Computer Science Department Technical Report
University of California
Los Angeles, CA 90024-1596

PERFORMANCE MODELING OF CONCURRENCY CONTROL

Farid Mehovic

March 1989
CSD-890011

UNIVERSITY OF CALIFORNIA

Los Angeles

Performance Modeling of Concurrency Control

A dissertation submitted in partial satisfaction of the

requirements for the degree Doctor of Philosophy

in Computer Science

by

Farid Mehović

1989

The dissertation of Farid Mehović is approved.

Steven A. Lippman

Bruce Rothschild

Alfonso F. Cardenas

Richard R. Muntz

Leonard Kleinrock, Committee Chair

University of California, Los Angeles

1989

to Aida Alibegović

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor, Professor Leonard Kleinrock, for his support, guidance, and encouragement during the course of this dissertation. I would also like to thank the other members of my committee for their time; Professors Alfonso Cardenas, Richard Muntz, Steven Lippman, and Bruce Rothschild. It was a pleasure working with them and with my fellow graduate students; Willard Korfhage, Simon Ming-Yung Horng, Christopher Ferguson, Robert Felderman, and Jau-Hsiung Huang. My thanks extend to all other members of the Computer Science Department, faculty, staff, and past and present students, especially to Verra Morgan, Professors Miloš Ercegovac and Jack Carlyle, Lily Chien, and Doris Sublette.

For giving me an opportunity to pursue graduate studies at UCLA, I sincerely thank the UCLA – University of Sarajevo Student Exchange Program, as well as the Institute for Control and Computer Science, Energoinvest, Sarajevo. I am particularly grateful to Professors Muhamed Aganagić, Bariša Krekić, Božidar Matić, and Suad Alagić.

Most of all, I want to thank my parents, my family, and my friends for supporting me and never giving up on me throughout this endeavor.

VITA

October 21, 1960	Born, Bijelo Polje, Yugoslavia
1982-1983	Research Assistant
	Faculty of Electrical Engineering
	University of Sarajevo, Yugoslavia
1983	Graduate Electrical Engineer
	Computer Science Major
	University of Sarajevo, Yugoslavia
1984	Research Consultant
	Institute for Control and Computer Science
	Energoinvest, Sarajevo, Yugoslavia
1985	M.S. in Computer Science
	University of California, Los Angeles
1987-1989	Teaching/Research Assistant
	Department of Computer Science
	University of California, Los Angeles

PUBLICATIONS

Kantardžić, M., Mehović, F., and Kovačević, S.: "Performance Evaluation of Software Packet for Retrieval: INDOS", International Symposium Jahorina 81, Sarajevo, March 1981.

Kantardžić, M., Mehović, F., and Kovačević, S.: "INDOS: Interactive General Purpose Document Handling System Implemented on Minicomputer", PRAKSA, Yugoslav Journal for Informatics, Year XV, No. 11, November 1980.

Kleinrock. L. and Mehović, F.: "Broadcast vs. Locking", UCLA, CS Technical Report, 1989.

Mehović and F., Kantardžić, M.: "Program Package for Statistical Analysis of Multidimensional Tables". International Symposium, Jahorina 80, Sarajevo, March 1980.

Mehović, F., Kantardžić, M., and Kovačević, S.: "Implementation of Inverted Lists in Document Handling System INDOS", International Symposium Cavtat 81, Cavtat, September 1981.

Mehović, F.: "Design and Implementation of TeleWeb On-line Service", Master Thesis, University of California, Los Angeles, December 1985.

ABSTRACT OF THE DISSERTATION

Performance Modeling of Concurrency Control

by

Farid Mehović

Doctor of Philosophy in Computer Science

University of California, Los Angeles, 1989

Professor Leonard Kleinrock, Chair

An analytical approach to performance modeling of concurrency control in databases is given. Systems where all concurrent transactions conflict are modeled first. The results obtained are then mapped to the realistic cases where concurrent transactions do not necessarily conflict. In some cases the mapping is exact, and in others it is approximate and gives results which in certain domains closely match those obtained through simulations.

The above approach simplifies the analysis and understanding of sharing common resources, in particular, database granules.

The mean transaction response time and "power", defined as system load divided by mean transaction response time, are found for seven different concurrency control scheme models. These are silent-redraw, silent-noredraw, silent/-broadcast-redraw, silent/broadcast-noredraw, broadcast-redraw, broadcast-noredraw, and locking. The first six belong to optimistic concurrency control, while

locking belongs to the class of pessimistic schemes.

The models considered have an infinite number of servers. This corresponds to a database system accessed by workstations, provided the database server has a sufficiently high capacity. The transaction service times consist of a deterministic part and an exponential part. This type of service time distribution includes pure deterministic and pure exponential service times as special cases. Transactions have static data access requirements.

CHAPTER 1

Introduction

1.1 Motivation

There has been much research done in the field of concurrency control in databases, as discussed in Chapter 5. The results given in the literature are typically obtained through iterative numerical calculations. Those few results that have a somewhat simpler form are often found for models that are not truly realistic.

It would be useful to have exact analytic results for some realistic models. It would also be helpful to find out if it is possible to separate queueing problems from problems of conflict due to overlapping resource demand patterns of transactions. Doing so would very much simplify the complex analysis of concurrency control. Would the gain through such a separation be large enough to leave us satisfied even with approximate results? If so, would the future refinements of the same approach give us better, and perhaps exact, results? And finally, could we use the results for simpler models to select the concurrency control scheme that would give us the best performance for a given realistic system?

Before we attack concurrency control in databases, we will try to deal with the problems of sharing common resources in the general case. The database problem per se will be dealt when we reach Chapter 5.

1.2 The Model

1.2.1 Systems with Independently Shared Resources

Consider a system with a number of resources, such as System One shown in Figure 1.1 containing resources A through G. Different shapes of resources drawn in System One represent the possibility of having different types of resources in a system.

Customers arrive to such a system each demanding exclusive access to a subset of the system resources for a given amount of time. Such a subset of the resources we call a *demand set*, and the time for which the access to the demand set is requested we call *service time*. Let C_1, C_2, and C_3 be customers arriving at System One at times t_1, t_2, and t_3, demanding access to subsets $S_1 = \{A, D\}$, $S_2 = \{B, C, E\}$, and $S_3 = \{C, G\}$, for periods of time X_1, X_2, and X_3, respectively. In the systems considered here, customers demand resources independently from each other. In addition to customers' demand sets being mutually independent, so too are their service times. Such systems we call *systems with independently shared resources*, or *ISR systems*. The ISR systems can also be called *systems with specific resource demands*.

Any customer C_i in an ISR system we can describe as a triplet $C_i(t_i, S_i, X_i)$.

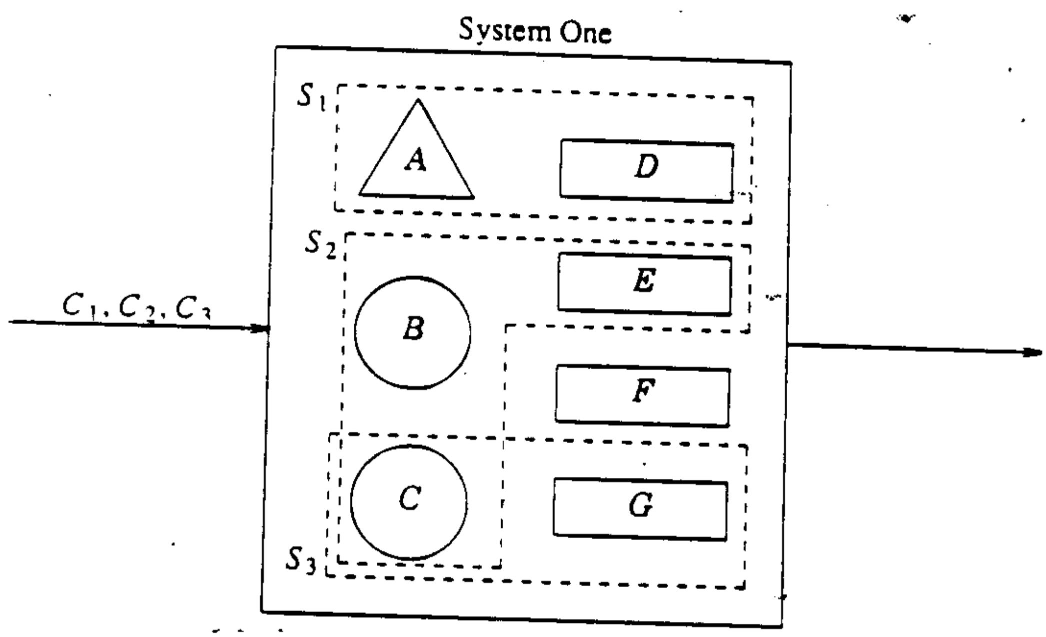


Figure 1.1: System with Independently Shared Resources

The number of resources in a demand set we call *demand set size*. The number of resources in a system is the *system size*.

Concurrent customers are those that happen to be in the system simultaneously. Those concurrent customers that demand access to the same resource(s) *conflict*. In other words, any two concurrent customers conflict if the intersection of their demand sets is nonempty. Assuming that all three customers in Figure 1.1 are concurrent, customers C_2 and C_3 conflict since $S_2 \cap S_3 = \{D\} \neq \emptyset$, while customer C_1 does not conflict with C_2 or C_3 since $S_1 \cap S_2 = \emptyset$ and $S_1 \cap S_3 = \emptyset$.

1.2.2 Partial versus Full Conflict Systems

If the nature of the demand sets is such that any two concurrent customers always conflict, we call such a system a *full conflict system*. If it is possible that two concurrent customers in a system do not conflict, but other conflicts may occur, the system is a *partial conflict system*. If no two concurrent customers ever conflict, the system is a *no conflict system*. System One in Figure 1.1 is a partial conflict system since C_2 and C_3 conflict, but C_1 and C_2 do not. Suppose that the demand sets in System One always include resource B. System One would then be a full conflict system. We would also have a full conflict system if every demand set included either $\{A, B\}$, $\{B, D\}$, or $\{A, D\}$. If demand sets were always empty, we would have a no conflict system.

Note that the G/G/1 queue is a full conflict system, where the common resource is the single server. Since in a G/G/∞ no two concurrent customers ever demand the same server, $G/G/\infty$ is a no conflict system.

Figure 1.2 shows examples of full conflict, no conflict, and partial conflict systems in queueing models.[1] A $G/G/m$ queue, $2 \leq m < \infty$, however, we cannot represent as an ISR system since servers are assigned to customers on the basis of availability.

ISR systems that have only one resource, such as System Two in Figure 1.2, we call *single resource systems*. Systems where every customer requires only one resource, such as System Two and System Four in Figure 1.2, we call *single request systems*. Assuming that demand sets contain at least one resource, except for no conflict systems, all single resource systems are also single request systems. Figure 1.3 shows the different types of ISR systems described in this section and their overlap.

1.2.3 Conflict Resolution and System Parameters

Since conflicting customers demand *exclusive* access to resources from their demand sets, more than one such customer cannot use the resources successfully at the same time. Let us consider several *conflict resolution schemes*, or *concurrency control schemes*, that may be used to guarantee customers exclusive access to their demand sets.

The *silent conflict resolution scheme*, or *silent CRS*, denoted as S, allows every customer to access its demand set right away, with no waiting. Consider customer

[1] The oval in Part b) of Figure 1.2 represents infinite servers.

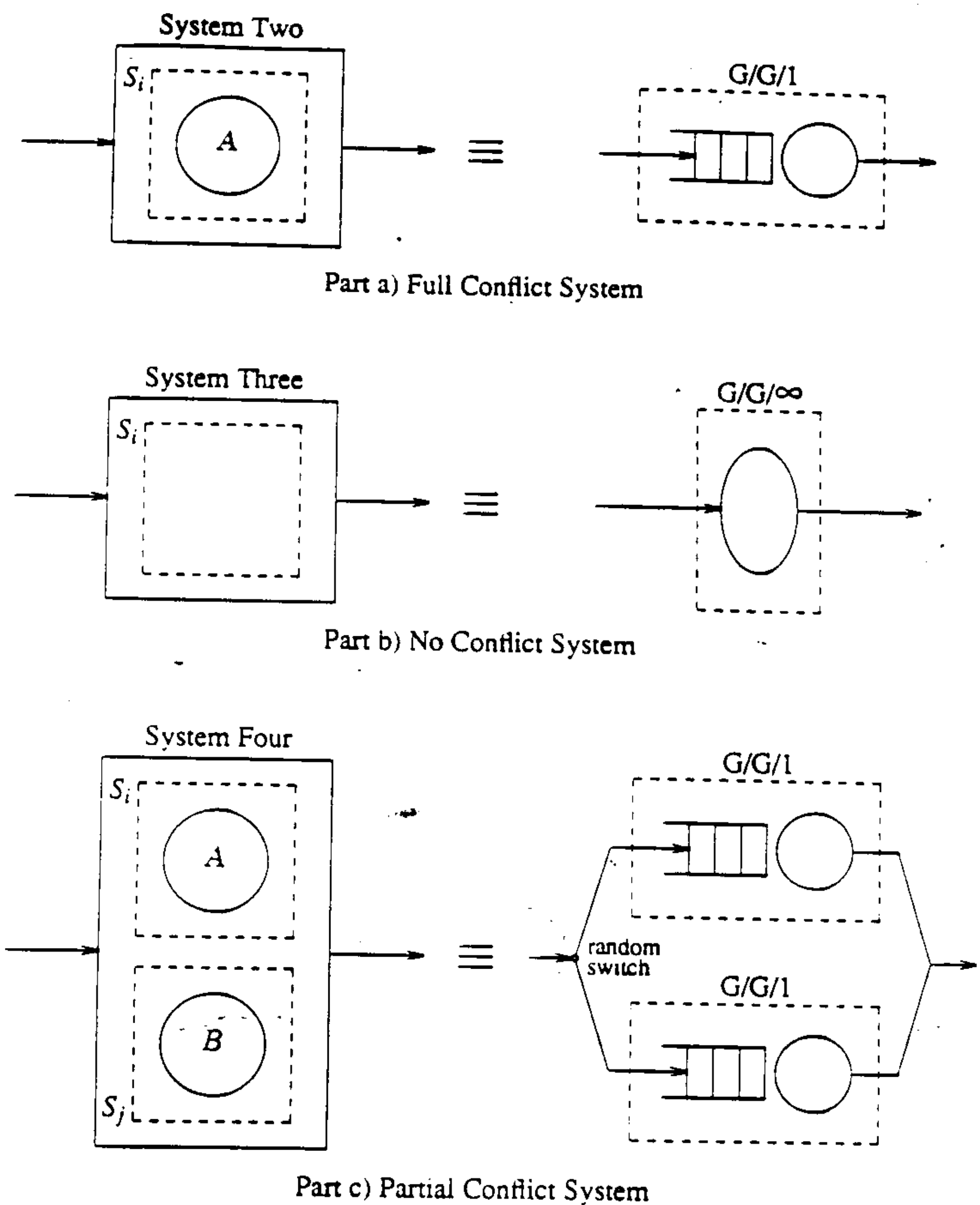


Figure 1.2: Three Types of Systems

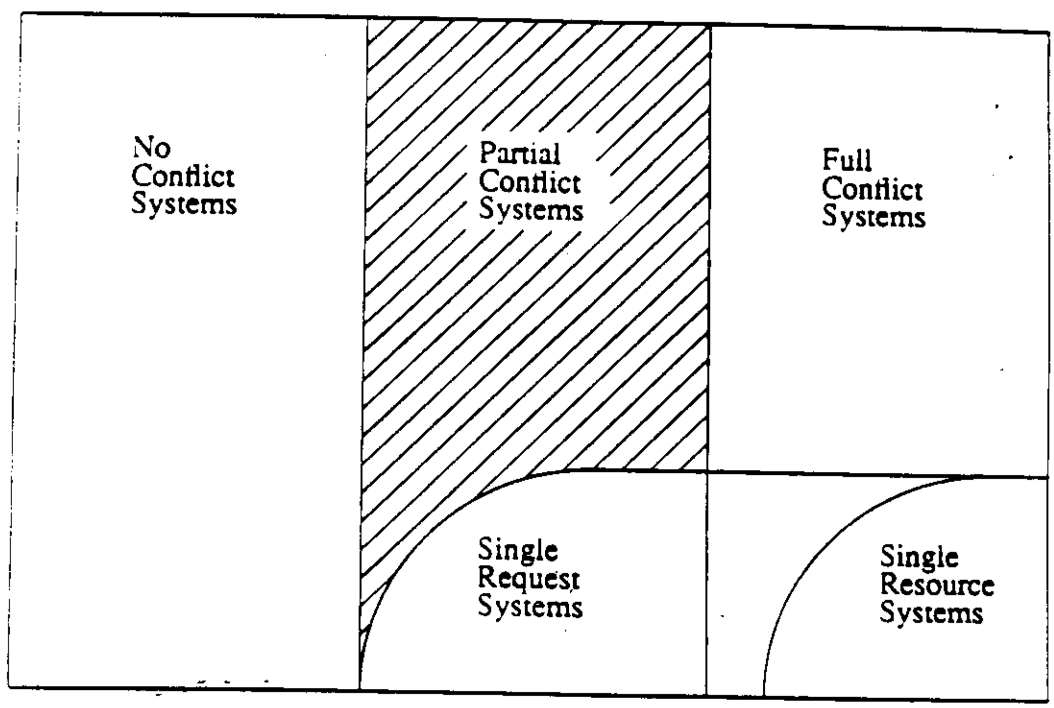


Figure 1.3: Conflict in ISR Systems

i in the system. If, at the end of its service time, any customer conflicting with customer i has left the system, the customer i again demands exclusive access to its demand set. In this case, we say that customer i *restarts*. If no conflicting customers have left the system, customer i leaves the system at the end of its service time. We say that customer i finishes *successfully* and *departs*. If the system broadcasts a message about the departure to all the customers conflicting with the departing customer, then they all restart right away. Such system is a *broadcast CRS*, denoted as B. Suppose now that the broadcast CRS is modified so that every customer completes its *first* service entirely, regardless of any broadcast messages. Such a system is a *silent/broadcast CRS*, denoted as sB. Consider a database under a database management system (DBMS). Customers in such a system are transactions. If a transaction leaves the system, then the DBMS may notify all other transactions about the departure, so that transactions conflicting with the departing one may restart right away.

We distinguish between customers' first (*initial*) service and restarted services. The average length of a restarted service equals the average length of the initial service multiplied by r, the *restarted to initial service ratio*, or just *restart to initial ratio*. Depending on whether the service times of restarted customers remain the same or are redrawn from another distribution, silent, broadcast, and silent/broadcast CRS's can be *redraw* or *noredraw*. For noredraw systems, all restarted service times of one customer are the same, and equal to r times the initial service duration. For redraw systems, restarted service times are drawn from a different distribution whose mean equals r times the mean of the (initial) service time distribution.

Also considered is a combination of silent and broadcast, in which the first (*initial*) service of a customer is not interrupted by system broadcast messages, while their restarted services are. In other words, the system is broadcast except for the first service of every customer which is "silent". We call this system *silent/broadcast* and denote as sB. Specifications for restarted services above apply to this system, too.

Silent and broadcast schemes are known as optimistic concurrency control schemes, since the customers are allowed access to their demand sets right away, hoping that they won't conflict.

A CRS that is pessimistic, is the *locking CRS*, denoted as L. In this scheme every arriving customer waits for all customers conflicting with it to depart, and only then is it granted access to its demand set. The conflicting customers are served in a first-come-first-serve fashion (FCFS).

An ISR system model is fully defined by the total set of resources, T, the CRS model, one of the seven described models, SR, SN, sBR, sBN, BR, BN, or L, the distribution of customers' interarrival times, $t_i - t_{i-1}$, the distribution of customers' service times, X_i, and the distribution of their demand sets, S_i. The size of S_i we denote as $|S_i|$ or s_i. The system size is $|T| = t$. Index i above represents any customer C_i.

The mean response time, also called the *mean system time*, of a customer is defined as the time it spends in the system. This is the time from the moment it arrives until it departs from the system. *Power* is defined as system load divided by the mean response time. The *system load*, or *system utilization factor*, ρ, is defined as the average arrival rate, λ, times the average service time, $\overline{x}$. Service times are modeled as consisting of a deterministic and an exponential component. This distribution we denote as D_qM, where $0 \leq q \leq 1$ represents the deterministic fraction of the average service time. The D_qM distribution includes pure deterministic and pure exponential distributions as special cases.

1.3 Summary of Results

The systems analyzed in this dissertation have an infinite number of servers. We first analyze the performance of full conflict systems, in terms of normalized average response time and normalized power. In order to calculate these values, we also find the distribution of the number of customers in the system.

Once we have obtained results for the full conflict systems, we map the results to partial conflict systems. The mapping is based on the nature of the resource demand patterns of customers. Two types of resource demand patterns are more extensively used. The *random resource demand* pattern assumes that resources in the demand sets are chosen randomly from the common pool of system resources. The *sequential resource demand* pattern assumes that the set of system resources is an ordered set, and any demand set then is a sequence of adjacent resources. In both cases, the demand set size is kept fixed.

Figure 1.4 gives an overview of the results we obtain for ISR systems with different characteristics. Each row in Figure 1.4 represents a different model for conflict resolution. Columns represent different service time distributions: memoryless; D_qM, for $0 < q < 1$; and deterministic. As the table shows, some results are calculated numerically through exact or approximate expressions for transition probabilities, while other results are analytic.

With respect to the level of conflict, results from Figure 1.4 are exactly mapped from full conflict systems to single request partial conflict systems, while mapping to other partial conflict systems are approximate. This is shown in Figure 1.5. No-conflict systems have trivial results and require no mapping. Referring to Figure 1.3, the shaded region represents the "only" systems for which the mapping is not exact. However, most of the realistic systems belong to that region. Approximations for those systems give results close to the simulation in some realistic domains of system load and level of conflict.

We apply results obtained for partial conflict systems to databases with static data access. Only simple winner queues and locking queues are used. The results obtained are exact for some systems and approximate for others. The approximate results give very small errors for a wide range of system load and probability of conflict.

