline{T}$ increased. The average queue length and the average

queueing time decreased as $\overline{T}$ increased. This agrees with what one would expect, particularly if the PE's have a cache and $\overline{T}$ is the mean time between faults. Both cases one and two are the same as cases examined in the unit deterministic connection time models ($\overline{C} = 1,\ C_v = 0$) of [7, 10, 12]. They yield exactly the same results. Both cases also point out the observation that for systems with $B << M$ the performance is bus-limited, and for systems with $B > M/2$ the performance is memory-limited.

In the third case the average connection time between a PE and a MM is four cycles and the average think time for a PE is zero cycles. Figure 7 shows the simulation results of BW, PU, L, and W as functions of B and C_v. Figure 8 shows the relative percentage error, $\%Error$, between the simulation's results and the model's results. Again there is close agreement between the SMI model's results and the simulation's results. The utilization measures (BW and PU) were within 8% of the simulation. The queue measures (L and W) were within 15% of the simulation. This case demonstrates the effect of the variation in the connection time on the system performance. The system performance declines as the variation in the connection time, C_v, increases. The memory bandwidth BW and the processor utilization PU decreases as C_v increases. While the average queue length L and the average queueing time W increase as C_v increases. This can be explained from the SMP of Fig. 2. Increasing C_v will increase the average sojourn time in state 3, therefore, P_3 will increase.

In the fourth case the average connection time between a PE and a MM is four cycles and the average think time for a PE is one cycle. Figure 9 shows the simulation results of BW, PU, L, and W as functions of B and C_v. Figure 10 shows the relative percentage error,$\%Error$, between the simulation's results and the model's results.

The last two cases highlight the importance of keeping C_v low. Reducing C_v from 2 to 0 can increase the BW by about 65% (Fig. 7(a)) and more than halve W. In systems where some PE's may be DMA channels that can perform block transfers and other PE's may be simply be transferring cache lines it may be advantageous to break up the block transfers and/or increase the cache line size so that C_v is reduced.