The D_qM nature of the service time distribution and partial restarts may be used as modeling tools to analyze concurrency control overhead, useful fraction

		M	D_qM	D
SR,r	M	[E]		[A]
	$D_{q_r}M$			
	D	[A]		[A]
SN,r				[A]
sBR,r	M	[E]		[A]
	$D_{q_r}M$	[E]		[A]
	D	[E]		[A]
sBN,r				[A]
BR,r	M	(E†)	[E]	[E]
	$D_{q_r}M$	[E]	[E]	[E]
	D	[E]	[E]	(E†)
BN,r				(E†)
L		(E)	(E)	(E)

□ numerical results

○ analytic results

E exact transition probabilities

A approximate transition probabilities

† analytic for $r = 1$, numerical for all r

Figure 1.4: Overview Table of Results

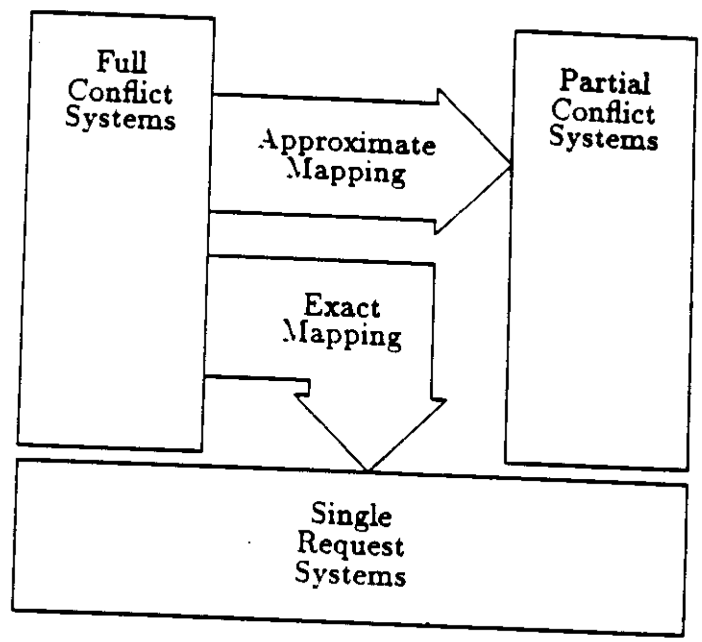


Figure 1.5: Mapping Results to Partial Conflict Systems

of transaction processing, and communication and interactive delays.

1.4 Structure of the Dissertation

The model, the parameters, and the performance measures are described in Chapter 2.

Chapter 3 contains the analysis of full conflict systems. The performance of some of the CRS models are recognized as being equivalent to the performance of regular queues, such as G/G/1, in the case of full conflict. For other models we first define special types of queues, "winner queues", which are equivalent to the full conflict case of those CRS models.

Partial conflict systems are covered in Chapter 4. A mapping from full to partial conflict systems is given, based on the level of conflict, which is given through the *conflict measure* defined in the same chapter. The cases for which the mapping is exact, and those for which the mapping is approximate, are also given. Finally, the domains of high and small errors are discussed.

Results from full conflict systems and mapping to partial conflict systems are applied to database concurrency control in Chapter 5. After a brief survey of results given in the literature, different database systems are described which can be modeled using the analysis given in the previous chapters as modeling tools. Then we show performance curves for different concurrency control schemes and resource demand patterns. We also discuss the possibility of selecting the concurrency control scheme that would give the best performance for a given database system.

The conclusion is given in Chapter 6. There we summarize the results and give pros and cons of the method of analysis. Areas and perspective of further research are then specified.

CHAPTER 2

Model

2.1 Input Parameters

The parameters used for the analysis of the IRS systems are given in Table 2.1.

Conflict resolution schemes are described in Section 1.2.3. They are: silent-redraw (SR), silent-nonredraw (SN), silent/broadcast-redraw (sBR), silent/broadcast-nonredraw (sBN), broadcast-redraw (BR), broadcast-nonredraw (BN), and locking (L).

Arrivals are assumed to be Poisson, i.e., the interarrival time probability density function is $a(t) = \lambda e^{-\lambda t},\ t \geq 0$. Distributions of initial and restarted service times consist of deterministic and exponential components. It is given as

$$b(x) = \begin{cases} 0, & 0 \leq x \leq q\overline{x} \\ \dfrac{\mu}{p} e^{-(\mu x - q)/p}, & x > q\overline{x} \end{cases} \tag{2.1}$$

$$b_R(x) = \begin{cases} 0, & 0 \leq x \leq rq_r\overline{x} \\ \dfrac{\mu}{rp_r} e^{-(\mu x - rq_r)/(rp_r)}, & x > rq_r\overline{x} \end{cases}$$

where $q + p = 1$. The above distribution type is described in Section 2.4.

Demand set distributions considered here are random (Rnd) and sequential (Seq), with the size of demand sets kept fixed at $s = |S|$. In the random distribu-

	Input	Function	Parameters	
1	Conflict Resolution Scheme	CRS $\in$ {SR,SN,sBR,sBN,BR,BN,L}		
2	Poisson Arrivals	$a(t)$	$\lambda \geq 0$	
3	Distribution of (Initial) Service Times D_qM	$b(x)$ $B(x)$	$\mu \geq 0$ $0 \leq q \leq 1$	$\rho = \lambda/\mu$ $p = 1 - q$
4[1]	Distribution of Restarted Service Times $D_{q_r}M$	$b_R(x)$ $B_R(x)$	$r \geq 0$ $0 \leq q_r \leq 1$	 $p_r = 1 - q_r$
5	System Set		T	$t = \|T\|$
6	Demand Set Distribution	$\sigma(S)$	$\sigma \in$ {Rnd,Seq} $0 \leq s = \|S\| \leq t$	
7	Infinite Number of Servers			

Table 2.1: Input Parameters

tion all s resources are chosen independently from the system set. In the case of sequential distribution, we assume that the system set of resources is an ordered set. The resources in every demand set are adjacent to each other, forming a sequential array. The first element of the array is randomly chosen from the system set.

Finally, we consider only systems with an infinite number of servers.

2.2 Utilization Factor

The utilization factor, or *system load*, in ISR systems is defined as the ratio of the average service time $\overline{x}$ and the average service interarrival time $\overline{t}$, $\rho = \overline{x}/\overline{t}$. With the definition of the average arrival rate $\lambda = 1/\overline{t}$, we can write $\rho = \lambda\overline{x}$. It may seem unusual to define the utilization factor like this in a system with an infinite number of servers. In $G/G/\infty$ queues, Kleinrock [12] defines system load to be $\rho = lim_{m\to\infty}(\lambda\overline{x}/m)$, where m is the number of servers, and ρ is the load *per server*. It is obvious that such a definition of ρ agrees with the fact that $G/G/\infty$ queues are always stable, i.e., ρ never reaches 1. In the ISR systems, however, due to conflicts of customers accessing the same resources, there are any number of interarrival and service time distributions for which ISR systems become unstable, i.e., ρ equals or exceeds 1, for a finite λ. The simplest example is the single-resource ISR system with locking, which is equivalent to $G/G/1$, as described later in Section 3.4.4. Instability, or queueing itself, of a $G/G/1$ queue is caused by sharing the same server, while in the single-resource ISR system it is caused by sharing a resource other than the server.

For the redraw ISR systems, due to resampling service times upon restarts, the average service time of the departing customers, $\overline{x_w}$, may become smaller than the actual average (initial) service time $\overline{x}$. This may cause the redraw ISR systems to remain stable for values of λ larger than $1/\overline{x}$. In particular, those systems will be unstable for $\lambda \geq 1/\overline{x}_w \geq 1/\overline{x}$. Our definition of ρ is $\lambda\overline{x}$ for all the ISR systems analyzed here.

2.3 Performance Measures

We are interested in the average time a customer spends in the system – average response time T. Using Little's result, we can represent the average response time as $T = N/\lambda$, where N is the average number of customers in the system. The performance graphs will contain normalized average response time and normalized power, described below.

2.3.1 Normalized Average Response Time

Normalized average response (system) time is defined as $T_n \stackrel{\text{def}}{=} T/\overline{x} = N/\rho$. The normalized response time of a customer in an empty system equals unity, $T_{n(min)} = 1$, and that is the best achievable performance. Graphs of the normalized average response time will show "perfect" performance as a dashed line. $T_n = 1$. However, there will be systems with performance better than perfect, as shown in Chapter 3. Those systems, of course, are not realistic.

2.3.2 Normalized Power

Normalized power P is defined as the ratio of the utilization factor and the normalized average response time, that is, $P = \rho/T_n = \rho^2/N$. Since the average response time of a customer in an empty system is unity, the normalized power then equals load, and that is the best achievable performance. Graphs of the normalized power will show "perfect" performance as a dashed line, $P = \rho$.

2.4 Probability Distribution D_qM

Consider a random variable Y defined as the sum of two random variables. Y_D and Y_M. Y will have a mean equal to $\overline{y}$. Let Y_D be deterministic with the value $Y_D = q\overline{y}$, where $0 \le q \le 1$, and $\overline{y}$ is an arbitrarily defined value. Let Y_M be drawn from the exponential distribution with the mean equal to $p\overline{y}$, where $q + p = 1$. Then, the following holds: $Y = Y_D + Y_M$, and $\overline{Y} = \overline{y}$. The probability distribution of the random variable Y is shown in Figure 2.1. We say that such a distribution is *q-deterministic* and *p-memoryless*, and we denote it as D_qM *distribution.*

The following defines D_qM analytically.

$$
\begin{aligned}
P[Y \le y] &= \begin{cases} 0, & 0 \le y \le q\overline{y} \\ 1 - e^{-(\mu y - q)/p}, & y > q\overline{y} \end{cases} \\
\frac{d}{dy} P[Y \le y] &= \begin{cases} 0, & 0 \le y \le q\overline{y} \\ \frac{\mu}{p} e^{-(\mu y - q)/p}, & y > q\overline{y} \end{cases}
\end{aligned}
\qquad (2.2)
$$

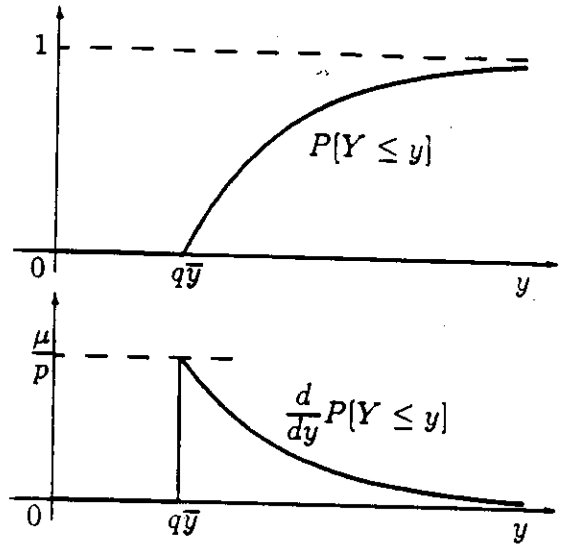


Figure 2.1: $D_q M$ Probability Distribution

where $\mu = 1/\overline{y}$. The $D_q M$ distribution includes two of the often utilized distributions as special cases. For $q = 0$ we have a pure exponential distribution, while for $q = 1$ we have a pure deterministic distribution:

$$
\begin{aligned}
P[Y \leq y]_{q=0} &= 1 - e^{-\mu y}, \; y \geq 0 \\
P[Y \leq y]_{q=1} &= \begin{cases} 0, & 0 \leq y \leq \overline{y} \\ 1, & y > \overline{y} \end{cases}
\end{aligned}
$$

The $D_q M$ probability distribution is used for modeling service time requirements of the customers. Consider a database system where transactions need a fixed amount of time for communication and database access, and an additional time of random length for the computation which is application dependent. The service time requirements of the such transactions may be modeled by the $D_q M$-distribution, where $q\overline{x}$ is the fixed part, and $(1-q)\overline{x}$ is the average computation time.

CHAPTER 3

Full Conflict Systems

3.1 Introduction

Full conflict systems are those ISR systems in which any two concurrent customers necessarily conflict. This chapter contains results of simulation and analysis for the seven different conflict resolution schemes. Some of the results are analytical. some are obtained numerically by solving a set of linear equations, and some are obtained numerically using an integral formula. Several systems are found to be equivalent to M/G/1 queues with different scheduling disciplines. Other systems are equivalent to a special class of queues, called *winner queues*. whose description and analysis are given in Section 3.2. More general winner queues, *winner queues with partial restarts*, are given in Section 3.3. Section 3.6 summarizes the results on full conflict systems.

3.1.1 Notation

We adopt the following notation for full-conflict ISR systems. The types of interarrival and (initial) service time distributions, and the number of servers are specified in the same manner as for the regular $G/G/m$ queues. In addition, next to the specification of the initial service time distribution, separated by a dash we also specify the distribution type of the restarted service times. Following that, in parentheses we specify the conflict resolution scheme and the initial to restart ratio r. For example, a full-conflict ISR system with broadcast-redraw CRS, Poisson arrivals, memoryless (initial) service time distribution, and q_r-deterministic restarted service time distribution is denoted as M/M-D_{q_r}M(BR,r)/∞. Since we are going to deal only with systems with infinite servers, we will omit the specification of the number of servers. Thus, system M/D_qM-D(SN,0.5), specifies a full-conflict ISR system with Poisson arrivals, q-deterministic initial service time distribution, deterministic service time distribution with the mean equal half of the mean of the initial service time distribution, and silent-noredraw CRS. The above definition of the notation for the full-conflict ISR systems is given in Figure 3.1.

If the restart-to-initial ratio equals 1, it may be omitted. If the restarted service time distribution is of the same type as the initial service time distribution, it may be omitted as well. For example, both service time distributions in system M/$D_{0.5}$M(sBR,r) are 0.5-deterministic; however, the average restarted service time is still r times the average initial service time. M/M-D(BN) is an ISR system with Poisson arrivals, memoryless initial service times, deterministic restarted service times with the mean equal to that of the initial service times ($r = 1$), and broadcast-noredraw CRS. System M/M(L) has Poisson input, memoryless service times, and locking CRS. In the case of locking CRS we specify neither r nor restarted service distribution since there are no restarts.

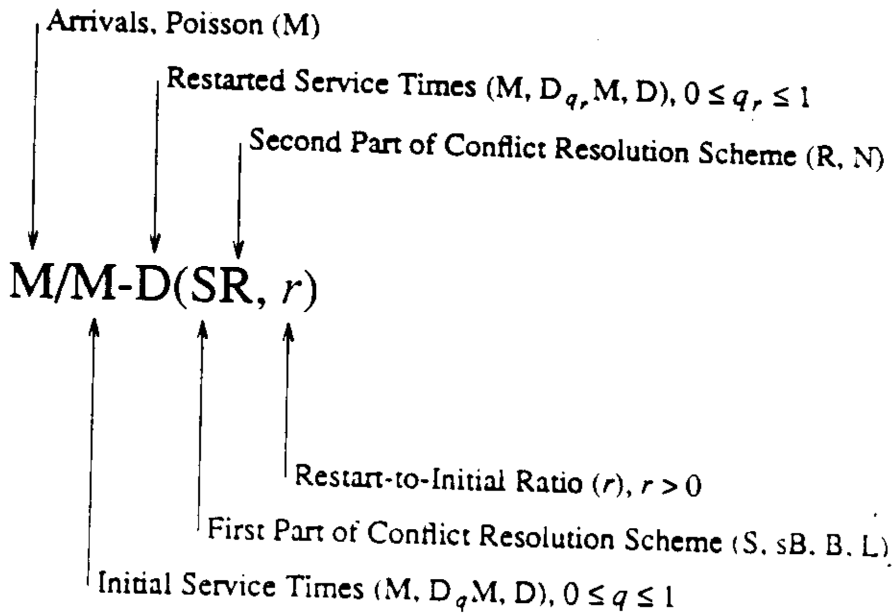


Figure 3.1: Notation for Full-Conflict ISR Systems

If we want to specify more general cases of certain CRS's, we may omit part or entire specification of CRS. So, M/M(sB) specifies both redraw and noredraw silent/broadcast full-conflict ISR systems. M/M(R) specifies all redraw full-conflict ISR systems, while M/M() specifies all full-conflict ISR systems with Poisson arrivals and memoryless initial and restarted service time distributions. Note that the systems where both service time distributions are deterministic, redraw and noredraw cases are equivalent. Thus, M/D(S,r) is eqivalent to both M/D(SR,r) and M/D(SN,r).

3.2 Simple Poisson Winner Queues

3.2.1 Definition and Structure

Consider a regular G/G queue with an arbitrary interarrival time density $a(t)$, and an arbitrary service time density $b(x)$. Assume now that each of the customers accesses the same resource. Let a given customer i start his service at time t_i, with service time X. If during the time interval $(t_i, t_i + X)$ none of the other customers leave the system, then, at time $t_i + X$, customer i will finish his service successfully and leave the system. In this case we say that customer i *wins*. If, on the other hand, some other customer j leaves the system in $(t_i, t_i + X)$, say at time $t_j + X_W$, where $t_i < t_j + X_W < t_i + X$, then customer i *loses*. In the former case, customer i is called a *winner*, while in the latter case, customer i is called a *loser* and customer j is a winner. Every time a customer loses, he restarts his service, and he does that over and over again until finally wins. A queue with this discipline we call a *winner queue*.

We differentiate between two types of systems, depending on the behavior of the system upon the departure of a winner. Consider again that customer i starts his service at time t_i, with service time X, and that customer j wins at time $t_i + X_W, t_i < t_j + X_W < t_i + X$. If, at time $t_i + X_W$, the system notifies all the other customers about the departure of customer j, then, customer i learns that he lost right away, and restarts his service immediately, i.e., at time $t_j + X_W$. In this case we say that the system *broadcasts* that a departure took place. We call this queue a *broadcast winner queue*. If the system does nothing upon a departure of a customer, i.e., if it remains *silent*, then all other customers do not learn that they lost until they finish their present service. This means that our loser i from the above example will restart not at time t_j+X_W, but at time t_i+X. A winner queue with this behavior we call a *silent winner queue*.

In addition to winner queues being silent or broadcast, the properties of service times *upon restart* of losers divide all winner queues into *redraw* and *noredraw* winner queues. In the redraw systems a service time of each restart is redrawn from the same service time distribution, $B(x)$, with density $b(x)$. Our loser, customer i, from the above example, will, then, be scheduled to finish his restarted service at time t_j+X_W+X', in the broadcast case, or at time t_i+X+X', in the silent case, where both X and X' are drawn from the same distribution, $B(x)$. In the noredraw systems, service times upon each restart are equal to the initial service time (they are *not redrawn*). So, the loser i, in such a system, will be rescheduled to finish his restart at time t_j+X_W+X, in the broadcast case, or at time t_i+2X, in the silent case.

The winner queues that are considered have infinite number of servers. They have the same notation as the full-conflict ISR systems. In fact, the winner queues *are the full-conflict ISR systems with optimistic CRSs*. However, the winner queues described in this section only cover ISR systems with restart-to-initial ratio 1 and identical initial and restarted service time distributions. We shall refer to those queues as *simple winner queues*. Other, more general, ISR systems are analyzed through the *winner queues with partial restarts* in Section 3.3.

3.2.1.1 Why Winner Queues?

The winner queues may be used to analyze optimistic concurrency control in databases. Let database transactions be the customers, and let the database be the resource shared by the customers. Let every customer have his own processor, such as in the case where every user executes his program on his own personal computer, and all PC's access the same database residing on a mainframe. Such a system can be modeled with a winner queue, if every customer accesses the entire database. Even in the case where customers access only part of the database, winner queues can be the basis for analysis of concurrency control. Such analysis is carried in Chapters 4 and 5.

Another type of queue, "with a precedence-based queueing discipline", used for the analysis of static locking is described in [36].

3.2.1.2 Structure

Here we study winner queues with Poisson arrivals, *Poisson winner queues*, with D_qM-distributed service times, $M/D_qM(CRS)$ queues. We define an embedded Markov chain at (winner) departure instants in time, and represent states by the number of customers in the system left by departures. Due to the memoryless arrivals, the average number of customers in the system left by departures equals the average number of customers in the system N. Using Little's result, we find the normalized average service time as $T_n = T/\overline{x} = N/\rho$, and then we calculate the normalized power as $P = \rho/T_n = \rho^2/N$.

The D_qM distribution of service times allows us to investigate how the determinicity of service times affects the performance of the system. We vary q from 0 to 1, i.e., we vary the distribution of service times from purely exponential to purely deterministic. We shall see that the level of determinicity, q, affects redraw and noredraw systems differently.

In Section 3.2.2 we first describe and discuss simulation results for four types of simple winner queues: silent-redraw, silent-noredraw, broadcast-redraw, and broadcast-noredraw. In Section 3.2.3 we explain the analytical approach and in Section 3.2.4 through Section 3.2.6 we give numerical results for winner queues M/M(SR), M/D(S), and M/D_qM(BR). A summary of the simple winner queues is given in Section 3.2.7. In that section we specify what systems from Figure 1 are covered with the simple winner queues analyzed in this chapter.

3.2.2 Simulation Results

In the four simulation runs, results for the queues specified in Table 3.1 are obtained.

Figures 3.2 and 3.4 show that redraw systems with service times more deterministic, i.e., with higher q, perform worse than redraw systems with lower q. In redraw systems pure exponential service times give the highest power, while

System	Parameters
M/D$_q$M(SR)	$q = 0, 0.1, 0.2, \ldots, 0.9, 1$
M/D$_q$M(SN)	$q = 0, 0.25, 0.5, 0.75, 1$
M/D$_q$M(BR)	$q = 0, 0.1, 0.2, \ldots, 0.9, 1$
M/D$_q$M(BN)	$q = 0, 0.25, 0.5, 0.75, 1$

Table 3.1: Simulation Runs for Poisson Winner Queues

pure deterministic give the lowest power. Quite the opposite is the situation with the noredraw systems, as illustrated in Figure 3.3 and 3.5. Here, the worst performance is in systems with $q = 0$ (pure exponential). The initial service times are independent of whether the system is a redraw or a noredraw one. It is the service times upon restarts that affect the performance differently. In redraw systems service times upon restarts tend to be smaller and smaller with decreasing q due to the nature of service time probability distribution. In noredraw systems, however, the customers with long initial service times will negatively affect the average response time because they have a small chance of winning and their service times upon restarts will stay fixed at the initial high value. Furthermore, the smaller q is, the higher the probability of service time being long. And so, the noredraw systems with less deterministic service times perform worse than those with more deterministic service times.

From Figure 3.2 through 3.5 we see that redraw systems perform better than

M/D $_q$M(SR)

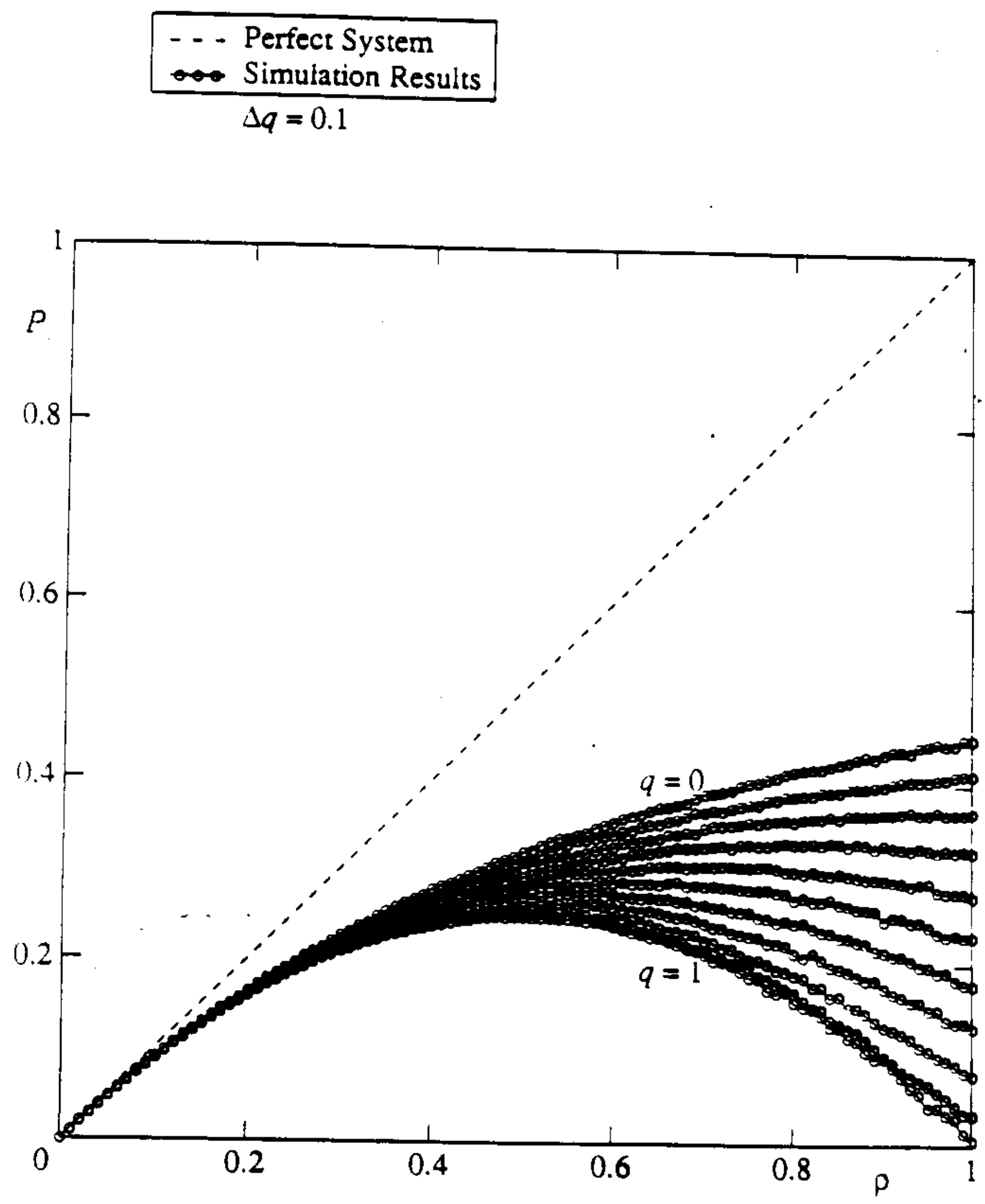


Figure 3.2: Simulation Results for $M/D_qM(SR)$

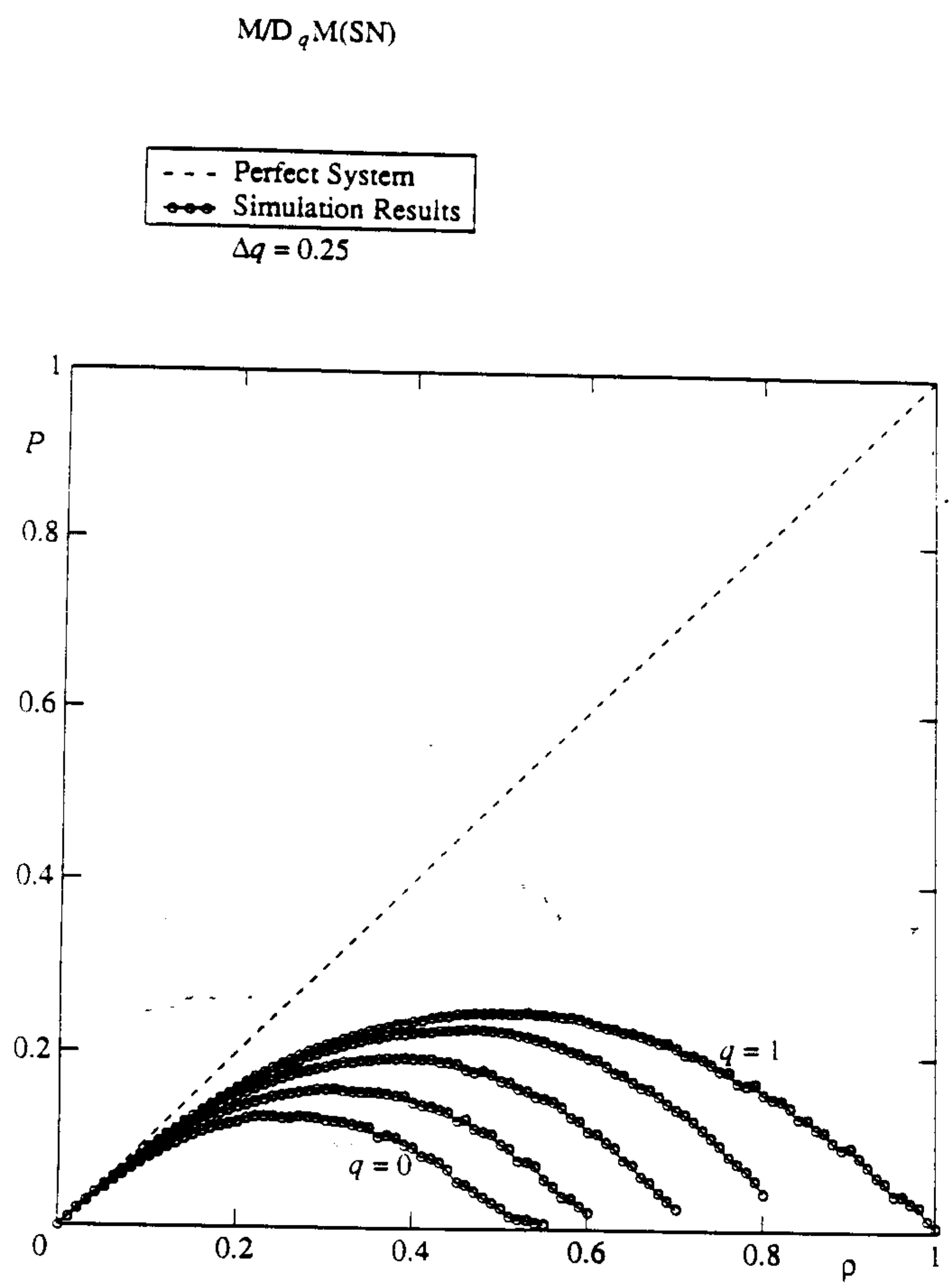


Figure 3.3: Simulation Results for $M/D_qM(SN)$

noredraw systems. This is, again, due to the nature of the service times. Figure 3.2 shows that a redraw of service times will cause shortening, probabilistically speaking, of service times, and, thus, will result in better performance of the redraw systems.

Since in broadcast systems unsuccessful services are terminated even before their prescheduled completion, these systems perform better than silent systems. Broadcast systems have superior performance compared to silent systems.

It is interesting to note that for the M/M(SR) system(s) the normalized power does not drop with an increase in ρ. In fact, as shall be seen in Section 3.4.1, the power approaches a constant as ρ goes to infinity. Such behavior of the system is due to the redraw of service requests upon restart and to the memoryless nature of the service time distribution. Successful service times are shorter than the requested service times, and for high ρ they tend to zero. The queue never becomes unstable for finite ρ. The average system time grows linearly with ρ.

From Figure 3.4 we see that M/M(BR) system gives performance values close to "perfect". In Section 3.4.2 we will see that M/M(BR) indeed gives perfect performance.

Having the simulation results shown in Figure 3.2 through 3.5, and understanding the differencies in the behavior of four types of Poisson winner queues (with a D_qM service time distribution), we analyze some of the systems in the following section.

M/D$_q$M(BR)

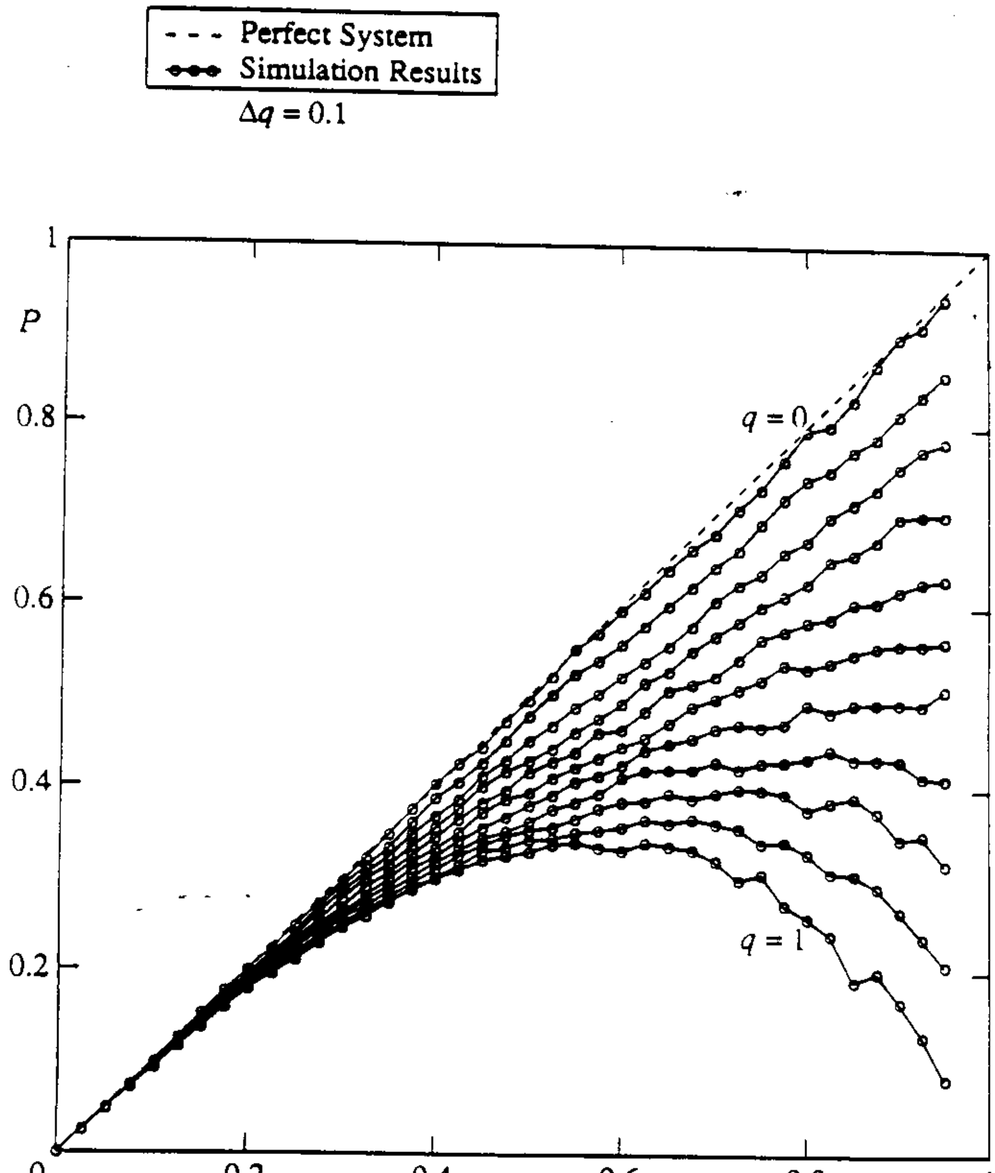


Figure 3.4: Simulation Results for M/D$_q$M(BR)

$M/D_qM(BN)$

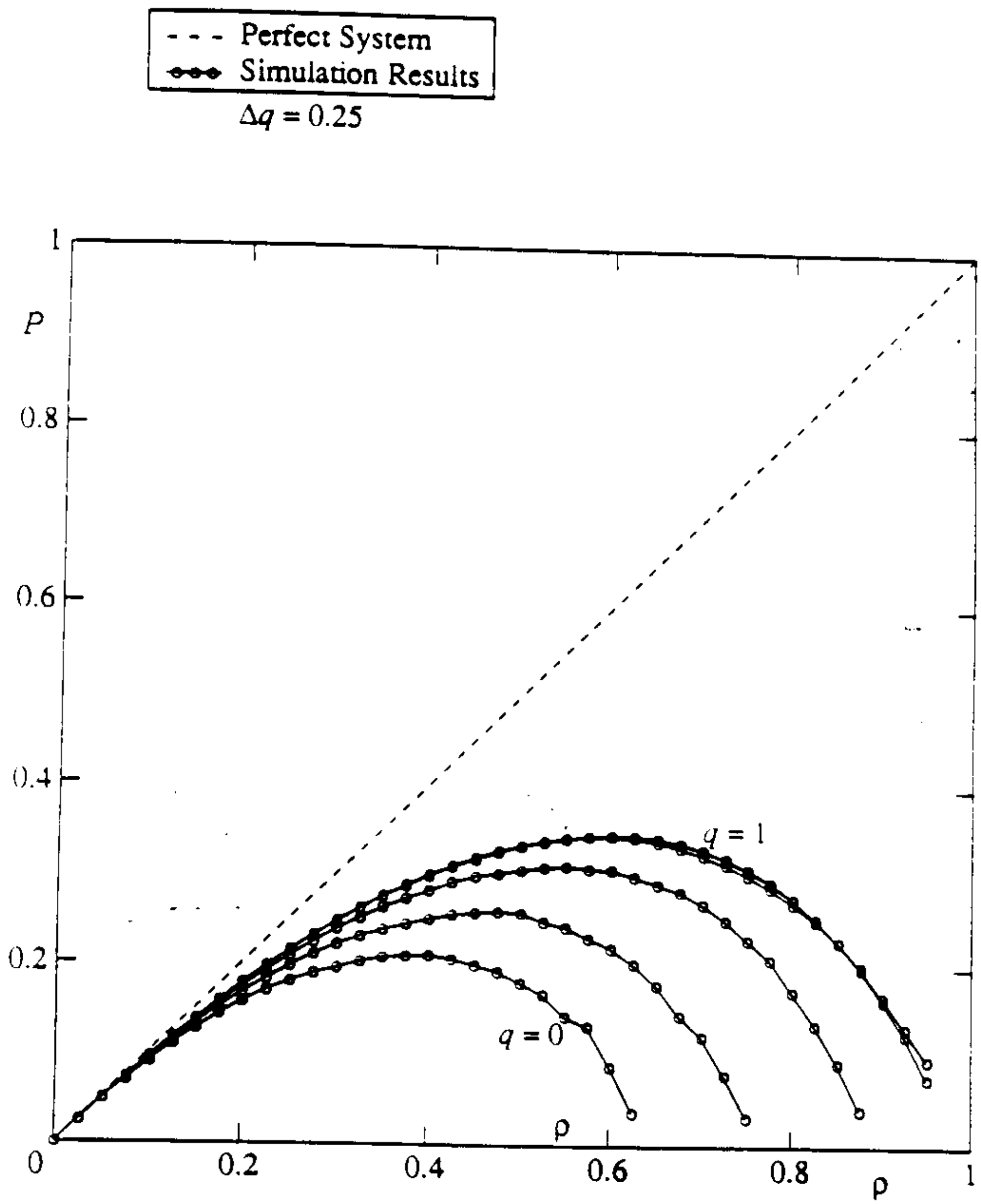


Figure 3.5: Simulation Results for $M/D_qM(BN)$

3.2.3 Analysis

In order to find the normalized power in Poisson winner queues, we use the embedded Markov chain to calculate the distribution of number in system left by departing customers. We define d_k to be the probability of k customers left in the system by a departure, i.e., $d_k \stackrel{\text{def}}{=} P[k$ customers left behind in the system$]$. Once we have calculated all $d_k, k = 1, 2, \ldots$, we can find average time spent by a customer in the system, since the arrivals to the system are memoryless. Thus, we can find the normalized average system time as

$$T_n = \frac{1}{\lambda \overline{x}} \sum_{k=1}^{\infty} k d_k = \frac{N}{\rho} \tag{3.1}$$

where λ is the arrival rate of the customers into the system, and N is the average number of customers in the system. The normalized power P is calculated as

$$P = \frac{\rho}{T_n} = \frac{\rho^2}{N} \tag{3.2}$$

We define states of embedded Markov chain to be the number of customers in the system left by departures. We divide the time axis into intervals between successive departures, as shown in Figure 3.6.

The shaded areas in Figure 3.6 represent the (successful) service of departing customers, and arrows represent the departures from the system. In Table 3.2 we define random variables X, X_r, X_s, V, and V.

In Table 3.3 we further define $p_{i,j}$'s, the transition probabilities between the states.

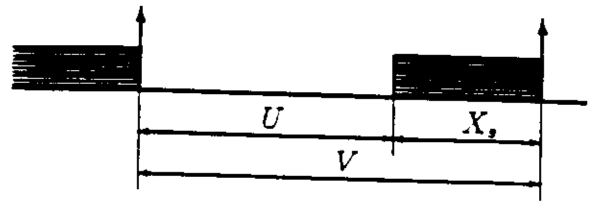


Figure 3.6: Two Successive Departures

Symbol	Definition
X	length of initial service
X_r	length of service upon restart
X_s	length of successful service
V	interdeparture time
U	time between previous departure and the beginning of the next $U = V - X_s$

Table 3.2: Definition of the Basic Random Variables

Symbol	Definition
$p_{i,j}$	P[a departure leaves j customers in the system, given the previous departure left i customers in the system], $i, j = 0, 1, 2, \ldots$

Table 3.3: Definition of $p_{i,j}$

In the following section we analyze the behavior of random variables U, V, X, in order to find the transition probabilities. Having found the transition probabilities, we can calculate the distribution of the number of customers in the system left by departures from the following equation.

$$d_k = \sum_{i=0}^{\infty} d_i p_{i,k}, \quad k = 0, 1, 2, \ldots \tag{3.3}$$

Knowing that $p_{i,k} = 0$, $k < i-1$, as shown in the following section, we get the recursive formula

$$d_k = \frac{1}{p_{k,k-1}}[d_{k-1} - \sum_{i=0}^{k-1} d_i p_{i,k-1}], \quad k = 1, 2, \ldots \tag{3.4}$$

The probability d_0 we calculate from the probability conservation law

$$d_0 = 1 - \sum_{i=1}^{\infty} d_i \tag{3.5}$$

The numerical procedure for finding the normalized response time and power is as follows. For an arbitrary M reasonably large, we calculate all the transition probabilities $p_{i,j}$, $0 \le i \le M$, $0 \le j \le M-2$. To preserve the conservation of the probabilities, we assign the following value to the $p_{i,M-1}$, $0 \le i \le M$.

$$p_{i,M-1} = 1 - \sum_{j=0}^{M-2} p_{i,j}, \quad 0 \le i \le M \tag{3.6}$$

We assign the value 1 to the probability d_0, and from the recursive Formula (3.4) we find all the probabilities d_k, $1 \le k \le M$. Let the sum of all the d_k, $0 \le k \le M$ be C. Now we divide every d_k, $0 \le k \le M$ by C. From the d_k's we find the number of customers in the system N. We shall refer to number M

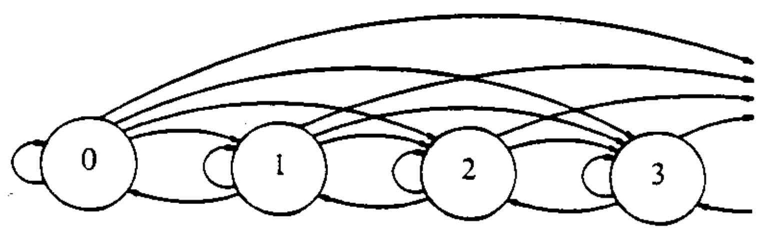


Figure 3.7: Poisson Winner Queue State Transition Diagram

below as the *precision* of the numerical solution. Through Equation (3.1) and (3.2) we use N in obtaining the normalized average response time T_n and normalized power P. The numerical method to go from the transition probabilities to the performance measures is shown later as the top part of Figure 3.10.

In the sections below we concentrate on finding the transition probabilities.

3.2.3.1 Finding the Transition Probabilities

The embedded Markov chain, with the arcs representing transition probabilities, is shown in Figure 3.7.

Since at most one customer may leave between two successive departures, we have

$$p_{i,j} = 0, \quad j < i - 1 \tag{3.7}$$

and thus those transitions are not shown in Figure 3.7

In Table 3.4 we define probability $p_{i,j}(v)$.

Symbol	Definition
$p_{i,j}(v)$	$\frac{1}{dv}$ P[a departure leaves j customers in the system, given the previous departure left i customers in the system, and the corresponding interdeparture time $v < V < v + dv$], $i, j = 0, 1, 2, \ldots, \ v \geq 0$

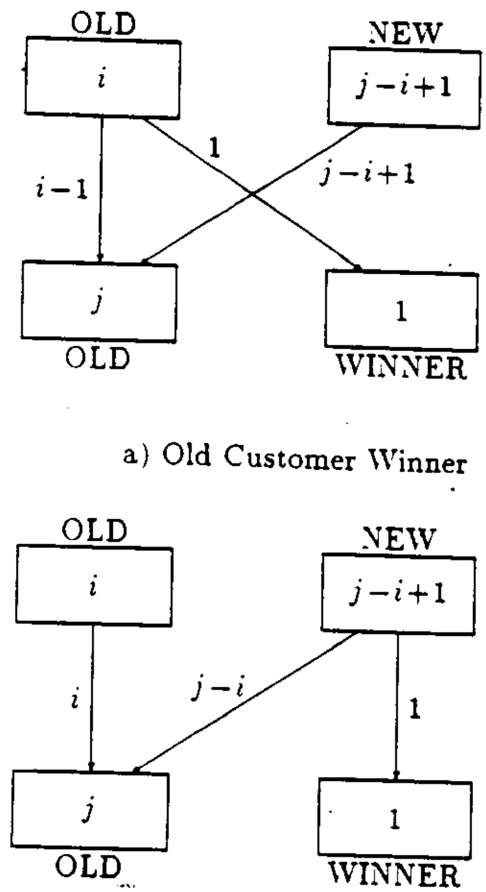
Table 3.4: Definition of $p_{i,j}(v)$

Once we find the probability $p_{i,j}(v)$, we may find $p_{i,j}$ as

$$p_{i,j} = \int_0^\infty p_{i,j}(v)dv \quad . \tag{3.8}$$

Consider any departure D from the system. Reltive to the departure D, we call all the customers left in the system at the departure of D *old customers*, and any customers that arrive after the departure of D *new customers*. The first departure after departure D can be made by either an old customer or by a new customer. Figure 3.8 shows transition graphs for the two cases.