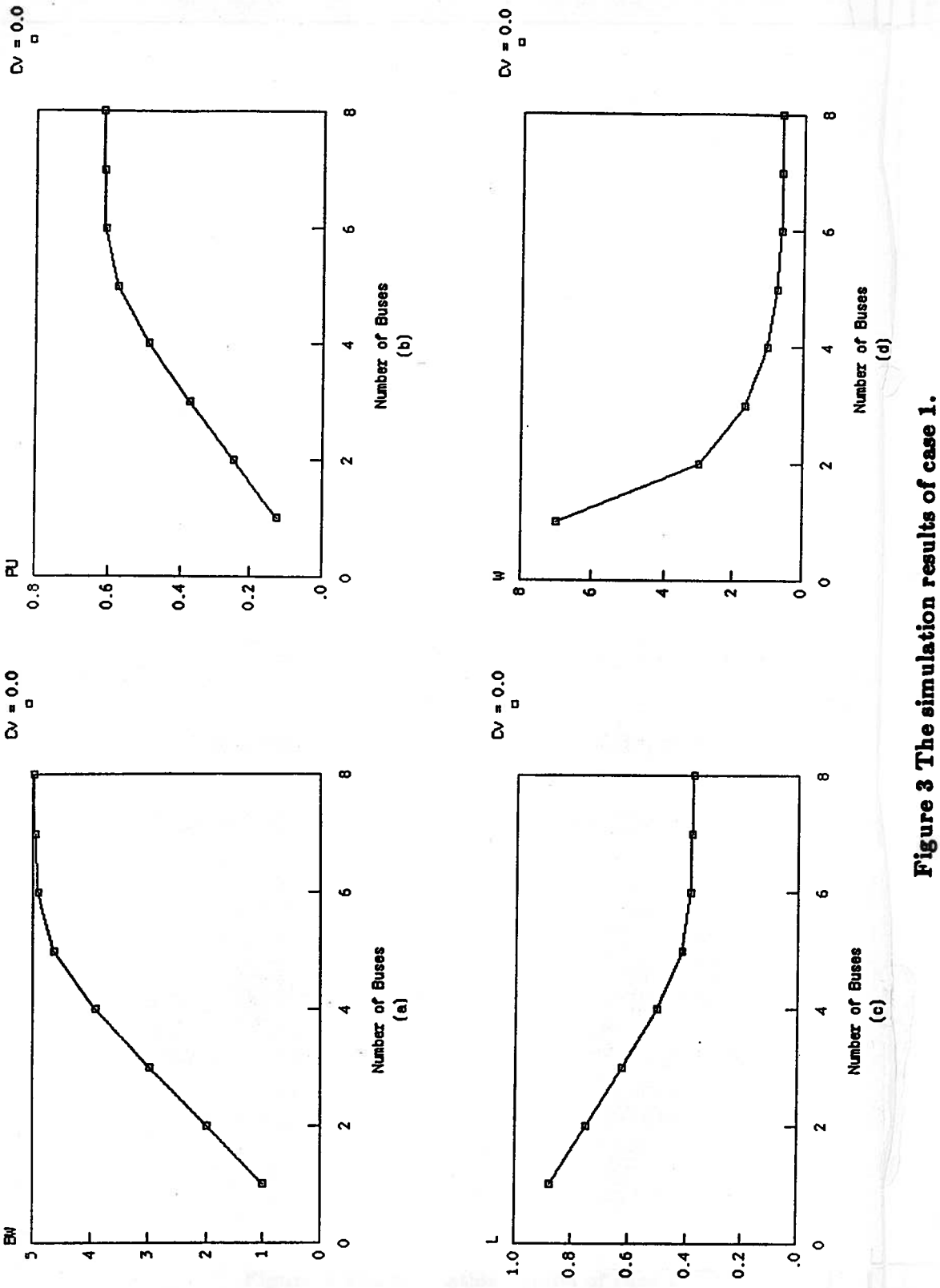

Figure 3 The simulation results of case 1.