The way the transition graph a) in Figure 3.8 is drawn is as follows. The graph represents a transition from state i to state j. We draw two rows of boxes. The first row is associated with the state i. One of the boxes represents old customers, and we write "i" in it. The other box represents new customers that arrived before the transition to the state j. We leave that box empty for now. The second row of boxes is associated with the state j. In the box that represents old customers we write "j", and in the box that represents a winner we write



a) Old Customer Winner

b) New Customer Winner

Figure 3.8: Poisson Winner Queue Transition Graphs

"1". We now draw arcs from the boxes in the first row to the boxes in the second row. The labels on the arcs represent number of customers that are transferred from one box to another. Since we know that the winner is an old customer (for the case a) we draw an arc labeled "1" from the top old customer box to the winner box. We know that all other old customers remained old, and so we draw an arc labeled "$i-1$" from the top old customer box to the bottom old customer box. Next we know that all other $j-i+1$ old customers at state j must have newly arrived, and so we draw an arc labeled "$j-i+1$" from the new customer box to the bottom old customer box. We have now completed drawing arcs since the sum of the labels on the arcs equals the sum of the bottom row of boxes. Now we take the sum of the labels of all the arcs that leave the new customer box, and we write that number in the box, i.e., we write "$j-i+1$". In the similar way we draw the transition graph for the case b) in Figure 3.8.

Let us consider again a departure D from the system. Let it leave i customers in the system and let the time be equal zero when departure D takes place. In Table 3.5 we define four probabilities, P_{OL}, P_{OW}, P_{NL}, and P_{NW}.

Given the probability that an old customer won, all the new customers lost, and the interdeparture time approaches v, is given by $P_{OW}(i,v)P_{NL}(j-i+1,v)$, as can be clearly observed from the part a) of Figure 3.8. From part b) of the same figure, we see that the probability that a new customer won, all the old customers lost, and the interdeparture time approaches v, is given by $P_{OL}(i,v)P_{NW}(j-i+$

Symbol	Definition
$P_{OL}(i,v)$	P[i old customers finish their next service after time v, given i old customers in the system]
$P_{OW}(i,v)$	$\frac{1}{dv}$ P[$i-1$ old customers finish their next service after time v, and one old customer finishes his next service in the interval $(v, v+dv)$, given i old customers in the system]
$P_{NL}(k,v)$	P[k new customers arrive before time v and they all finish their service after time v]
$P_{NW}(k,v)$	$\frac{1}{dv}$ P[k new customers arrive before time v, $k-1$ of them finish service after time v, and one of them finishes in the interval $(v, v+dv)$]

Table 3.5: Definition of P_{OL}, P_{OW}, P_{NL}, and P_{NW}

1, v). We can now write

$$\begin{aligned} p_{i,j}(v) &= P_{OW}(i,v)P_{NL}(j-i+1,v) + P_{OL}(i,v)P_{NW}(j-i+1,v), \\ & \quad i,j \geq 0 \end{aligned} \tag{3.9}$$

Since we know that

$$P_{OW}(i,v) = i\frac{dP[V_{old} \leq v]}{dv}P^{i-1}[V_{old} > v] = -i\frac{d}{dv}P^{i}[V_{old} > v] = -\frac{d}{dv}P_{OL}(i,v) \tag{3.10}$$

where V_{old} is as defined in the next section, we only need to find probabilities $P_{OL}(i,v)$, $P_{NL}(k,v)$, and $P_{NW}(k,v)$ in order to find the $p_{i,j}$'s. This process is as the middle part of Figure 3.10.

3.2.3.2 Finding the $P_{OL}(i,v)$ and $P_{OW}(i,v)$

Consider an old customer left in the system by a departure at time zero. We define U_{old} to be a random variable representing the time until the end of his present unsuccessful service, and $V_{old} = U_{old} + X_r$ to be a random variable representing the time until the end of his restarted service. Let $\mathcal{U}_{old}(v)$ represent the probability distribution function of the random variable U_{old}. Let $\mathcal{V}_{old}(v)$ represent probability distribution function of the random variable V_{old}. The following holds.[1]

$$\begin{aligned} P[V_{old} \leq v] &\stackrel{\text{def}}{=} \mathcal{V}_{old}(v) \\ &= \mathcal{U}_{old}(v) \otimes b_r(v) \end{aligned} \tag{3.11}$$

[1] $\otimes$ represents convolution.

$$= \int_0^v \mathcal{U}_{old}(u) b_r(v-u)\, dv$$

$$P_{OL}(i,v) = (P[V_{old} > v])^i = [1 - \mathcal{V}_{old}(v)]^i \tag{3.12}$$

The restarted service time PDF $B_r(x)$ is given as a parameter of the system. For the simple winner queues it is also the distribution of the initial service times. We now only need to find $\mathcal{U}_{old}(u)$ in order to find $P_{OL}(i,v)$. For all broadcast systems $U_{old} = 0$, and thus we have

$$\mathcal{U}_{old}(u) = 1, \quad u \geq 0$$

which gives us, in the broadcast case

$$P_{OL}(i,v) = (P[X_r > v])^i = [1 - B_r(v)]^i \tag{3.13}$$

For the broadcast redraw system with a D_qM service time distribution defined in Equation (2.1), we have:

$$P_{OL}(i,v) = \begin{cases} 1, & v \leq q\overline{x} \\ e^{-i(\mu v - q)/p}, & v > q\overline{x} \end{cases} \tag{3.14}$$

where μ is defined as $\mu \stackrel{\text{def}}{=} 1/\overline{x}$ and $p = 1 - q$. Using Equation (3.10) we get

$$P_{OW}(i,v) = \begin{cases} 0, & v \leq q\overline{x} \\ i\frac{\mu}{p} e^{-i(\mu v - q)/p}, & v > q\overline{x} \end{cases} \tag{3.15}$$

For the silent redraw systems with memoryless restarted (and initial) service times we have

$$\mathcal{U}_{old}(u) = B_r(u) \tag{3.16}$$

since for the systems at hand $b_r(x) \equiv b(x)$. Using Equation (2.1) and (3.11) we get

$$\mathcal{V}_{old}(v) = \int_0^v \mu e^{-\mu u}[1-e^{-\mu(v-u)}]du = (1+\mu v)e^{-\mu v}$$

and thus, using Equation (3.12)

$$P_{OL}(i,v) = (1+\mu v)^i e^{-i\mu v}, \quad v \geq 0 \tag{3.17}$$

and from Equation (3.10)

$$P_{OW}(i,v) = i\mu^2 v(1+\mu v)^{i-1}e^{-i\mu v}, \quad v \geq 0 \tag{3.18}$$

For the silent winner queues with deterministic restarted (and initial) service times we will make an approximation by assuming that the arrivals of old customers are memoryless within the time interval $[0,\overline{x}]$, i.e., they are exponentially distributed but also forced to arrive in $[0,\overline{x}]$. This gives us the following approximate expression for $\mathcal{U}_{old}(u)$.

$$\mathcal{U}_{old}(u) = \begin{cases} \dfrac{1-e^{-\mu u}}{1-1/e}, & 0 \leq u \leq \overline{x} \\ 1, & u > \overline{x} \end{cases} \tag{3.19}$$

For deterministic service times we have

$$B_r(x) = \begin{cases} 0, & x \leq \overline{x} \\ 1, & x > \overline{x} \end{cases}$$

and thus, from Equation (3.11) and (3.12) we have

$$P_{OL}(i,v) = \begin{cases} 1, & 0 \leq v \leq \overline{x} \text{ or } i = 0 \\ \left(\dfrac{e^{1-\mu v}-1/e}{1-1/e}\right)^i, & \overline{x} < v \leq 2\overline{x}, i \geq 1 \\ 0, & v > 2\overline{x}, i \geq 1 \end{cases} \tag{3.20}$$

From the last equation and Equation (3.10) and we get

$$P_{OW}(i,v) = \begin{cases} \frac{i\mu e^{1-\mu v}}{1-1/e}\left(\frac{e^{1-\mu v}-1/e}{1-1/e}\right)^{i-1}, & \overline{x} < v \le 2\overline{x} \\ 0, & \text{otherwise} \end{cases} \tag{3.21}$$

In this section we found the exact values for the probabilities $P_{OL}(i,v)$ and $P_{OW}(i,v)$ for M/D$_q$M(BR) and M/M(SR) systems, and approximations for system M/D(S). We point out that for the M/D$_q$M(SR), $q > 0$, systems we cannot yet find the probability distribution of the random variable U_{old}, and so, we are unable to find $P_{OL}(i,v)$. For all noredraw systems, except for the approximation M/D(S), not only do we not have the distribution of U_{old}, but we also do not have the distribution of random variable X_r since, in general, it differs from the given (initial) service time distribution, i.e., $b_r(x) \neq B_r(x)$.

3.2.3.3 Finding the $P_{NL}(k,v)$

Figure 3.9 shows the time axis with k new customers arriving in the interval $(0,v)$. Interarrival times of the customers are: $v-y_1, y_1-y_2, \ldots, y_{k-1}-y_k$. Referring to the definition of $P_{NL}(k,v)$ in Table 3.5, and defining $\beta(x) \stackrel{\text{def}}{=} P[X > x] = 1-B(x)$, we can write

$$\begin{aligned} P_{NL}(k,v) &= \int_0^v \lambda e^{-\lambda(v-y_1)}\beta(y_1)\int_0^{y_1}\lambda e^{-\lambda(y_1-y_2)}\beta(y_2)\int_0^{y_2}\cdots \\ &\quad \int_0^{y_{k-1}}\lambda e^{-\lambda(y_{k-1}-y_k)}\beta(y_k)e^{-\lambda y_k}dy_k\cdots dy_2dy_1 \\ &= \lambda^k e^{-\lambda v}\int_0^v \beta(y_1)\int_0^{y_1}\beta(y_2)\int_0^{y_2}\cdots\int_0^{y_{k-1}}\beta(y_k)dy_k\cdots dy_2dy_1 \end{aligned}$$

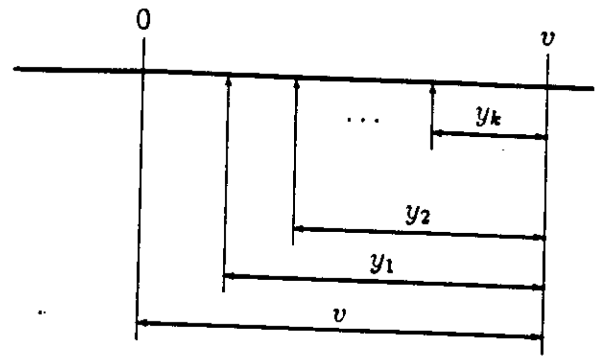


Figure 3.9: New Arrivals

If we define

$$\gamma(v) \stackrel{\text{def}}{=} \int_0^v \beta(z)\,dz = \int_0^v [1-B(z)]\,dz. \tag{3.22}$$

then we have

$$P_{NL}(k,v) = \lambda^k e^{-\lambda v} \int_0^v \int_0^{y_1} \int_0^{y_2} \cdots \int_0^{y_{k-1}} d\gamma(y_k) \cdots d\gamma(y_2) d\gamma(y_1)$$

and finally

$$P_{NL}(k,v) = \frac{[\lambda\gamma(v)]^k}{k!} e^{-\lambda v} \tag{3.23}$$

For a D_qM service time probability distribution we have

$$\beta(x) = \begin{cases} 1, & x \le q\overline{x} \\ e^{-(\mu x - q)/p}, & x > q\overline{x} \end{cases} \tag{3.24}$$

$$\gamma(x) = \begin{cases} x, & x \le q\overline{x} \\ \frac{q}{\mu} + \frac{p}{\mu}[1 - e^{-(\mu x - q)/p}], & x > q\overline{x} \end{cases} \tag{3.25}$$

$$P_{NL}(k,v) = \begin{cases} \frac{(\lambda v)^k}{k!} e^{-\lambda v}, & v \le q\overline{x} \\ \frac{\rho^k [1 - p e^{-(\mu v - q)/p}]^k}{k!} e^{-\lambda v}, & v > q\overline{x} \end{cases} \tag{3.26}$$

where $\rho \stackrel{\text{def}}{=} \lambda\overline{x} = \lambda/\mu$. For $q = 0$ (memoryless service times) Equation (3.26) converts to

$$P_{NL}(k,v) = \frac{\rho^k(1-e^{-\mu v})^k}{k!}e^{-\lambda v}, \quad v \geq 0 \tag{3.27}$$

and for $q = 1$ (deterministic service times) Equation (3.26) converts to

$$P_{NL}(k,v) = \begin{cases} \frac{(\lambda v)^k}{k!}e^{-\lambda v}, & v \leq \overline{x} \\ \frac{\rho^k}{k!}e^{-\lambda v}, & v > \overline{x} \end{cases} \tag{3.28}$$

In this section we found the probability $P_{NL}(k,v)$ for the general Poisson winner queue M/G(CRS), and gave expressions for the special case, queue M/D_qM-(CRS).

3.2.3.4 Finding the $P_{NW}(k,v)$

We define probability $P_{NW,i}(k,v), i = 1,2,\ldots,k$ the same as $P_{NW}(k,v)$ in Table 3.5, with the restriction that the i-th customer is the one that wins. We can now write

$$\begin{aligned} P_{NW,i}(k,v) &= \lambda^k e^{-\lambda v}\int_0^v \beta(y_1)\int_0^{y_1}\beta(y_2)\int_0^{y_2}\cdots \\ &\int_0^{y_{i-1}} b(y_i)\int_0^{y_i}\cdots\int_0^{y_{k-1}}\beta(y_k)dy_k\cdots dy_2dy_1 \end{aligned} \tag{3.29}$$

where $b(x)$ is the probability density of the service times. We find $P_{NW}(k,v)$ as

$$P_{NW}(k,v) = \sum_{i=1}^{k} P_{NW,i}(k,v) \tag{3.30}$$

Equations (3.29) and (3.30) are as far as we can go for a general Poisson queue. For the M/D_qM(CRS) system we derive $P_{NW}(k,v)$ as follows. From

Equation (2.1) we get

$$b(x) = \begin{cases} 0, & x \leq \overline{x} \\ \frac{\mu}{p} e^{-(\mu x - q)}, & x > q\overline{x} \end{cases} \tag{3.31}$$

From Equation (3.24) and (3.31) we have the following relation between $b(x)$ and $\beta(x)$:

$$b(x) = \begin{cases} 0, & x \leq q\overline{x} \\ \frac{\mu}{p}\beta(x), & x > q\overline{x} \end{cases} \tag{3.32}$$

From Equation (3.29) and (3.32) we see that $P_{NW,i}(k,v) = 0$ for $0 \leq v \leq q\overline{x}$. For $v > \overline{x}$ we have

$$\begin{aligned} P_{NW,i}(k,v) &= \lambda^k e^{-\lambda v} \frac{\mu}{p} \int_{q\overline{x}}^{v} \beta(y_1) \int_{q\overline{x}}^{y_1} \beta(y_2) \int_{q\overline{x}}^{y_2} \cdots \int_{q\overline{x}}^{y_{i-1}} \beta(y_i) \\ &\quad \int_0^{y_i} \beta(y_{i+1}) \int_0^{y_{i+1}} \cdots \int_0^{y_{k-1}} \beta(y_k) dy_k \cdots dy_2 dy_1, \quad v > q\overline{x} \\ P_{NW,i}(k,v) &= \lambda^k e^{-\lambda v} \frac{\mu}{p} \int_{q\overline{x}}^{v} \beta(y_1) \int_{q\overline{x}}^{y_1} \beta(y_2) \int_{q\overline{x}}^{y_2} \cdots \int_{q\overline{x}}^{y_{i-1}} \beta(y_i) \\ &\quad \frac{\gamma^{k-i-1}(y_i)}{(k-i-1)!} dy_i \cdots dy_2 dy_1, \quad v > q\overline{x} \end{aligned}$$

We now define

$$\beta_0(x) \stackrel{\text{def}}{=} \begin{cases} 0, & x \leq q\overline{x} \\ \beta(x), & x > q\overline{x} \end{cases}$$

$$\gamma_0(x) \stackrel{\text{def}}{=} \int_0^x \beta_0(z)\, dz$$

The following holds

$$\gamma(x) = \int_0^{q\overline{x}} \beta(z)\, dz + \gamma_0(x) = \gamma(q\overline{x}) + \gamma_0(x)$$

and from Equation (3.22) we get

$$\gamma(x) = q\overline{x} + \gamma_0(x) \tag{3.33}$$

$P_{NW,i}(k,v)$ becomes now, for $v > q\overline{x}$

$$\begin{aligned} P_{NW,i}(k,v) &= \lambda^k e^{-\lambda v}\frac{\mu}{p}\int_0^v \int_0^{y_1}\int_0^{y_2}\cdots\int_0^{y_{i-1}} \frac{[q\overline{x}+\gamma_0(y_i)]^{k-i-1}}{(k-i-1)!} \\ & \quad d\gamma_0(y_i)\cdots d\gamma_0(y_1), \quad v > q\overline{x} \end{aligned}$$

which gives us

$$P_{NW,i}(k,v) = \frac{\mu}{p}\frac{\lambda^k}{k!}e^{-\lambda v}\sum_{n=i}^{k}\binom{k}{n}(q\overline{x})^{k-n}\gamma_0^n(v), \quad v > q\overline{x}$$

Using Equation (3.30) we find $P_{NW}(k,v)$ as follows.

$$\begin{aligned} P_{NW}(k,v) &= \sum_{i=1}^{k}\frac{\mu}{p}\frac{\lambda^k}{k!}e^{-\lambda v}\sum_{n=i}^{k}\binom{k}{n}(q\overline{x})^{k-n}\gamma_0^n(v) \\ &= \frac{\mu}{p}\frac{\lambda^k}{k!}e^{-\lambda v}\sum_{n=0}^{k} n\binom{k}{n}(q\overline{x})^{k-n}\gamma_0^n(v) \\ &= \frac{\mu}{p}\frac{\lambda^k}{k!}e^{-\lambda v}\gamma_0(v)\frac{d}{d\gamma_0(v)}[q\overline{x}+\gamma_0(v)]^k, \quad v > q\overline{x} \end{aligned}$$

And finally,

$$P_{NW}(k,v) = \frac{\mu}{p}\lambda^k\gamma_0(v)\frac{\gamma^{k-1}(v)}{(k-1)!}e^{-\lambda v}, \quad v > q\overline{x} \tag{3.34}$$

After including the expressions for $\gamma(v)$ and $\gamma_0(v)$ from Equation (3.25) and (3.33), we get

$$P_{NW}(k,v) = \begin{cases} \lambda[1-e^{-(\mu v-q)/p}]\dfrac{\rho^{k-1}[1-pe^{-(\mu v-q)/p}]^{k-1}}{(k-1)!}e^{-\lambda v}, & v > q\overline{x}, k \geq 1 \\ 0, & \text{otherwise} \end{cases} \tag{3.35}$$

For memoryless service times, $q = 0$, Equation (3.35) becomes

$$P_{NW}(k,v) = \begin{cases} \lambda \frac{\rho^{k-1}(1-e^{-\mu v})^k}{(k-1)!} e^{-\lambda v}, & v \geq 0, k \geq 1 \\ 0, & \text{otherwise} \end{cases} \tag{3.36}$$

For deterministic service times, $q = 1$, Equation (3.35) becomes

$$P_{NW}(k,v) = \begin{cases} \lambda \frac{\rho^{k-1}}{(k-1)!} e^{-\lambda v}, & v > \overline{x}, k \geq 1 \\ 0, & \text{otherwise} \end{cases} \tag{3.37}$$

In this section we found the probability $P_{NW}(k,v)$ for M/D$_q$M(CRS). It is possible to find $P_{NW}(k,v)$ for other service time probability distributions starting from equations (3.29) and (3.30), and using a technique similar to the one used here for M/D$_q$M(CRS).

3.2.3.5 Derivation of Results in Short

The summary of notation used in the analysis is given in Table 3.6.

Figure 3.10 shows the numerical process of obtaining the performance values T_n and P, starting with given densities $b(x)$ and $b_R(x)$, which are identical for the systems at hand, and ending with T_n and P at the top of the *derivation graph*.

Function	Defined in
$b(x)$	Section 2.1
$b_R(x)$	Section 2.1
$\mathcal{U}_{old}(u)$	Section 3.2.3.2
$\mathcal{V}_{old}(u)$	Section 3.2.3.2
$\gamma(x)$	Equation (3.22)
$\gamma_0(x)$	Equation (3.25)
P_{OL}	Table 3.5
P_{OW}	Table 3.5
P_{NL}	Table 3.5
P_{NW}	Table 3.5
$p_{i,j}(v)$	Table 3.4
$p_{i,j}$	Table 3.3
d_k	Section 3.2.3
T_n	Section 3.2.3
P	Section 3.2.3

Table 3.6: Notation for Poisson Winner Queues

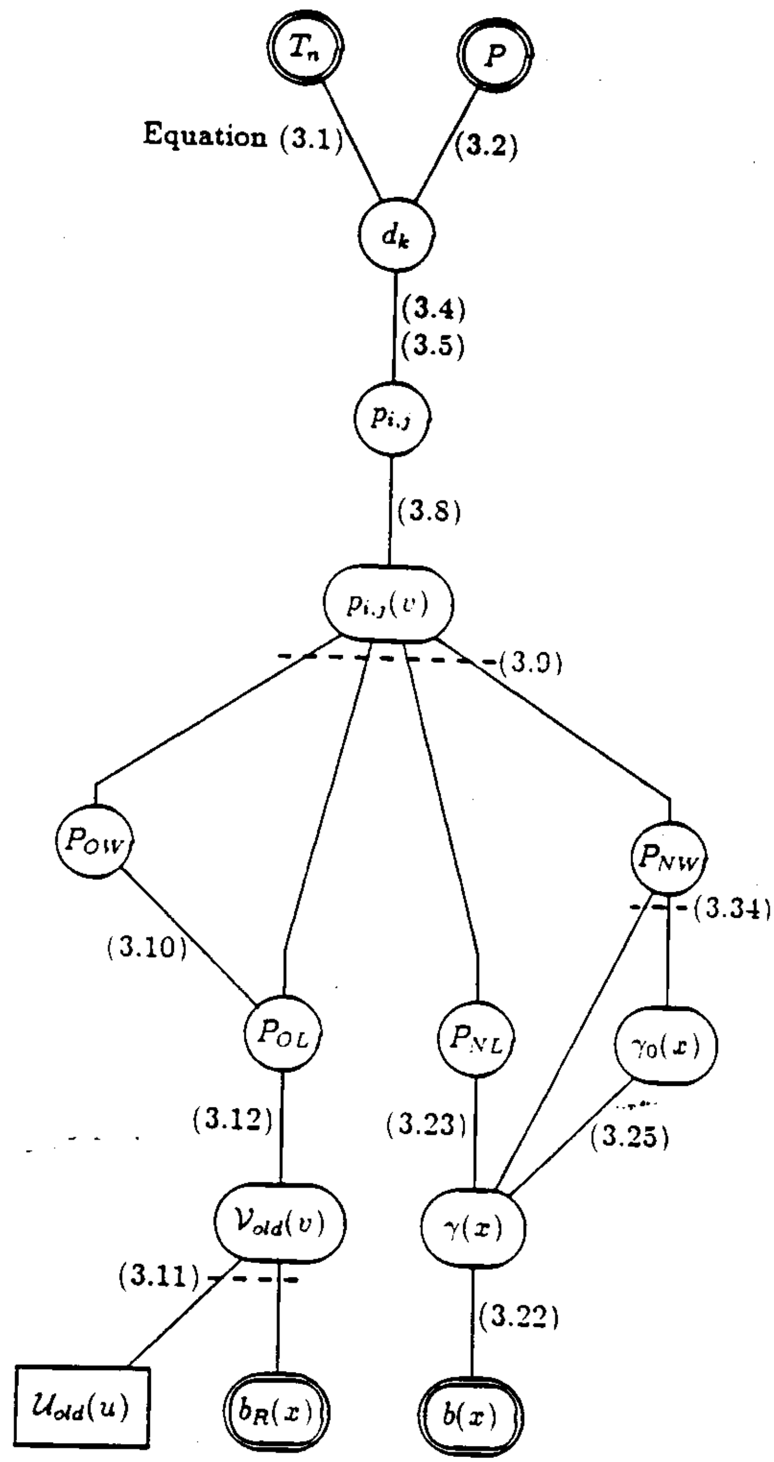


Figure 3.10: Derivation Graph for Simple Winner Queues

3.2.4 System S1: M/M(SR) with Exact $p_{i,j}$'s

By substituting Formula (3.17), (3.18), (3.27), and (3.36) into (3.9), we get the following expressions for an M/M(SR) system.

$$p_{i,j}(v) = \begin{cases} (j+1)\mu \dfrac{[\rho(1-e^{-\mu v})]^{j+1}}{(j+1)!} e^{-\lambda v}, & i=0,\ j \geq 0,\ v \geq 0 \\ \mu[(j+1)\mu v + j - i + 1](1+\mu v)^{i-1} & \\ \cdot \dfrac{[\rho(1-e^{-\mu v})]^{j-i+1}}{(j-i+1)!} e^{-(\lambda+i\mu)v}, & i \geq 1,\ j \geq i-1,\ v \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{3.38}$$

which after integration according to Equation (3.8) gives

$$p_{i,j} = \begin{cases} \rho^{j+1} \sum_{m=0}^{j+1} \dfrac{(-1)^m}{m!\,(j+1-m)!} \dfrac{j+1}{\rho+m}, & i=0,\ j \geq 0 \\ \rho^{j-i+1} \sum_{m=0}^{j-i+1} \dfrac{(-1)^m}{m!\,(j+1-m)!} & \\ \sum_{k=0}^{i-1} \dfrac{(i-1)!}{(i-1-k)!} \dfrac{1}{(\rho+i+m)^{k+2}} & \\ \cdot[(j+1)(k+1)+(\rho+i+m)(j-i+1)], & i \geq 1,\ j \geq i-1 \\ 0, & \text{otherwise} \end{cases} \tag{3.39}$$

Following the process of numerical calculation depicted in Figure 3.10, we calculate the normalized power P and normalized response time T_n versus load ρ in Figure 3.11 and 3.12. In the same figures we also show the simulation results for M/M(SR) given previously as one of the curves in Figure 3.2.

For high ρ, power for M/M(SR) seems to be approaching a constant. Figure 3.13 shows numerical calculations for different values of M, and the dotted curve represents the approximate solution. The explanation is that successful service times approach zero for high ρ, and average service time grows linearly

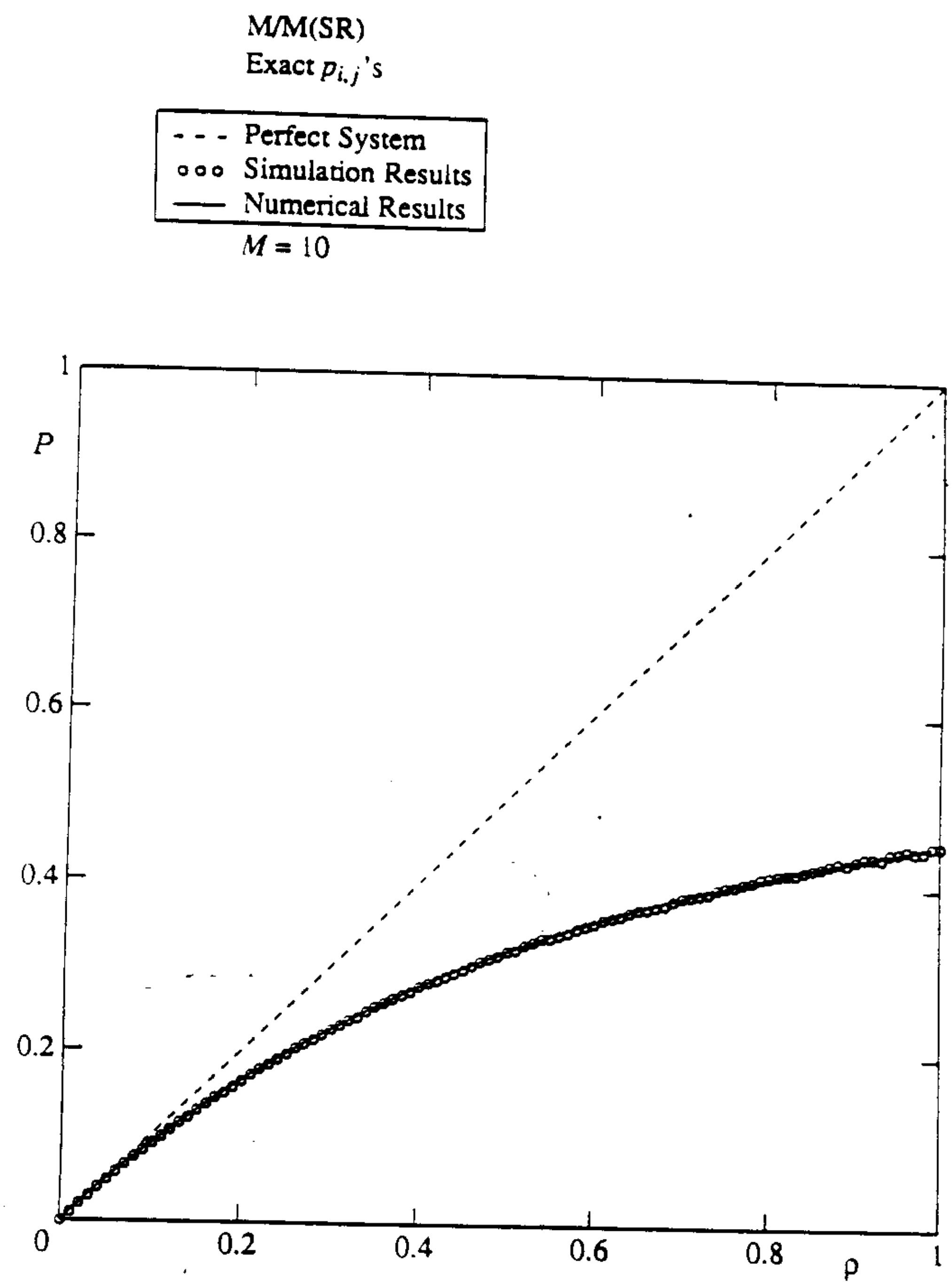


Figure 3.11: Normalized Power for M/M(SR)

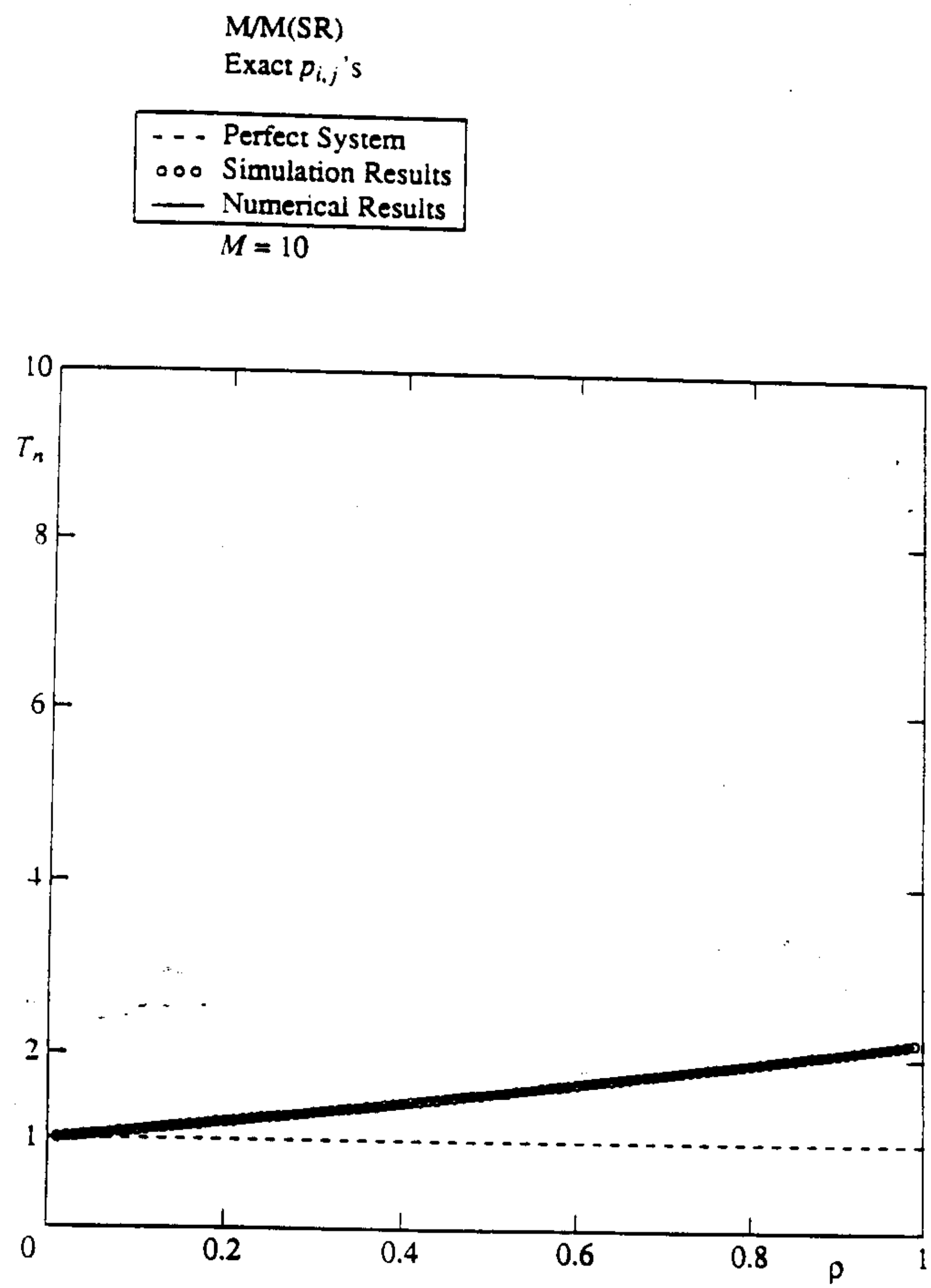


Figure 3.12: Normalized Average Response Time for M/M(SR)

with ρ.

A two dimensional Markov chain that models M/M(SR) system is given in [13] and described in Section 3.4.1. The results shown in Figure 3.13 are in agreement with results obtained by simulation in [14].

3.2.5 System S2: M/D(S) with Approximate $p_{i,j}$'s

By substituting Formula (3.20), (3.21), (3.28), and (3.37) into (3.9), we get the following expressions for the M/D(S) system.

$$p_{i,j}(v) = \begin{cases} \lambda \frac{\rho^{k-1}}{(k-1)!} e^{\lambda v}, & i = 0,\ j \geq 0,\ v \geq \overline{x} \\ \mu[(j+1)e^{1-\mu v} - (j-i+1)/e] & \\ \cdot \frac{(e^{1-\mu v} - 1/e)^{i-1}}{(1-1/e)^{i}} \frac{\rho^{j-i+1}}{(j-i+1)!} e^{\lambda v}, & i \geq 1,\ j \geq i-1,\ \overline{x} < v \leq \overline{x} \\ 0, & \text{otherwise} \end{cases} \tag{3.40}$$

which after integration according to Equation (3.8) gives

$$p_{i,j} = \begin{cases} \frac{\rho^{j}}{j!} e^{-\rho}, & i = 0,\ j \geq 0 \\ \frac{\rho^{j-i+1}}{(j-i+1)!} \frac{e^{-(\rho+1)}}{(1-1/e)^{i}} \sum_{k=0}^{i-1} \binom{i-1}{k} (-1/e)^{i-1-k} & \\ \cdot \left\{ [e - e^{-(\rho+k)}] \frac{j+1}{\rho+k+1} - [1 - e^{-(\rho+k)}] \frac{j-i+1}{\rho+k} \right\}, & i \geq 1,\ j \geq i-1 \\ 0, & \text{otherwise} \end{cases} \tag{3.41}$$

Following the process of numerical calculation depicted in Figure 3.10, we calculate the normalized power P and normalized response time T_n versus load ρ in Figure 3.14 and 3.15. In the same figures we also show the simulation results for M/M(SR) given previously as one of the curves in Figure 3.2 and 3.3

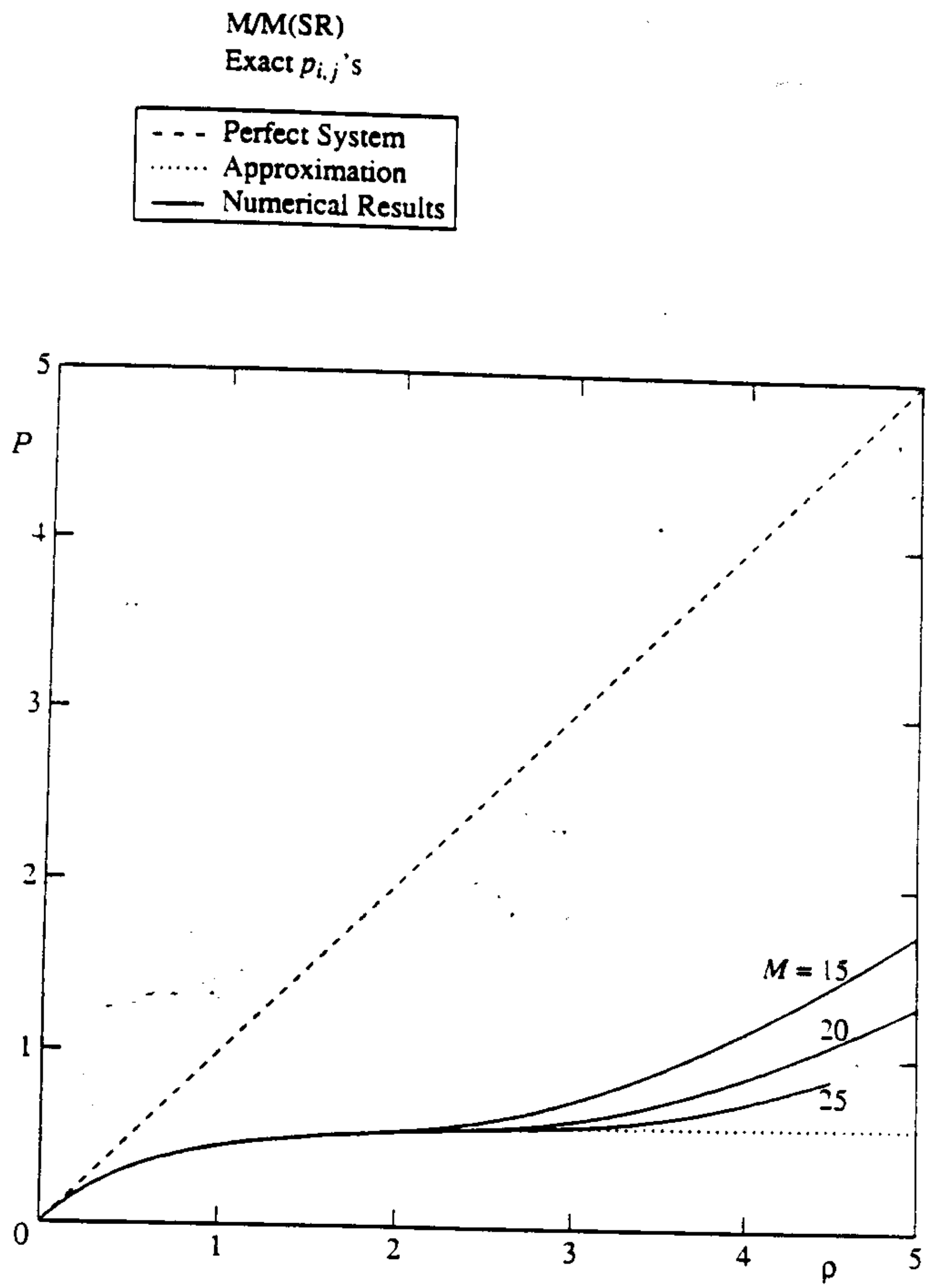


Figure 3.13: Normalized Power for M/M(SR) with High ρ

(M/D(SR) is equivalent to the M/D(SN) due to deterministic service times).

Note the two tails of the power plotted in Figure 3.14. The lower tail is the power calculated with higher precision, i.e., M is higher. This is due to errors caused by numerical calculations. The exact power for the M/D(S) should drop to zero for $\rho = 1$. It is interesting that the shape of the normalized power for M/D(S), as Figure 3.14 shows, looks very much like the normalized power for the queue M/M/1, given as $P_{M/M/1} = \rho(1-\rho)$! At this point we do not know if it actually *is the same* as for the M/M/1.