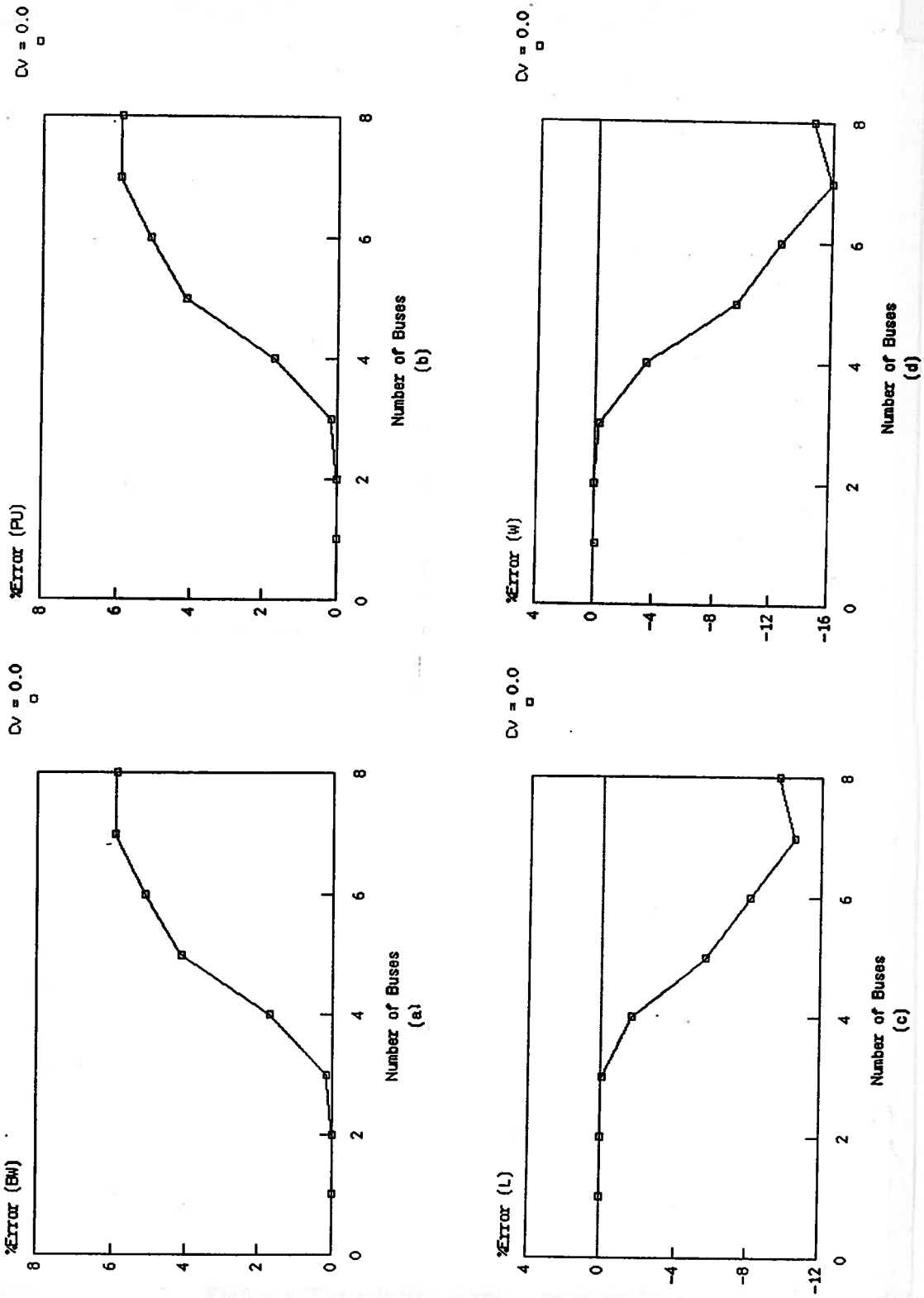

Figure 4 The relative percentage error for case 1.

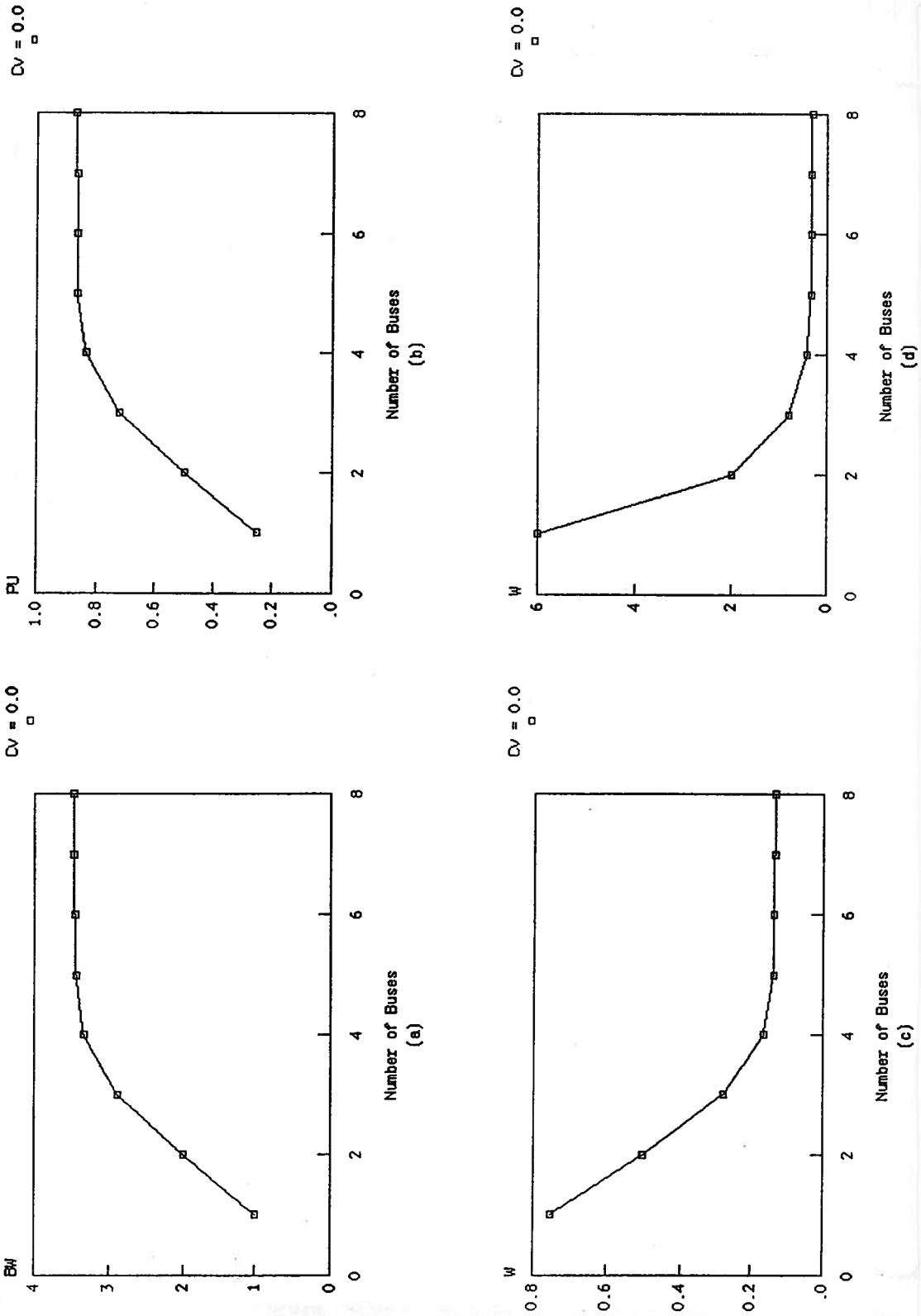

Figure 5 The simulation results of case 2.