3.2.6 System S3: M/D$_q$M(BR) with Exact $p_{i,j}$'s

By substituting Formula (3.14), (3.15), (3.26), and (3.35) into (3.9), we get the following expressions for an M/D$_q$M(BR) system.

$$p_{i,j}(v) = \begin{cases} \lambda[1-e^{-(\mu v-q)/p}]\dfrac{\rho^j[1-pe^{-(\mu v-q)/p}]^j}{j!}e^{-\lambda v}, & v \geq q\overline{x},\ i=0,\ j\geq 0 \\ e^{-i(\mu v-q)/p}\dfrac{\lambda}{j-i}\dfrac{\rho^{j-i}[1-pe^{-(\mu v-q)/p}]^{j-i}}{(j-i)!} & \\ \cdot\{iq/p+(j+1)[1-e^{-(\mu v-q)/p}]\}e^{-\lambda v}, & v\geq \overline{x},\ i\geq 1,\ j\geq i-1 \\ 0, & \text{otherwise} \end{cases} \tag{3.42}$$

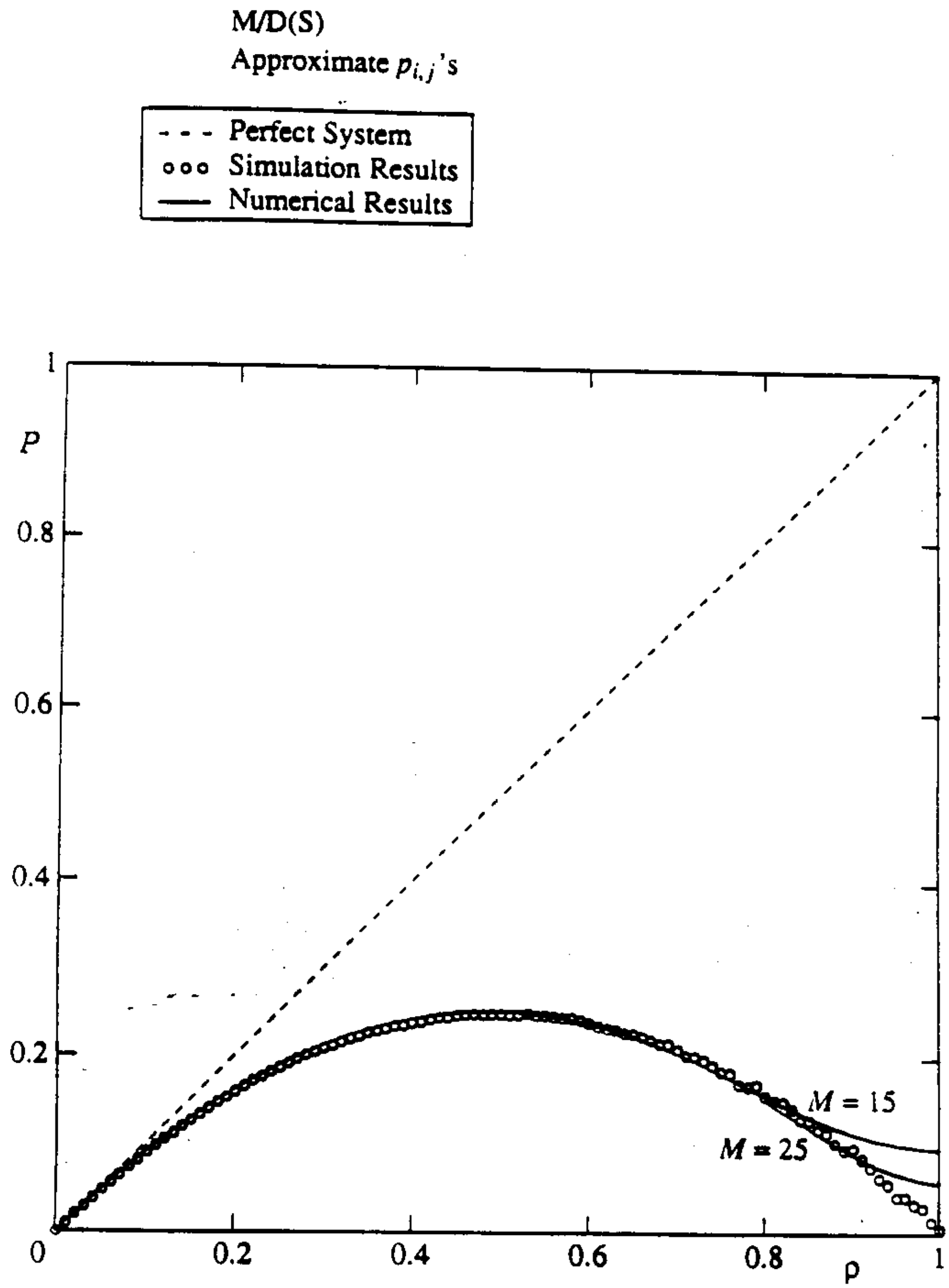


Figure 3.14: Normalized Power for M/D(S)

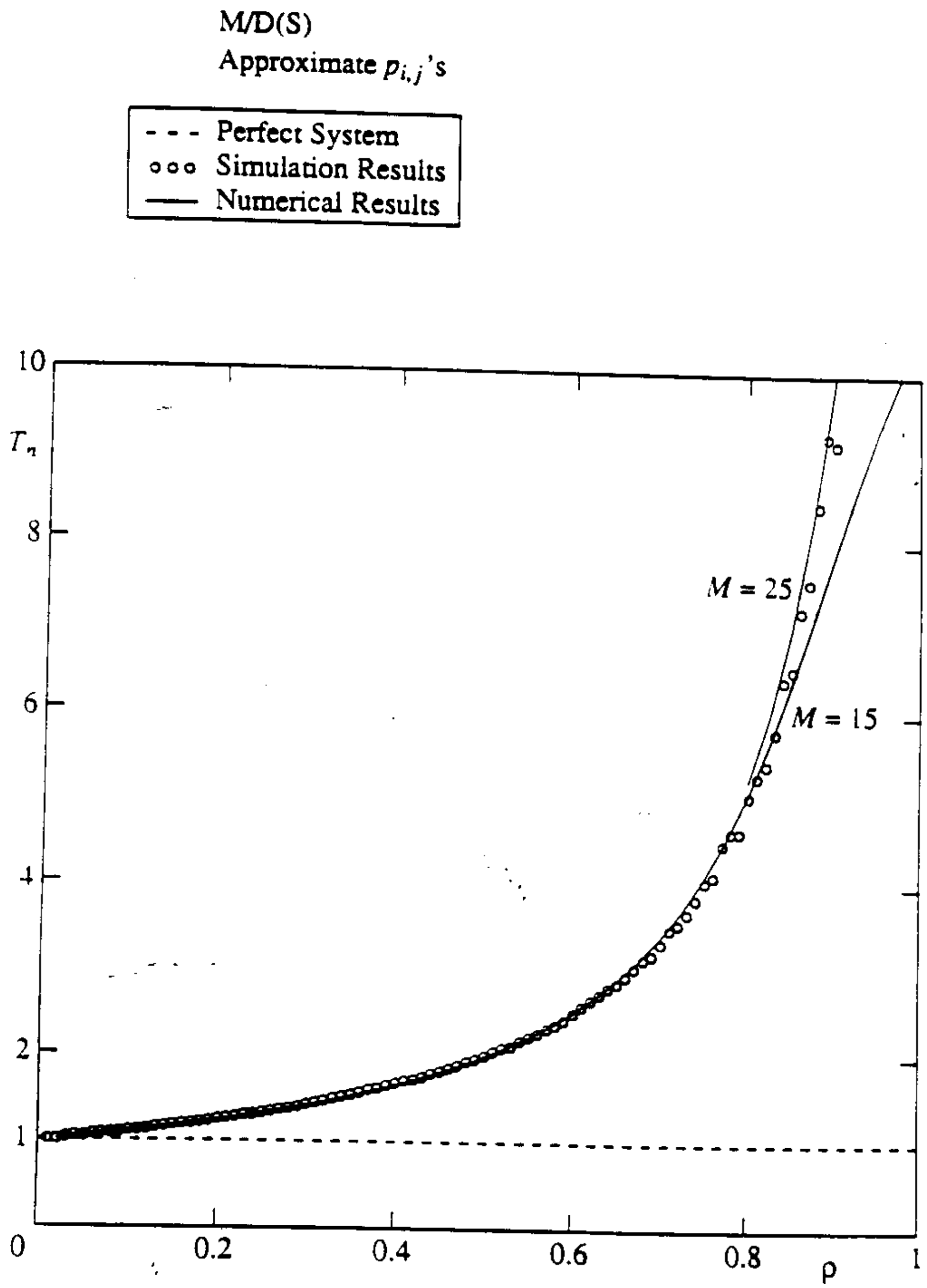


Figure 3.15: Normalized Average Response Time for M/D(S)

which after integration according to Equation (3.8) gives

$$p_{i,j} = \begin{cases} \rho^{j+1} e^{-q\rho} \sum_{m=0}^{j} \frac{(-p)^m}{m!\,(j-m)!} \left[\frac{1}{\rho+m/p} - \frac{1}{\rho+(m+1)/p} \right], & i=0,\ j \geq 0 \\ \frac{\rho^{j-i+1}}{j-i+1} e^{-q\rho} \sum_{m=0}^{j-i} \frac{(-p)^m}{m!\,(j-i-m)!} \\ \cdot \left[\frac{j+1+iq/p}{\rho+(i+m)/p} - \frac{j+1}{\rho+(i+m+1)/p} \right], & i \geq 1,\ j \geq i-1 \\ 0, & \text{otherwise} \end{cases} \tag{3.43}$$

Following the process of numerical calculation depicted in Figure 3.10. we calculate the normalized power P and normalized response time T_n versus load ρ in Figure 3.16 and 3.17. In the same figures we also show the simulation results for M/D$_q$M(BR) given previously as one of the curves in Figure 3.4.

Figure 3.16 shows that M/M(BR) system gives a "perfect" performance. Also, for $q=1$ we get the same normalized power as for an ordinary M/D/1 queue. We will discuss those cases later in Section 3.4

Figure 3.18 and 3.19 plot the normalized power and the average response system time, respectively, for M/D$_q$M(BR) in three dimensions. The dimensions are ρ, q, and P (T_n).

3.2.7 Summary

In Sections 3.2.4 through 3.2.6 we have obtained results for the winner queues M/M(SR), M/D(S), and M/D$_q$M(BR), respectively. The queues M/M(SR), M/M(BR), and M/D(BR) will be considered again in Section 3.4. Figure 3.20 shows graphically what systems are solved through the analysis of the simple

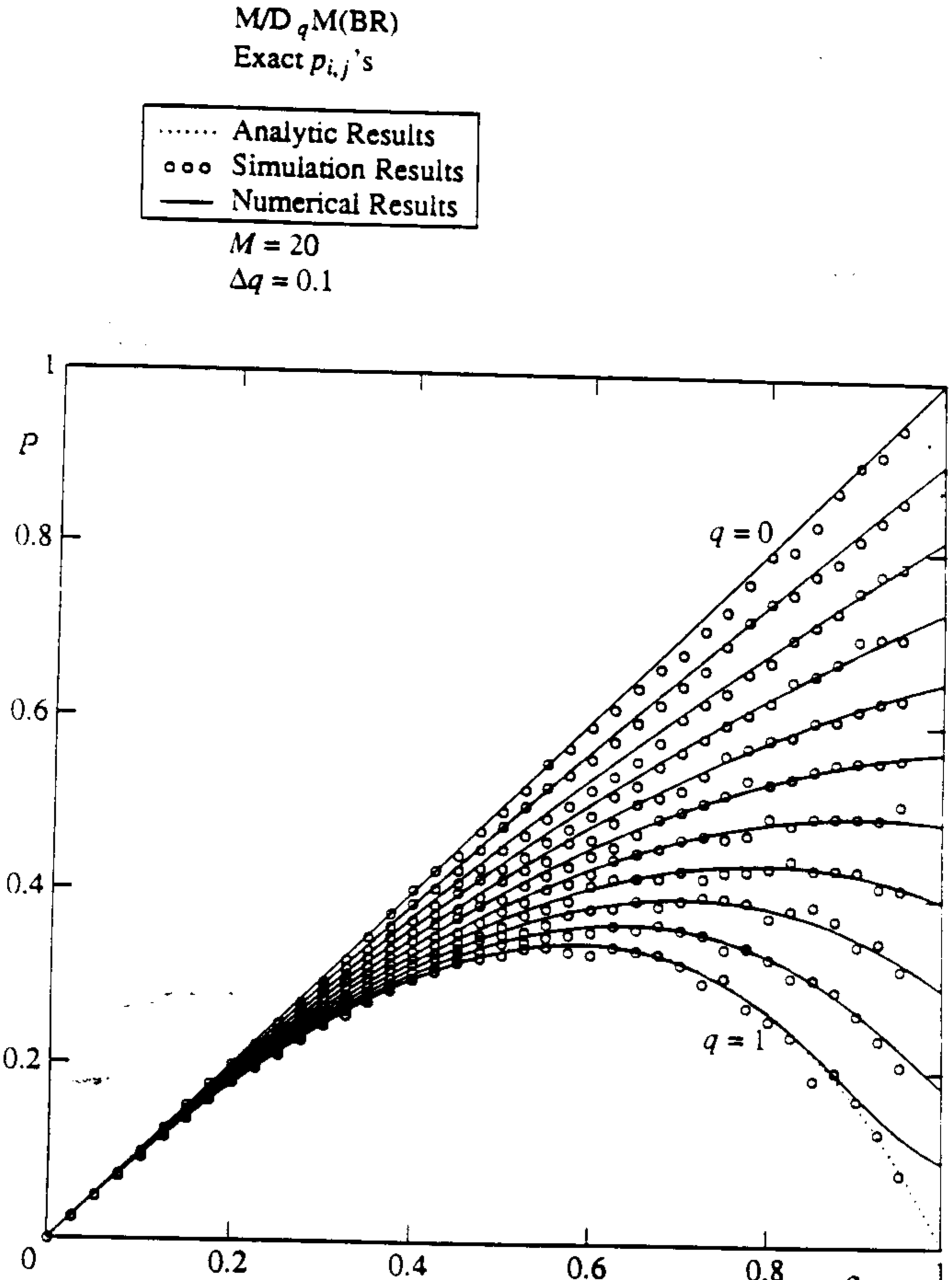


Figure 3.16: Normalized Power for $M/D_qM(BR)$

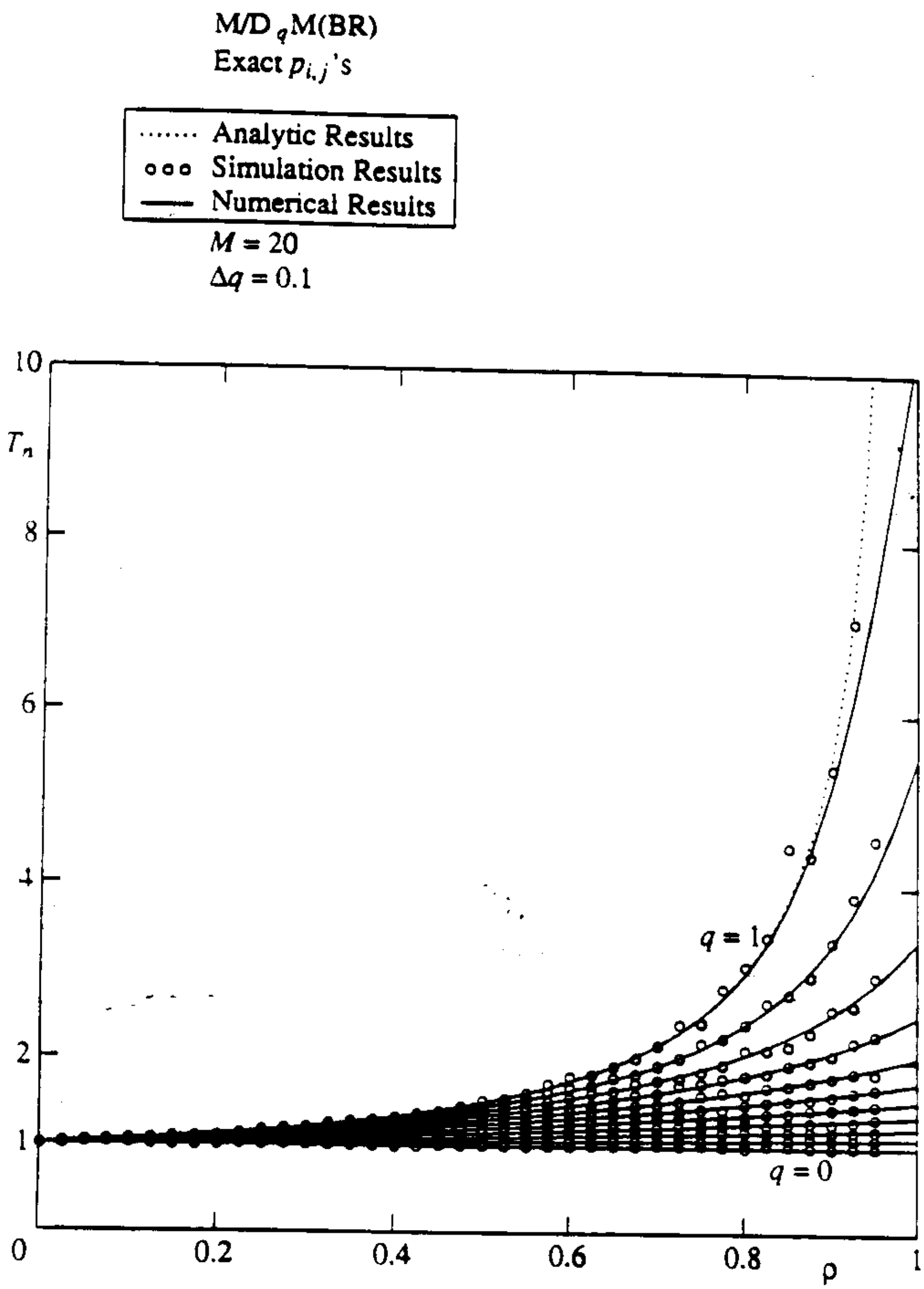


Figure 3.17: Normalized Average Response Time for M/D$_q$M(BR)

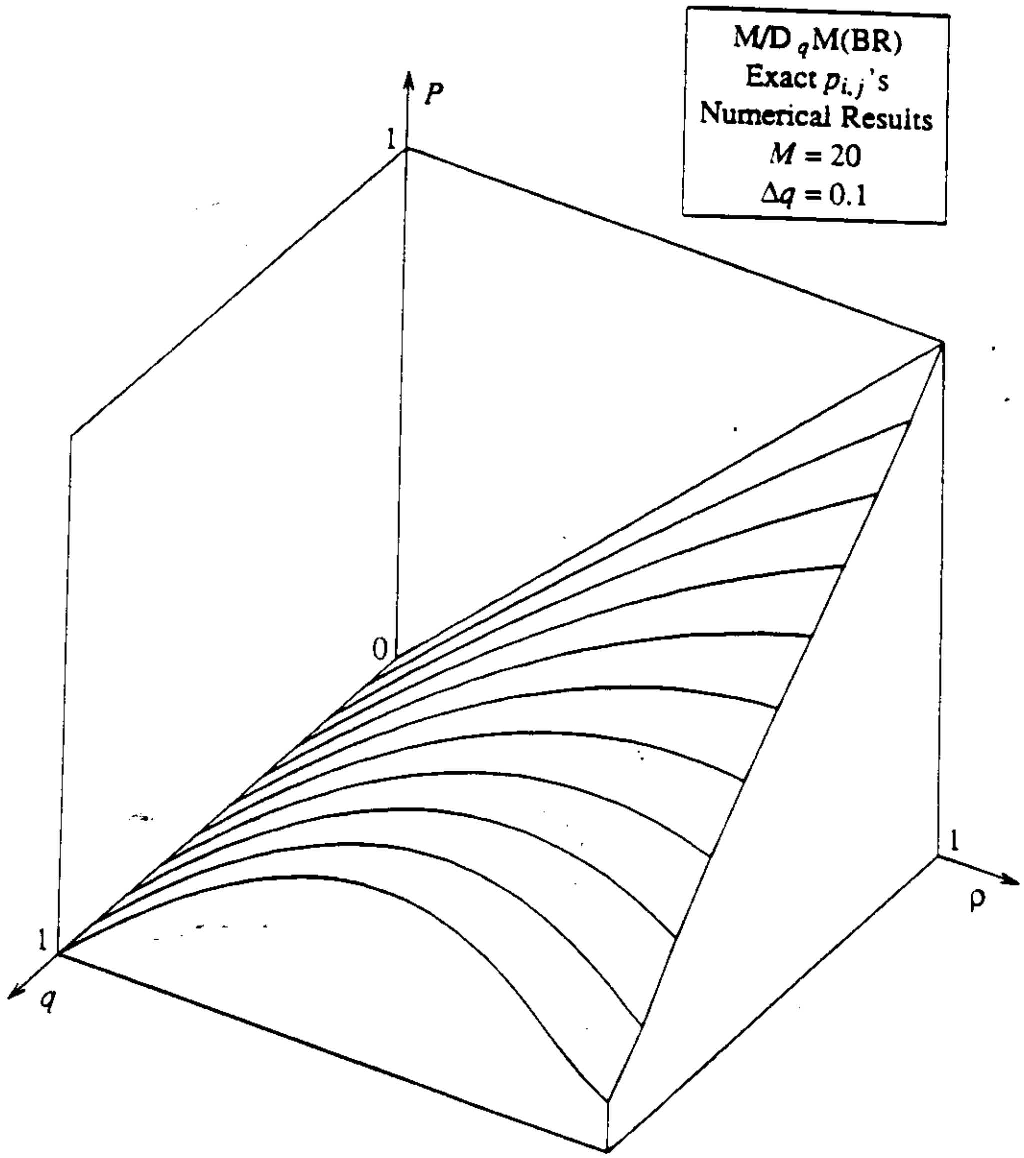


Figure 3.18: Normalized Power in 3-D for $M/D_qM(BR)$

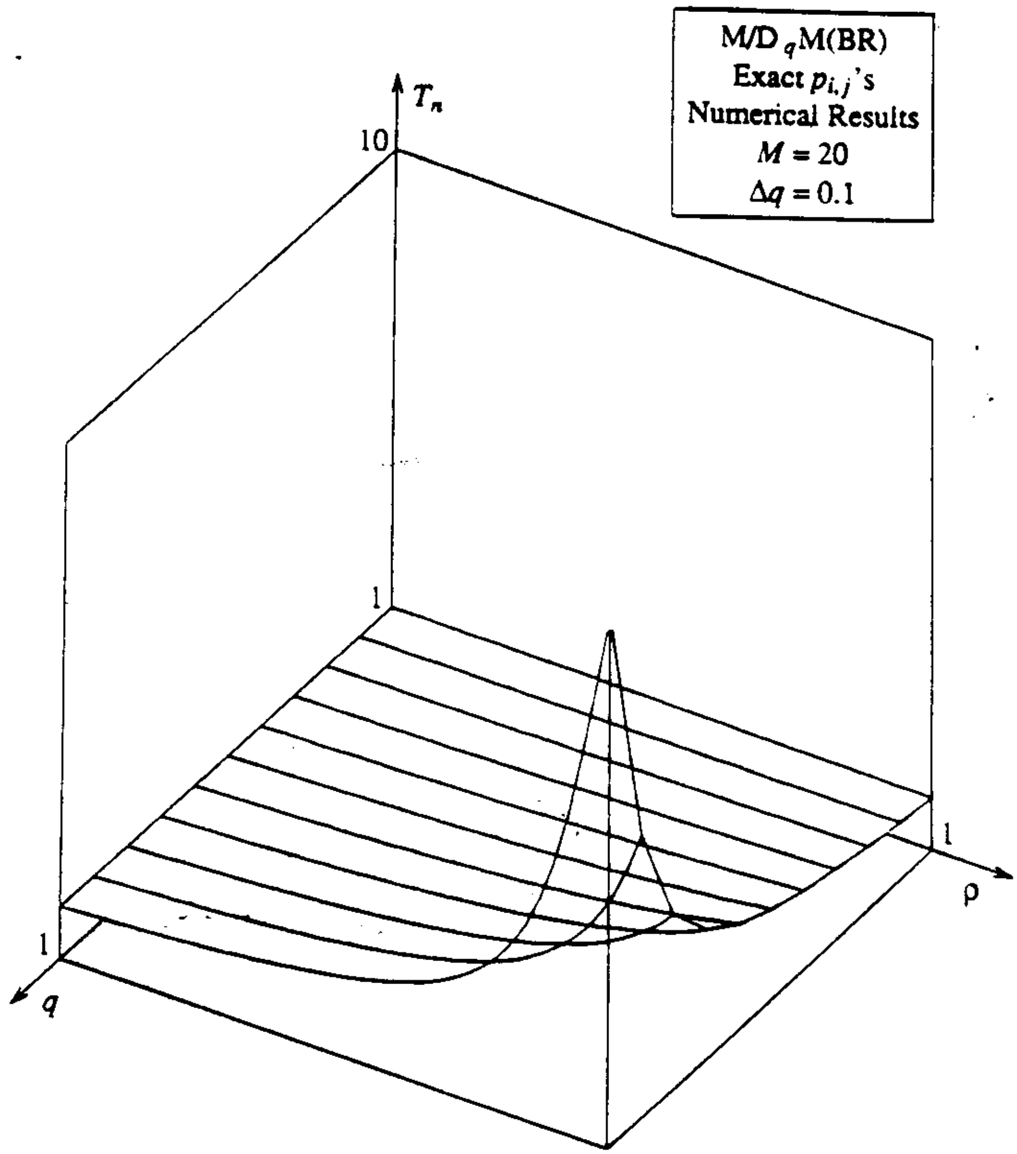


Figure 3.19: Normalized Average Response Time in 3-D for M/D$_q$M(BR)

winner queues.