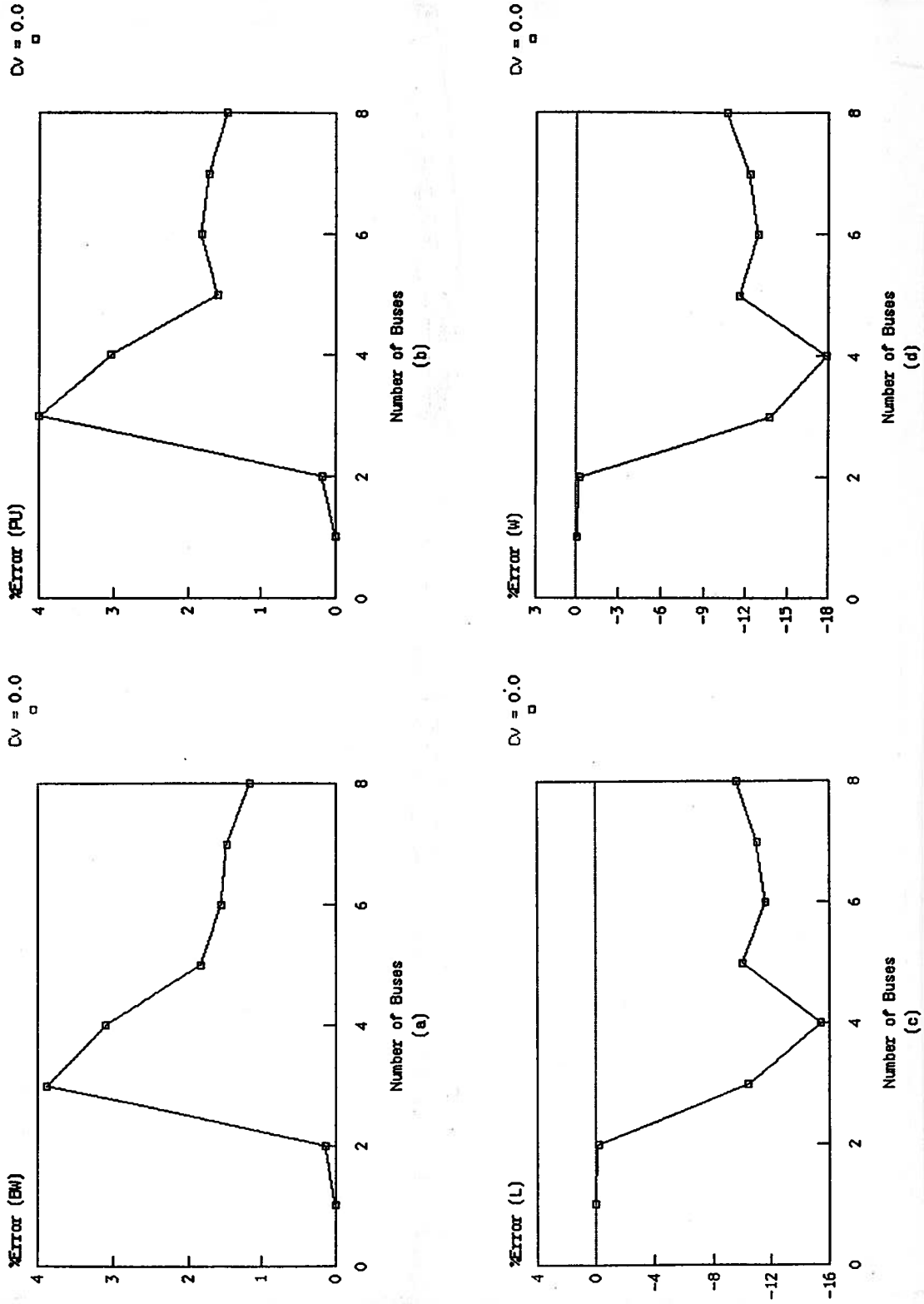

Figure 6 The relative percentage error for case 2.

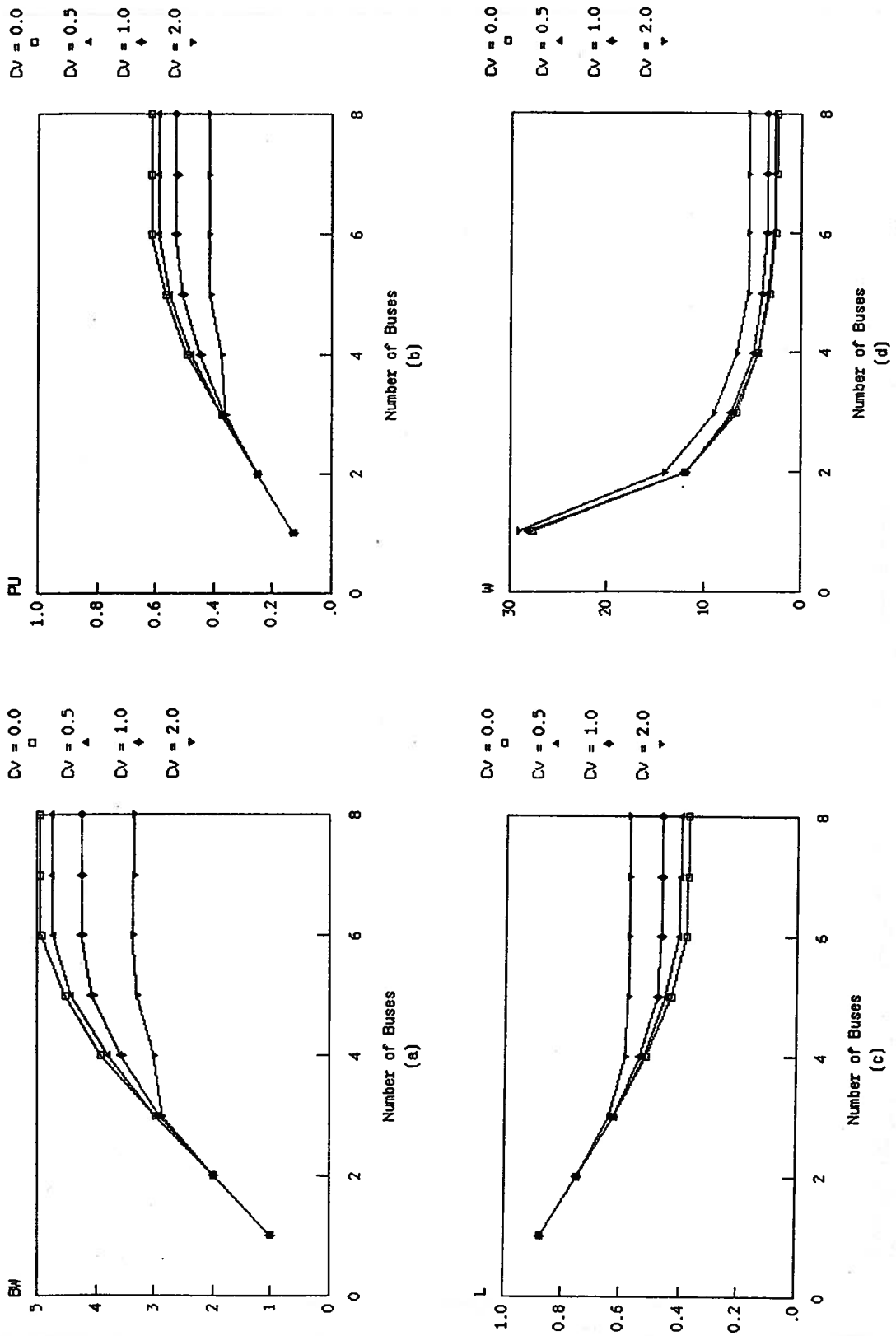

Figure 7 The simulation results of case 3.