3.3 Poisson Winner Queues with Partial Restarts

3.3.1 Definition and Structure

Winner queues with partial restarts are more general than the simple winner queues $M/D_qM(CRS)$. In these systems, the initial-to-restart ratio r may be *less* or *greater* than 1. In addition to that, the distribution of the restarted service times may be *different* from the initial service time distribution. The most general case of the winner queues with partial restarts considered here is the queue $M/D_qM\text{-}D_{q_r}M(CRS,r)$, where CRS is one of the previously defined conflict resolution schemes.

3.3.1.1 Structure

Here we study winner queues with Poisson arrivals and partial restarted service times. *Poisson winner queues with partial restarts*, with D_qM-distributed initial service times, $D_{q_r}M$-distributed restarted service times, $M/D_qM\text{-}D_{q_r}M$-(CRS,r) queues. We define an embedded Markov chain at (winner) departure instants in time, and represent states by the number of customers in the system left behind by departures. Let d_k and N_d represent distribution and average number of customers left behind by departures, and let p_k and N represent distribution and average number of customers in the system. Due to the memoryless

		M	D_qM	D
SR,r	M	E†		
	$D_{q_r}M$			
	D			A†
SN,r				A†
sBR,r	M			
	$D_{q_r}M$			
	D			
sBN,r				
BR,r	M	E†		
	$D_{q_r}M$		E‡	
	D			E†
BN,r				E†
L				

Sys tem	Sec tion	Equa tion	Figure
S1	3.2.4	(3.39)	3.11 thru 3.13
S2	3.2.5	(3.41)	3.14 thru 3.15
S3	3.2.6	(3.43)	3.16 thru 3.19

□ numerical results

○ analytic results

E exact transition probabilities

A approximate transition probabilities

† only for r=1

‡ only for r=1 and q_r=q

Figure 3.20: Overview Table of Results for the Simple Winner Queue

arrivals, $d_k = p_k$ and $N_d = N$ [12]. Using Little's result, we find the normalized average response time as $T_n = N/\rho$, and then we calculate the normalized power as $P = \rho/T_n = \rho^2/N$.

In the winner queues with partial restarts we will vary q in the initial service time distribution D_qM, and find the behaviour of the system similar to the simple winner queues. While in the simulations we will vary both q and q_r simultaneously, in the numerical results we will show how the performance is affected when we vary q at a constant q_r, and vice versa, when we vary q_r at a constant q. In both the simulation and the numerical results we will show the effect of changing the initial-to-restart ratio r on the performance of the systems.

In Section 3.3.2 we first describe and discuss simulation results for six types of winner queues with partial restarts: silent-redraw, silent-noredraw, silent/broadcast-redraw, silent/broadcast-noredraw, broadcast-redraw, and broadcast-noredraw. In Section 3.3.3 we explain the analytical approach and in Section 3.3.4 through Section 3.3.10 we give numerical results for the winner queues with partial restarts M/M(SR,r), M/M-D(SR,r), M/D-M(SR,r), M/D(S,r), M/M-D_{q_r}M-(sBR,r), M/D-D_{q_r}M(sBR,r), M/D(sB,r), and M/D_qM-D_{q_r}M(BR,r). Summary of the winner queues with partial restarts is given in Section 3.3.11. In that section we specify which systems from Figure 1 are covered with the simple winner queues analyzed in this chapter.

System	Parameters
$M/D_qM(SR,r)$	$q = 0, 0.5, 1,\ r = 0, 0.25, 0.5, 0.75, 1$
$M/D_qM(SN,r)$	$q = 0, 0.5, 1,\ r = 0, 0.25, 0.5, 0.75, 1$
$M/D_qM(sBR,r)$	$q = 0, 0.5, 1,\ r = 0, 0.25, 0.5, 0.75, 1$
$M/D_qM(sBN,r)$	$q = 0, 0.5, 1,\ r = 0, 0.25, 0.5, 0.75, 1$
$M/D_qM(BR,r)$	$q = 0, 0.5, 1,\ r = 0, 0.25, 0.5, 0.75, 1$
$M/D_qM(BN,r)$	$q = 0, 0.5, 1,\ r = 0, 0.25, 0.5, 0.75, 1$

Table 3.7: Simulation Runs for Poisson Winner Queues with Partial Restarts

3.3.2 Simulation Results

In the six simulation runs, results for the queues specified in Table 3.7 are obtained.

Figure 3.21 through 3.35 show that, for a given r, we have the same behavior as the corresponding $M/D_qM(CRS)$ queues. This means that redraw systems perform better than noredraw systems, broadcast systems perform better than silent systems, and that systems with smaller q perform better for redraw systems, while for noredraw systems higher q gives better results. The above behavior is depicted in Table 3.8, where arrows represent direction from worse to better systems.

Further observations from Figure 3.21 through 3.35 concern the effect of different values of r. Smaller r means that service times upon restarts are smaller.

Systems Compared	Better	Worse
Silent vs. Broadcast		√
	√	
Redraw vs. Noredraw	√	
		√
Redraw/Memoryless vs. Redraw/Deterministic	√	
		√
Noredraw/Memoryless vs. Noredraw/Deterministic		√
	√	
Small r vs. Large r	√	
		√

Table 3.8: Effect of System Parameters to the Performance

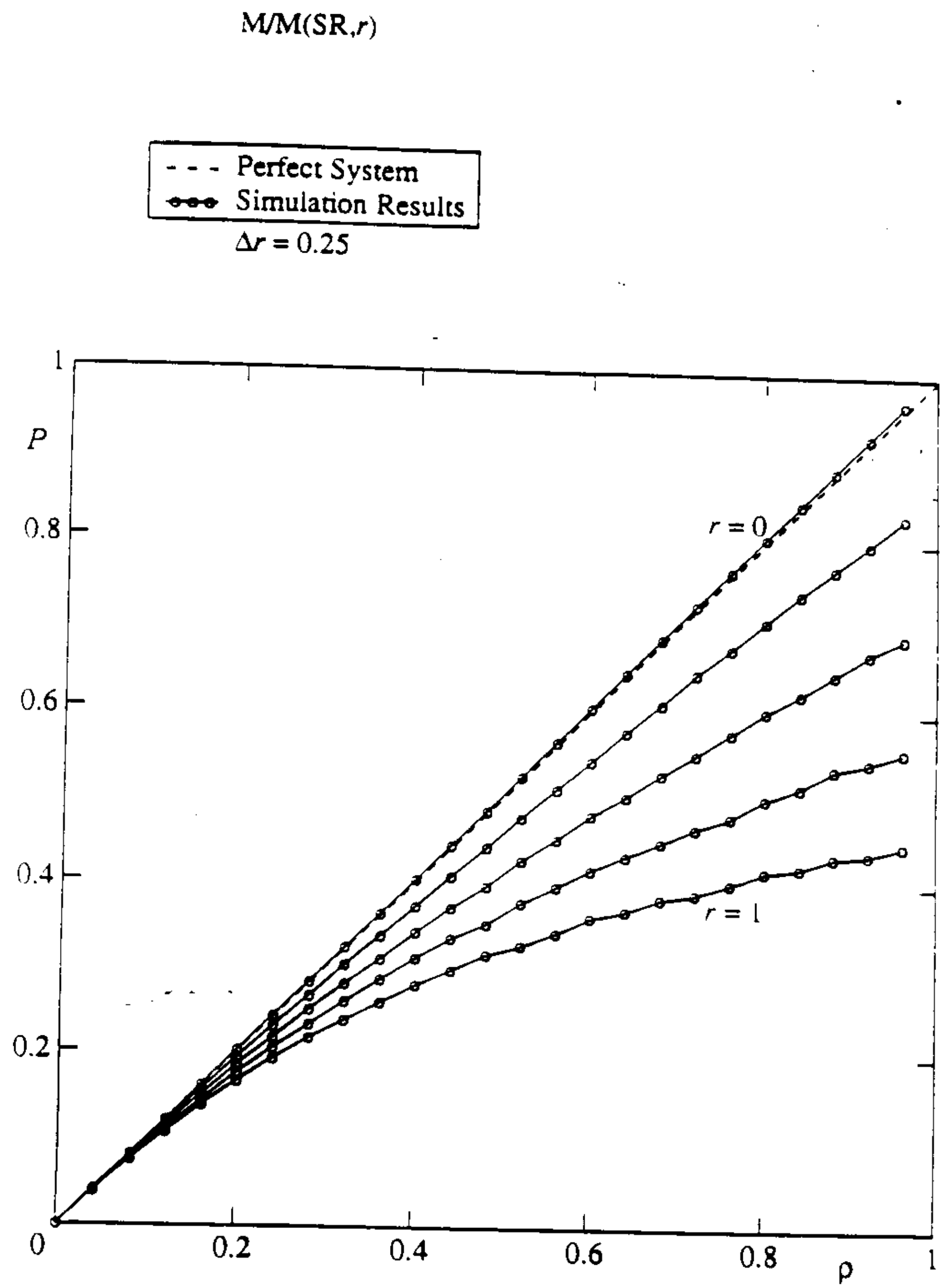


Figure 3.21: Simulation Results for M/M(SR,r)

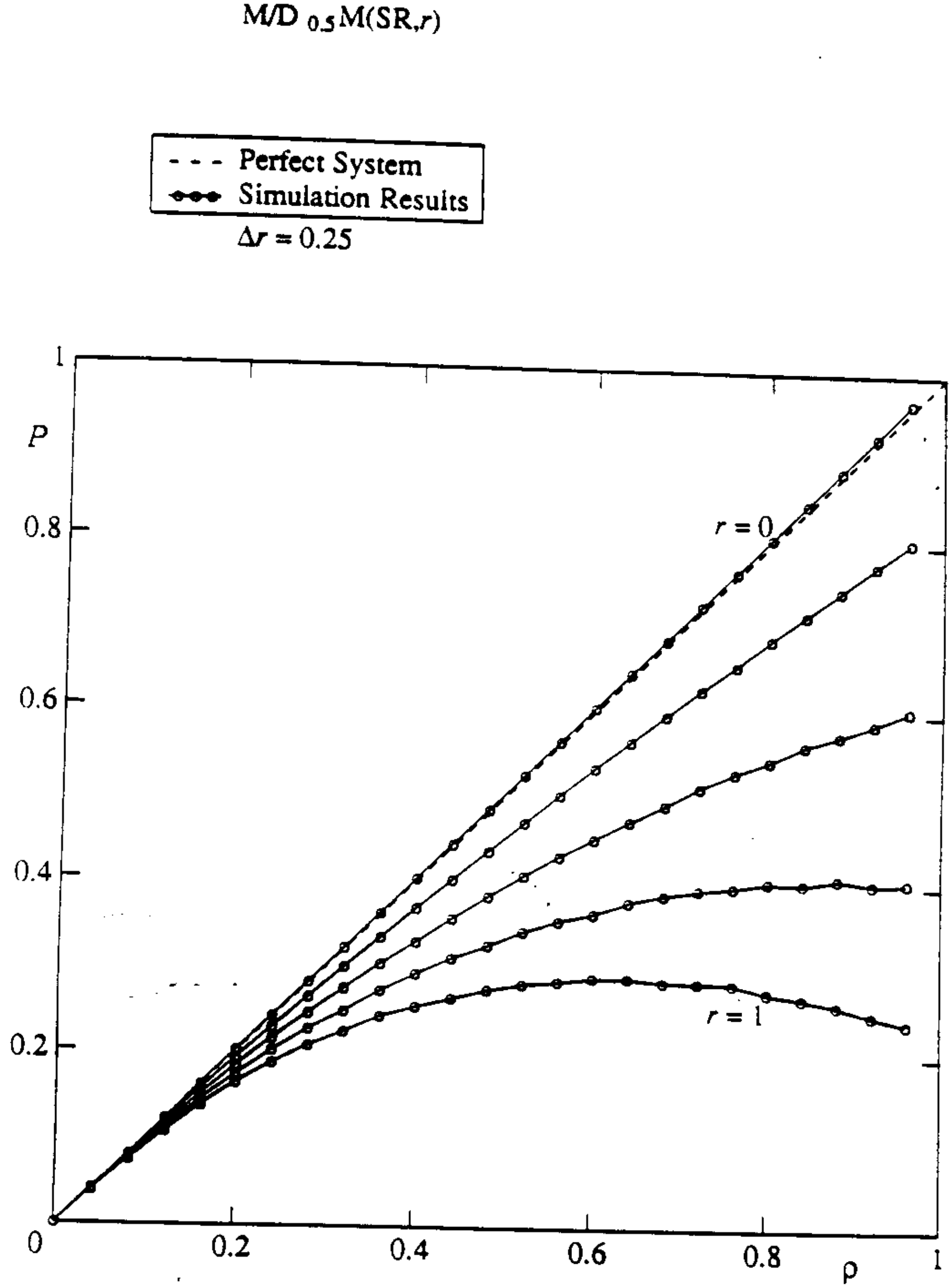


Figure 3.22: Simulation Results for $M/D_{0.5}M(SR,r)$

M/D(S,r)

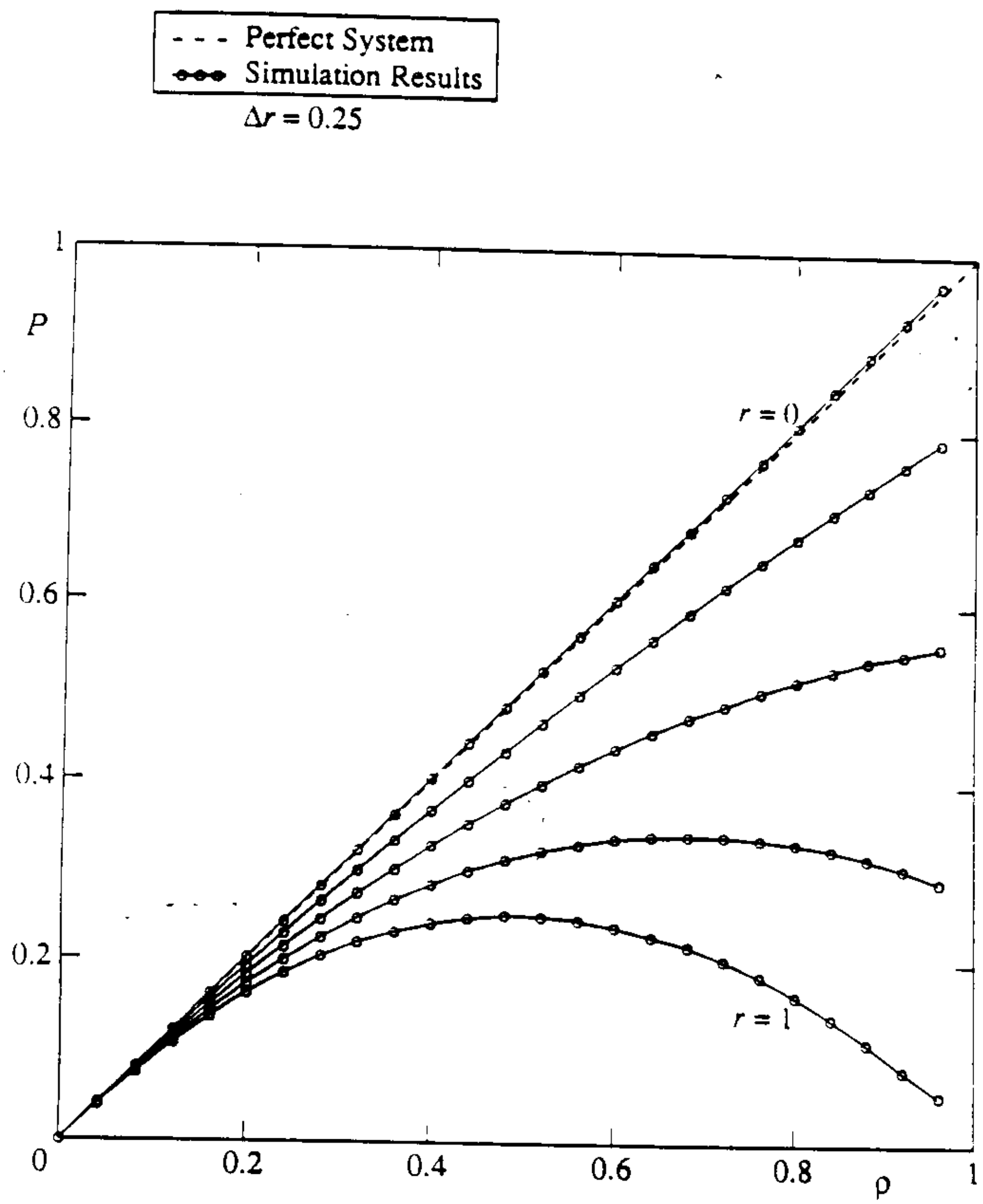


Figure 3.23: Simulation Results for M/D(S,r)

M/M(SN,r)

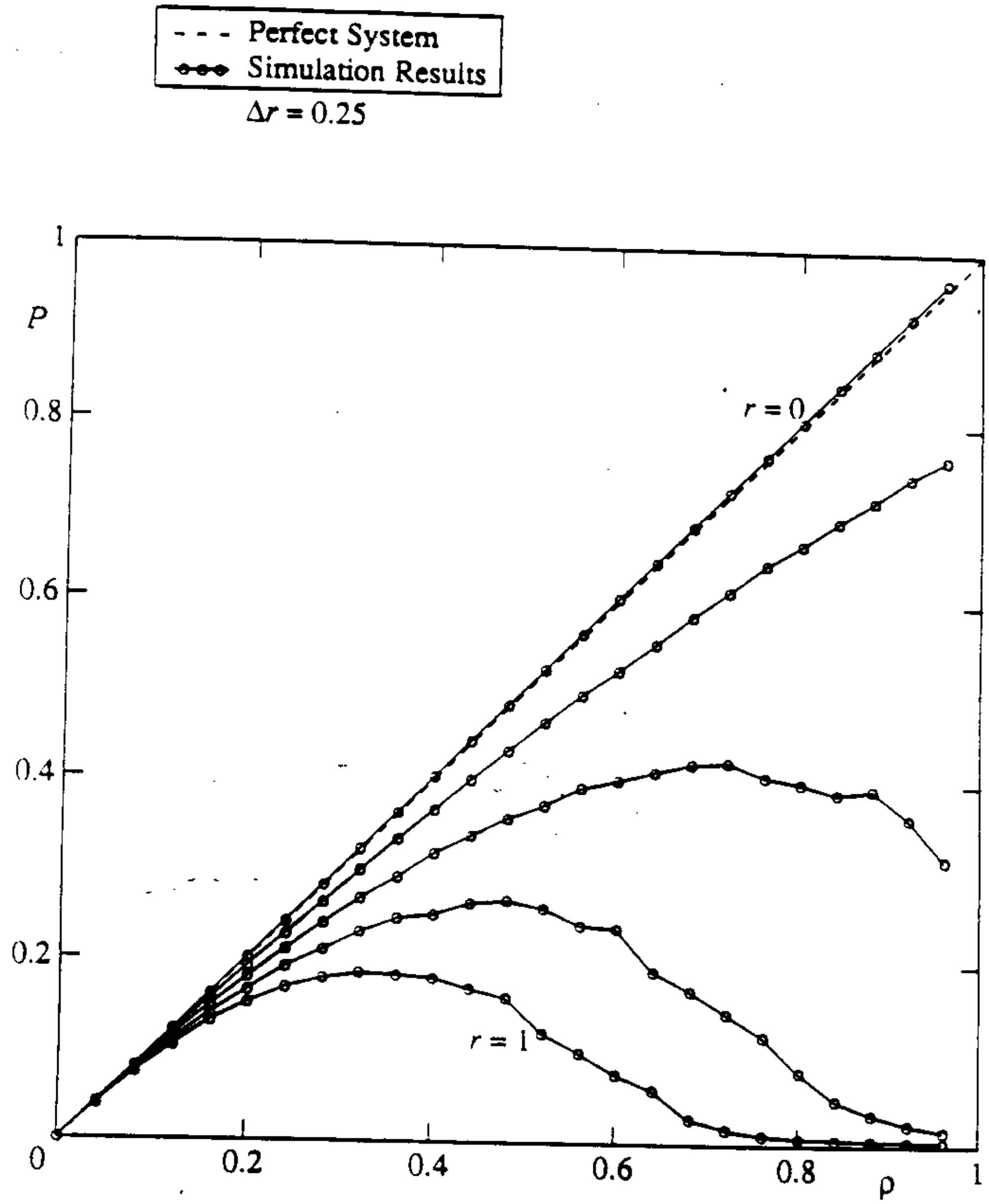


Figure 3.24: Simulation Results for M/M(SN,r)