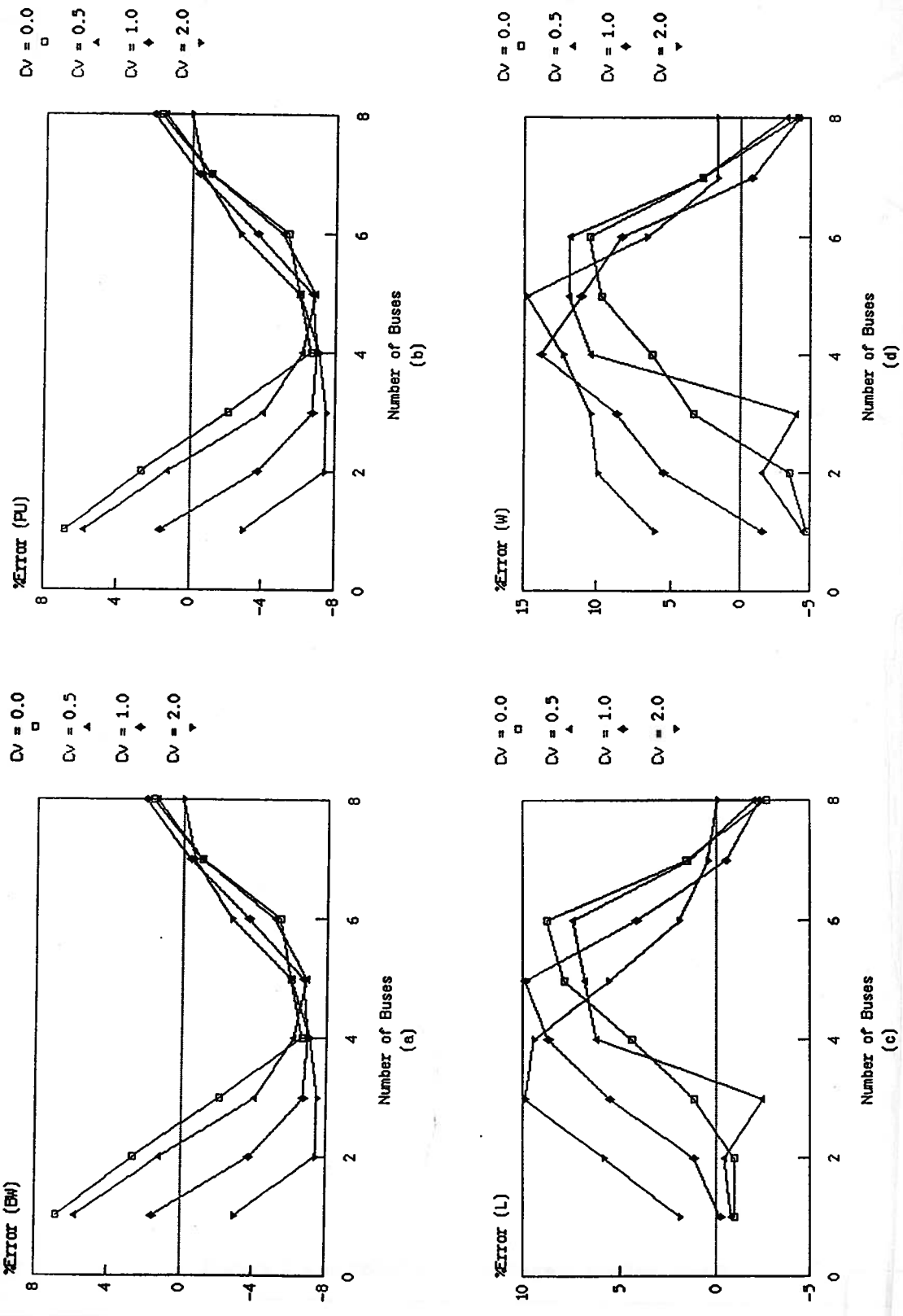

Figure 8 The relative percentage error for case 3.

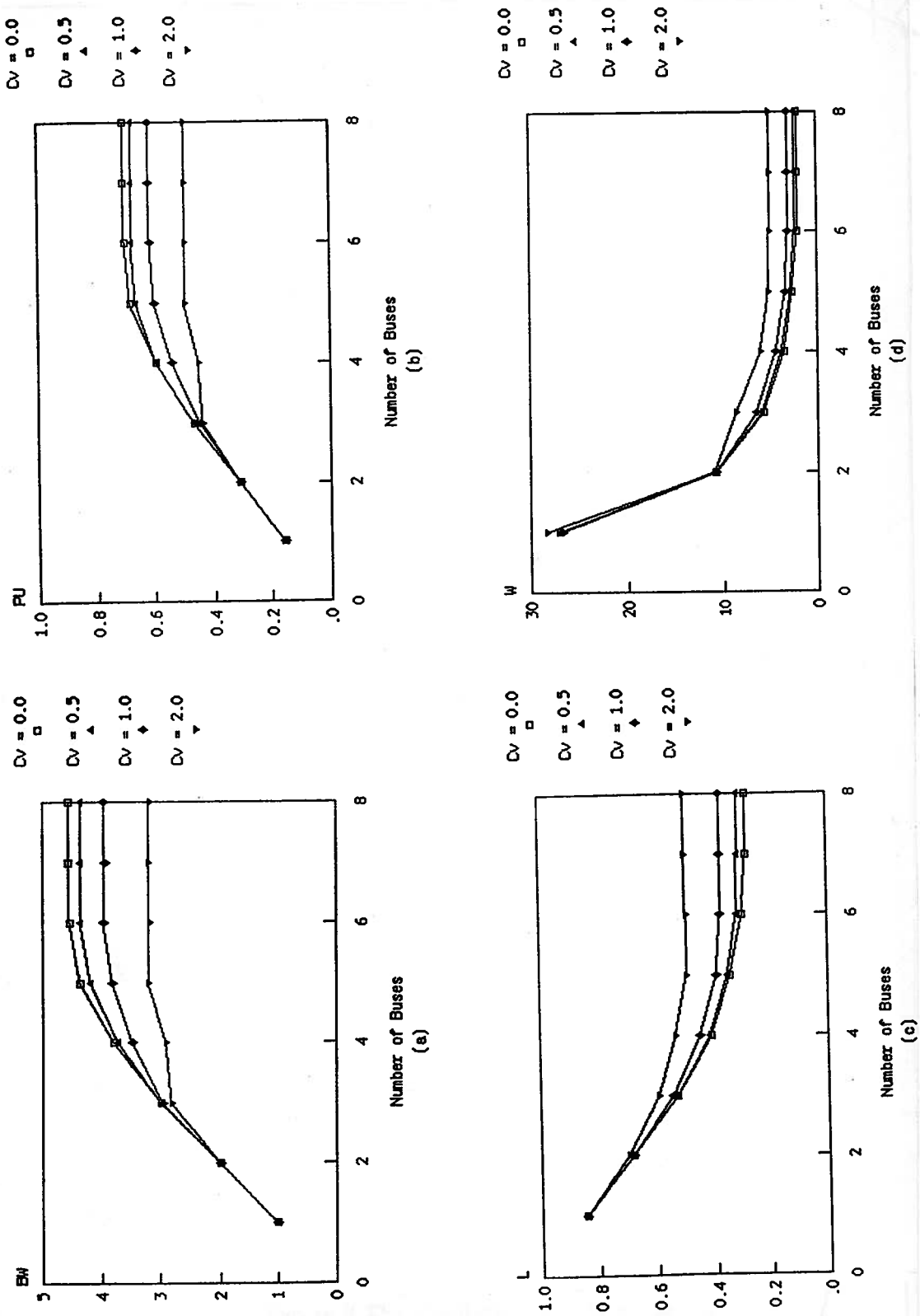

Figure 9 The simulation results of case 4.

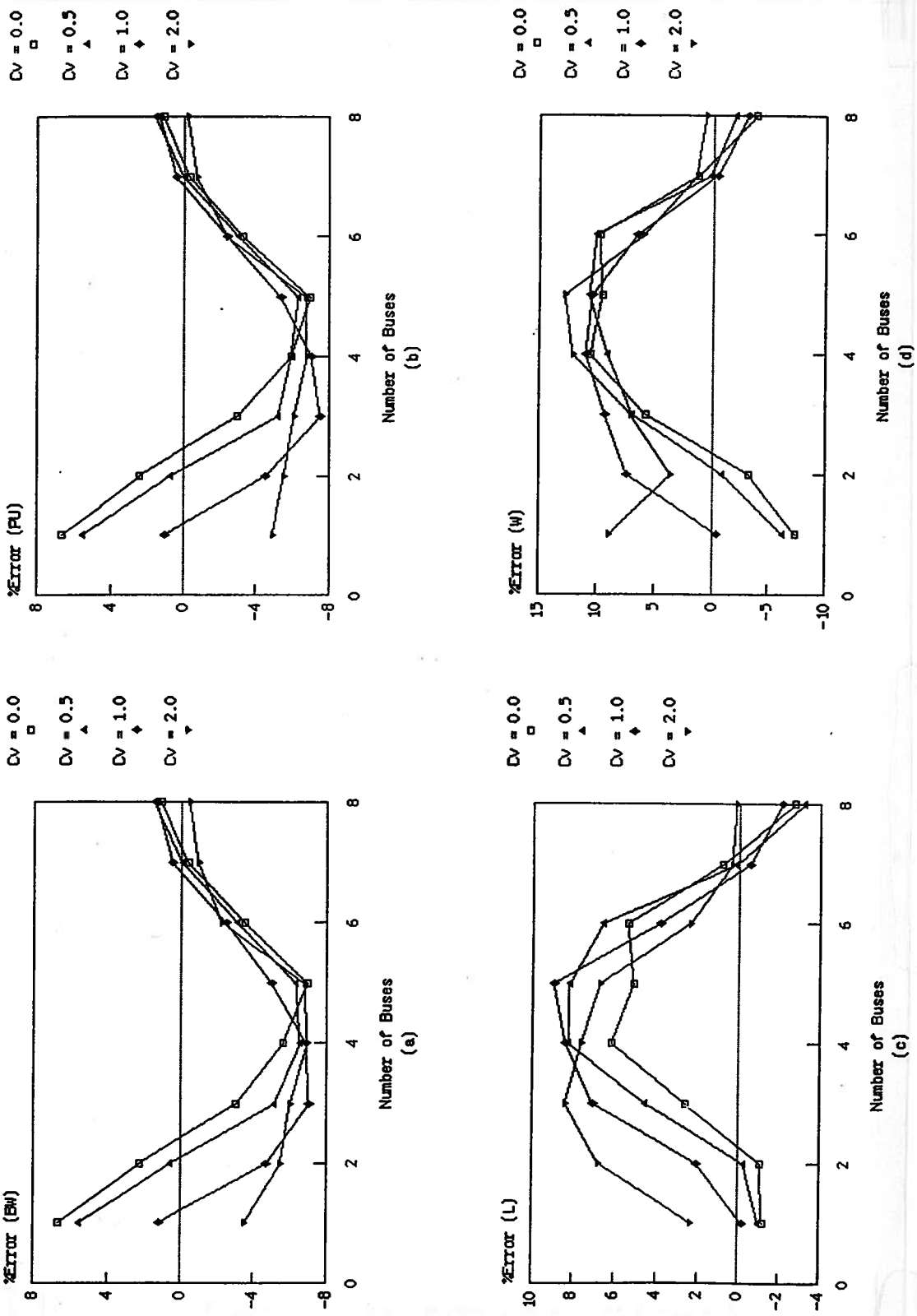

Figure 10 The relative percentage error for case 4.

5. References

[1] F. Baskett and A. J. Smith, "Interference in Multiprocessor Computer Systems with Interleaved Memory," *Comm. of ACM*, vol. 19, no. 6, pp. 327-334, June 1976.

[2] D. P. Bhandarkar, "Analysis of Memory Interference in Multiprocessors ," *IEEE Trans. on Computers*, vol. C-24, no. 9, pp. 897-908, September 1975.

[3] L. N. Bhuyan, "A Combinatorial Analysis of Multibus Multiprocessors," in *Proc. 1984 Int'l Conf. on Parallel Processing*, pp. 225-227, August 1984.

[4] S. D. Conte and C. de Boor, in *Elementary Numerical Analysis* McGraw-Hill Book Company, 1980.

[5] A. Goyal and T. Agerwala, "Performance Analysis of Future Shared Storage Systems," *IBM Journal of Res. and Develop.*, vol. 28, no. 1, pp. 95-108, January 1984.

[6] C. H. Hoogendoorn, "A General Model for Memory Interference in Multiprocessors," *IEEE Trans. on Computers*, vol. C-26, no. 10, pp. 998-1005, October 1977.

[7] T. Lang, M. Valero, and I. Alegre, "Bandwidth of Crossbar and Multiple-Bus Connections for Multiprocessors," *IEEE Trans. on Computers*, vol. C-31, no. 12, pp. 1227-1233, December 1982.

[8] B. A. Makrucki, in *A Stochastic Model of Multiprocessing* Ph.D. Thesis, The University of Michigan, 1984.

[9] T. N. Mudge and H. B. Al-Sadoun, "Memory Interference Models with Variable Connection Time," *IEEE Trans. on Computers*, vol. C-33, no. 11, pp. 1033-1038, November 1984.

[10] T. N. Mudge, J. P. Hayes, G. D. Buzzard, and D. C. Winsor, "Analysis of Multiple-Bus Interconnection Networks," in *Proc. 1984 Int'l Conf. on Parallel Processing*, pp. 228-232, August 1984.

[11] T. N. Mudge, H. B. Al-Sadoun, and B. A. Makrucki, "A Semi-Markov Model for Memory Interference in Multiprocessors," Computing Research Lab Report # CRL-TR-44-84, November 1984.

[12] T. N. Mudge, J. P. Hayes, G. D. Buzzard, and D. C. Winsor, "Analysis of Multiple-Bus Interconnection Networks", in review.

[13] Special Issue on Interconnection Networks, *Computer*, vol. 14, no. 12, December 1981.

[14] S. M. Ross, in *Applied Probability Models with Optimization Applications.* San Francisco, Calif.: Holden-Day, Inc., 1970.

[15] C. E. Skinner and J. R. Asher, "Effects of Storage Contention on System Performance," *IBM Systems Journal*, vol. 8, no. 4, pp. 319-333, 1969.

[16] M. Valero et al., "A Performance Evaluation of the Multiple-Bus Network for Multiprocessor Systems," in *Proc. ACM Conf. on Performance Eval.*, pp. 200-206, 1983.