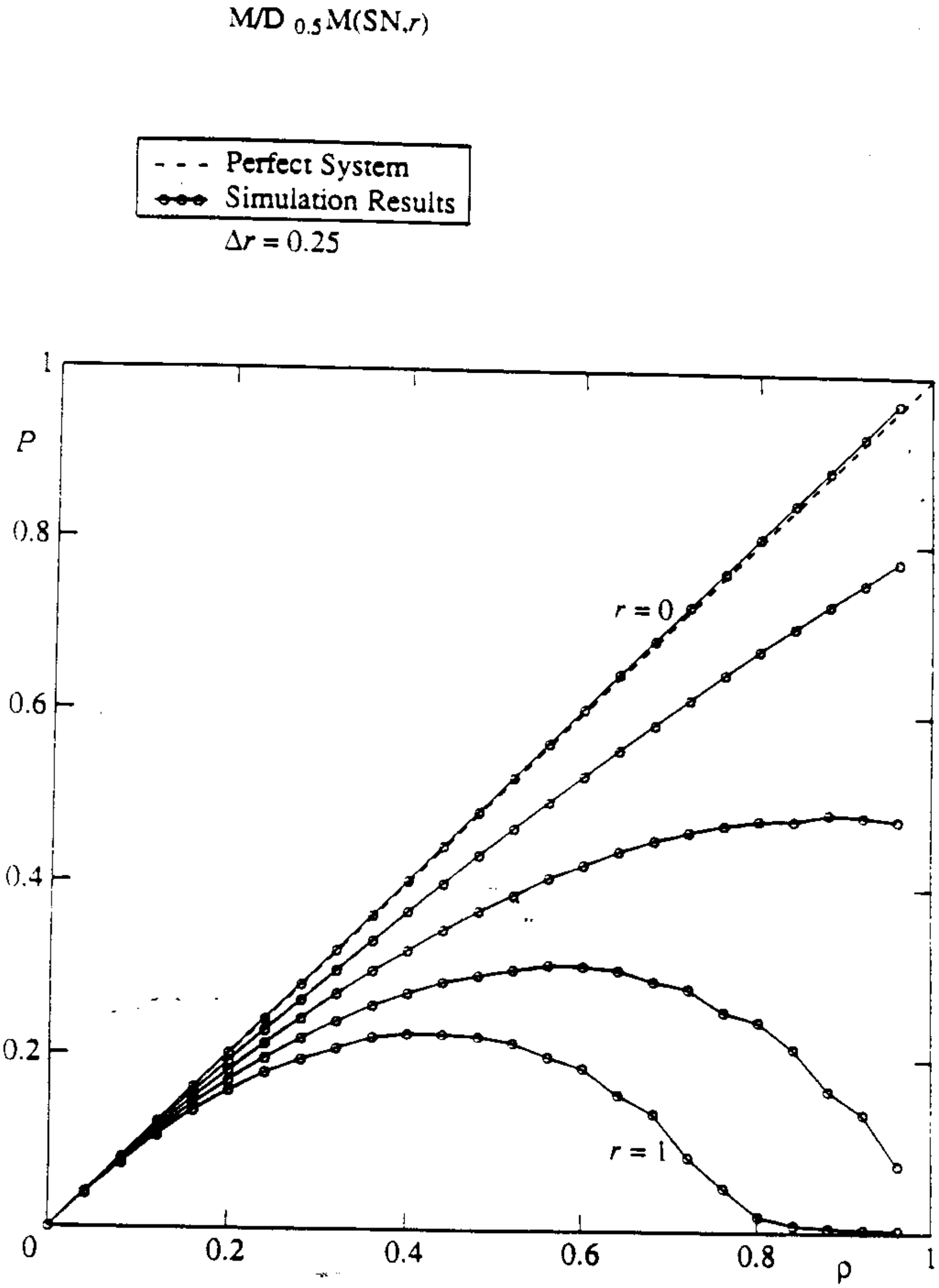


Figure 3.25: Simulation Results for $M/D_{0.5}M(SN,r)$

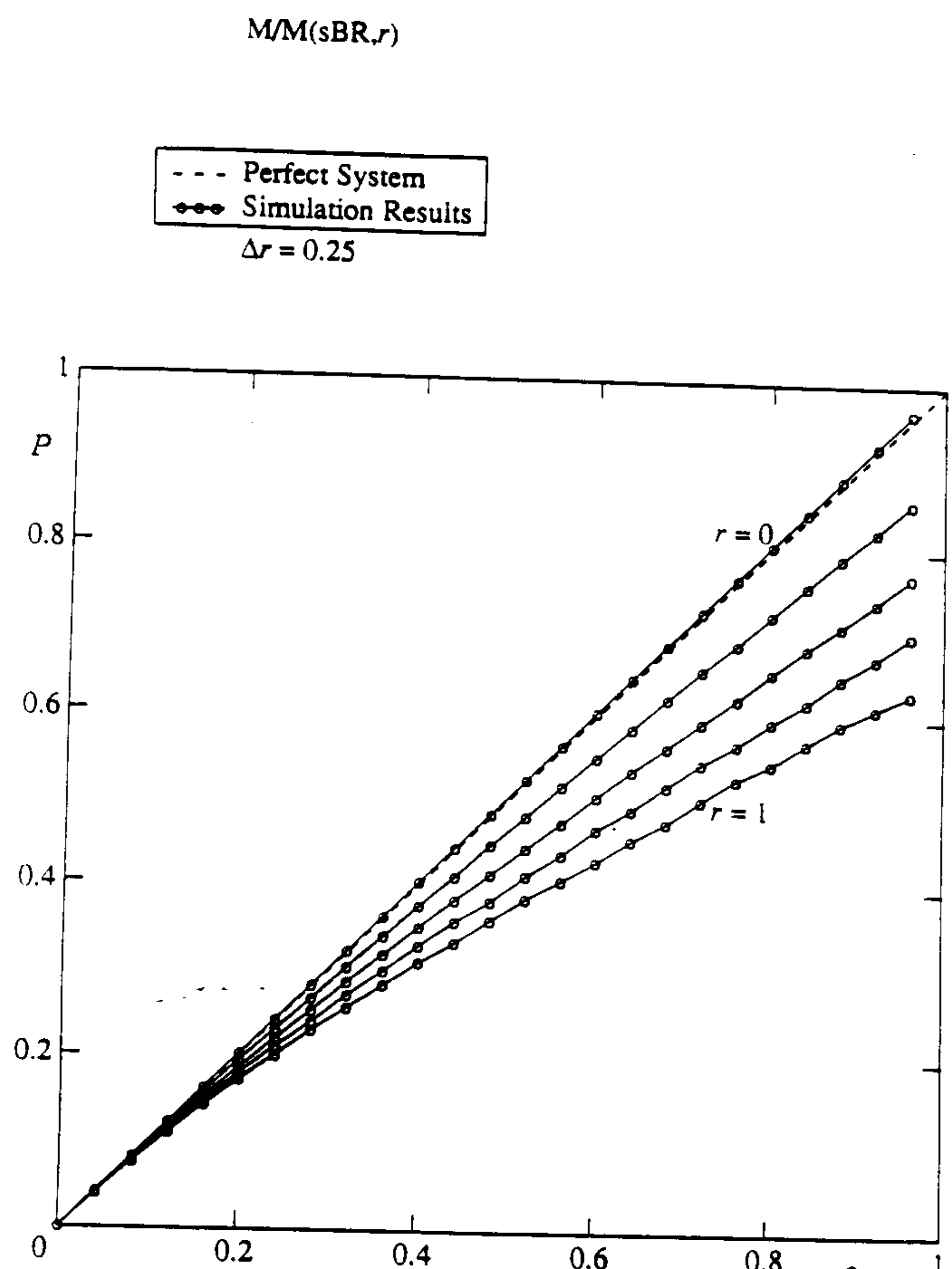


Figure 3.26: Simulation Results for M/M(sBR,r)

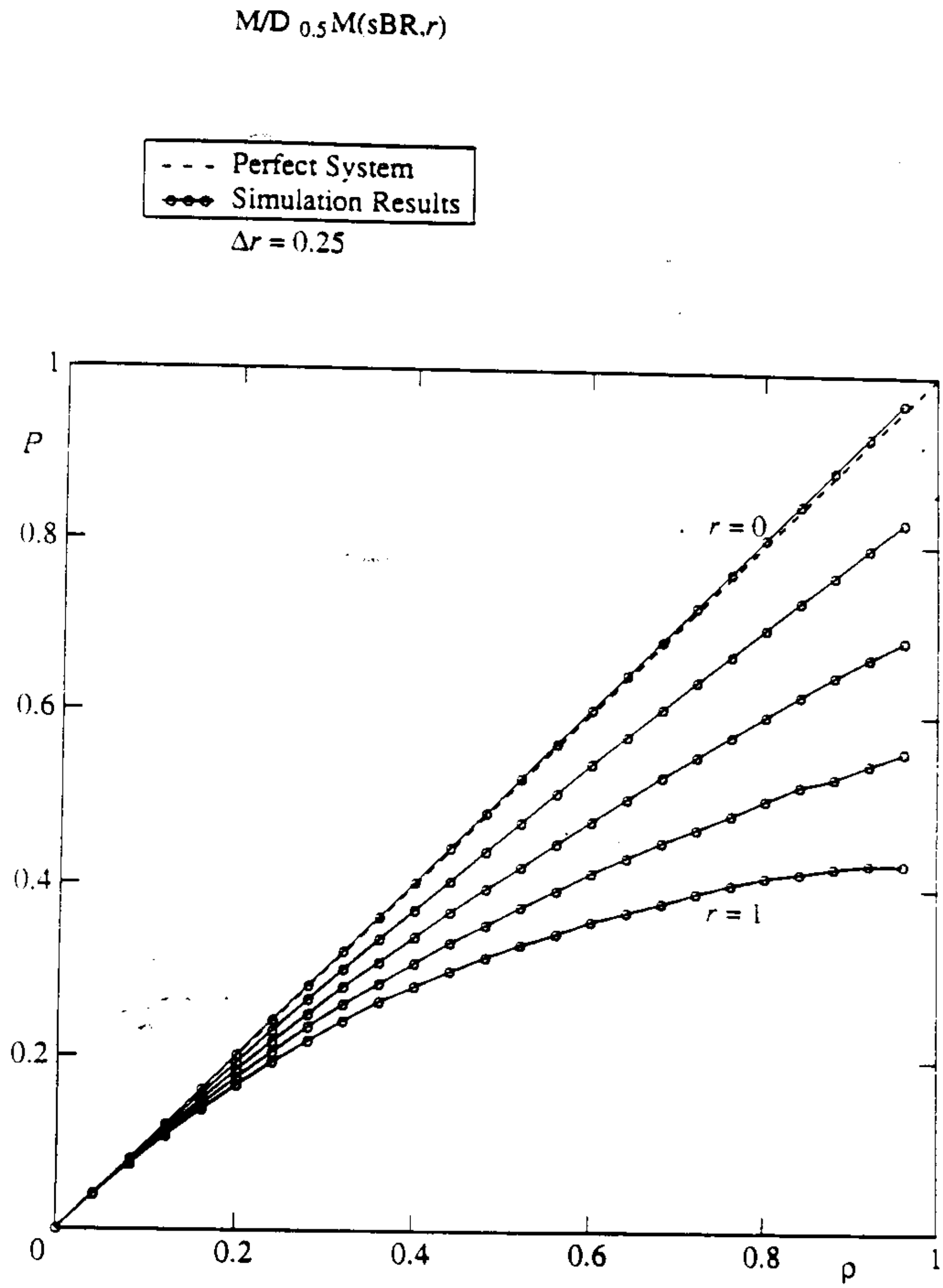


Figure 3.27: Simulation Results for $M/D_{0.5}M(sBR,r)$

M/D(sB,r)

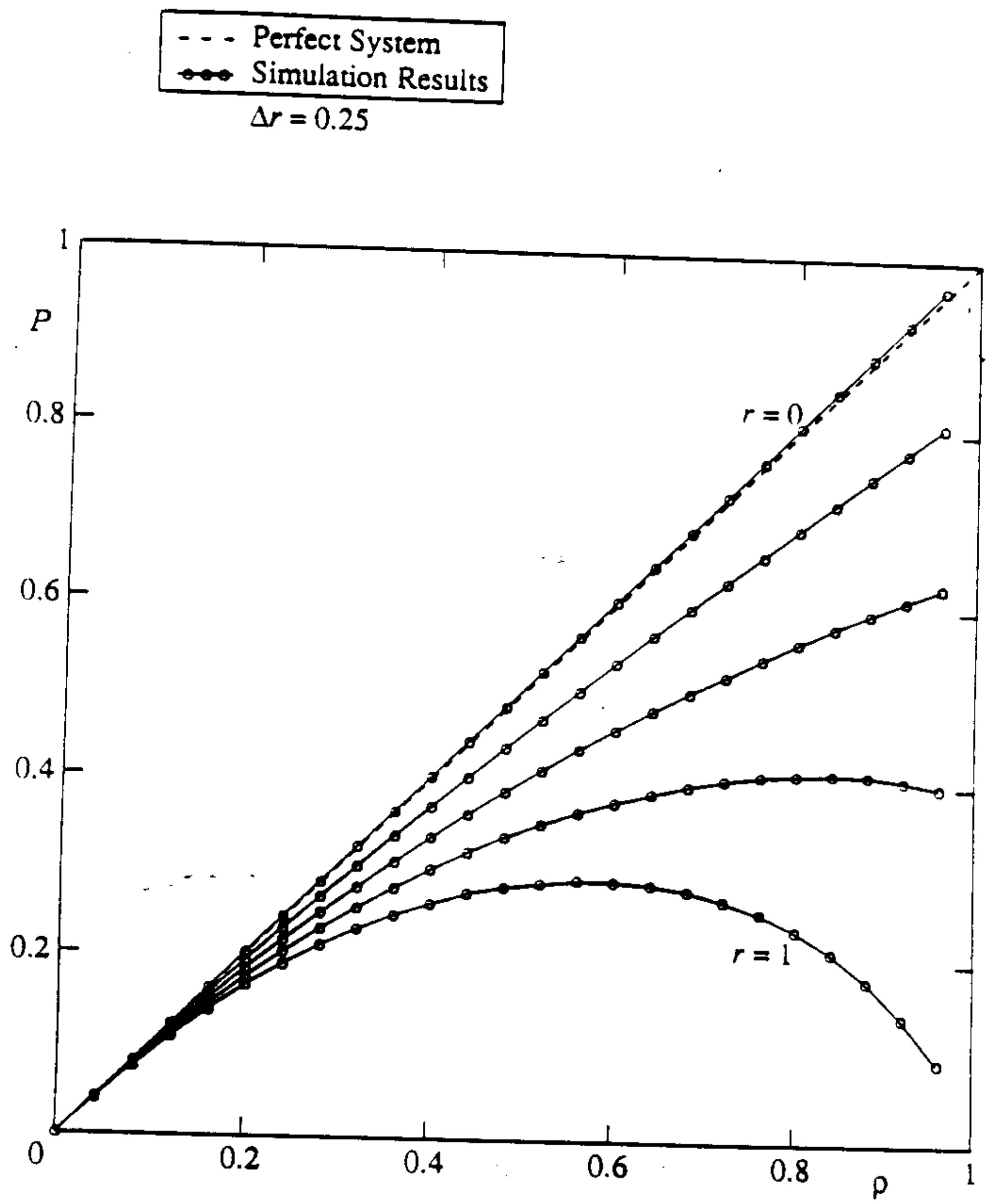


Figure 3.28: Simulation Results for M/D(sB,r)

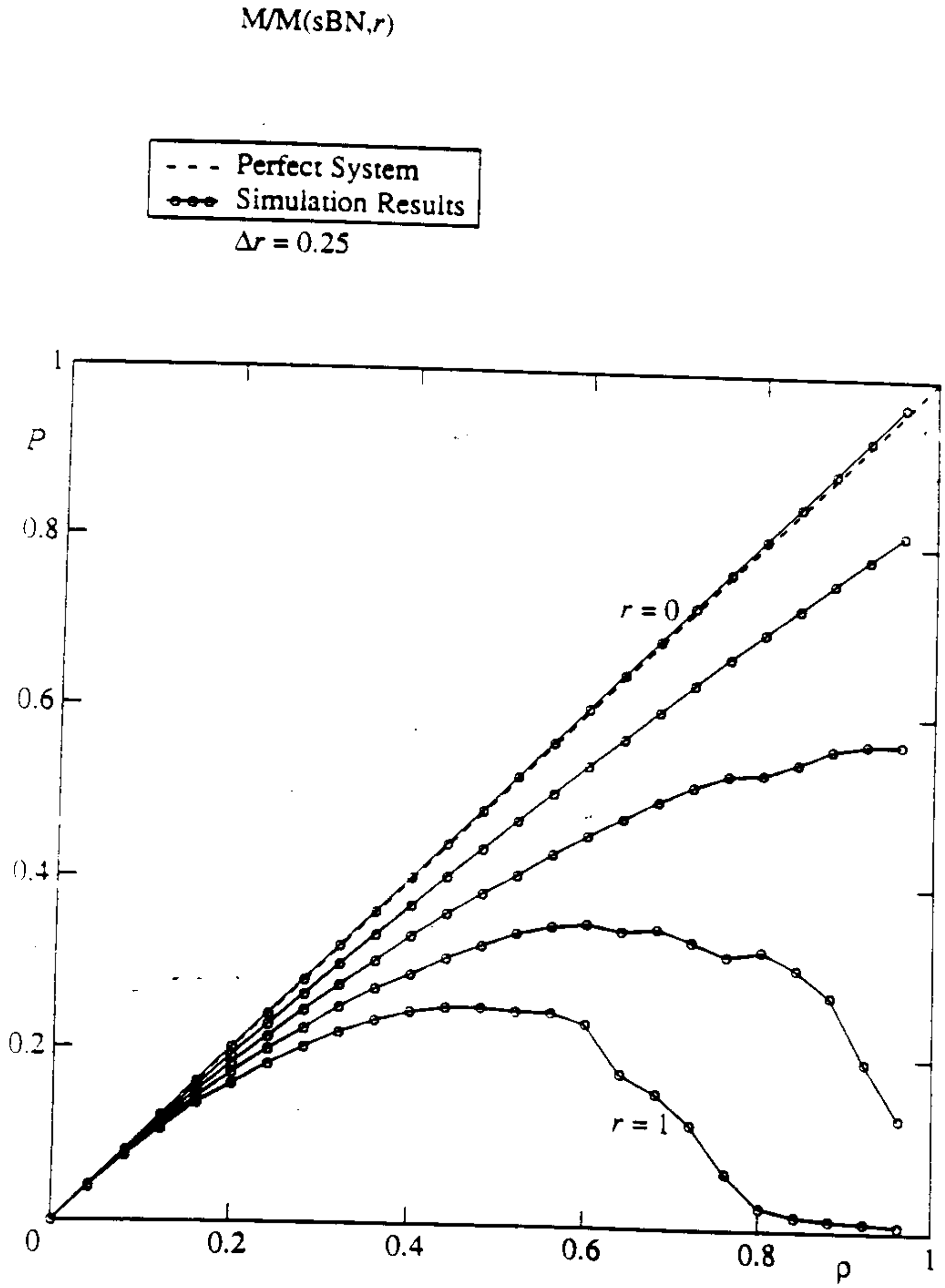


Figure 3.29: Simulation Results for M/M(sBN,r)

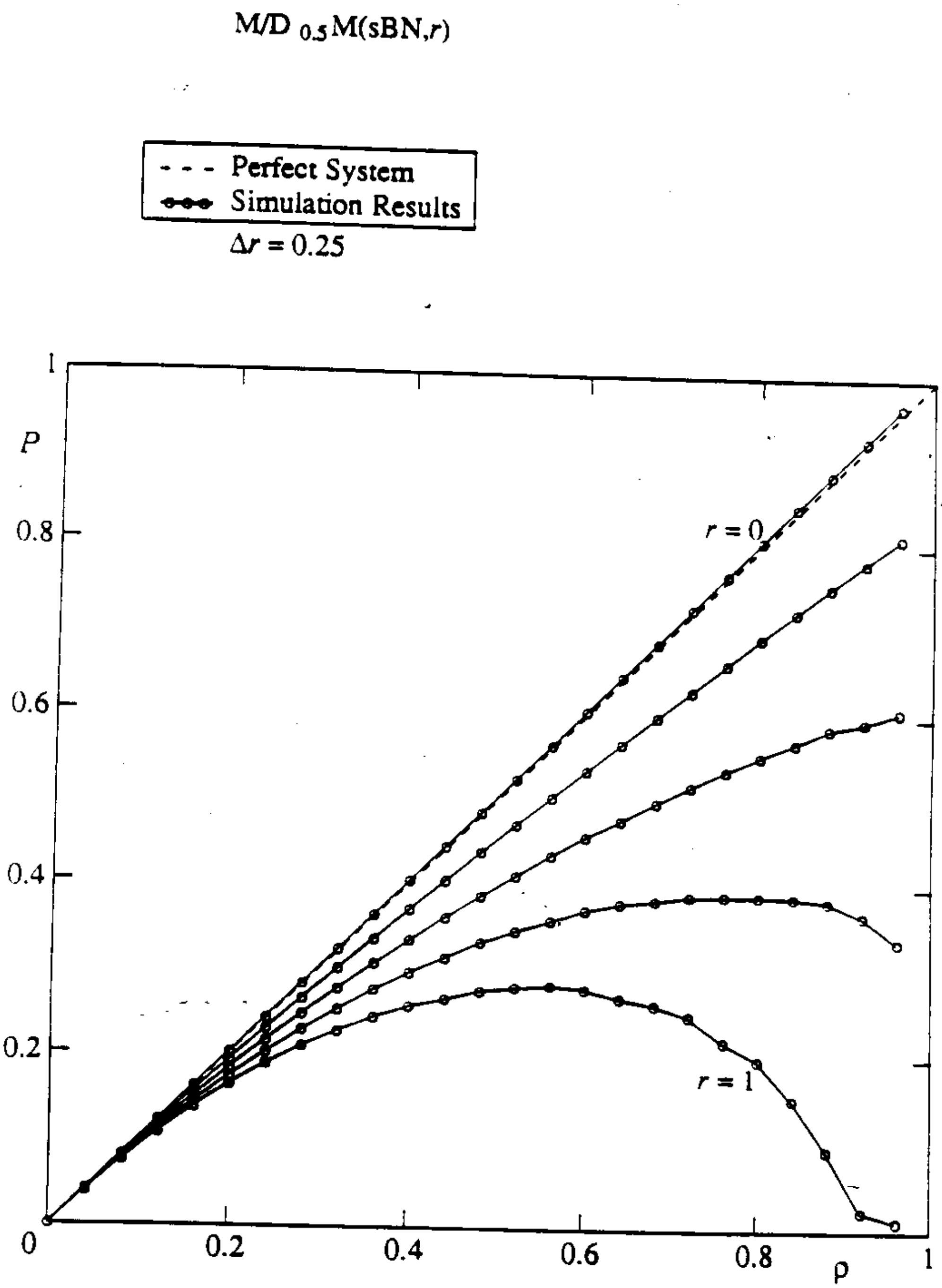


Figure 3.30: Simulation Results for $M/D_{0.5}M(sBN,r)$

resulting in better performance. For $r = 1$ we have the same results as for $M/D_qM(CRS)$ queues presented in Section 2.1. In fact, those queues are the special case of $M/D_qM(CRS,r)$ queues, i.e., $M/D_qM(CRS) \equiv M/D_qM(CRS,1)$. For $r = 0$ we have "perfect" systems, where every customer's first service is completely useful, and the customer leaves after that whether the service was successful or unsuccessful. For BR and BN protocols we have a non-realistic situation, where power gets better than "perfect" for small r. This happens because the system time of a loser is shorter than his assigned service time, which results in the average system time being smaller than the average service time $\overline{x}$. To make the broadcast protocol more realistic, we modify this rule and force each customer to complete his first service entirely, regardless of any broadcast messages from winners. In other words, their first service is *silent*, or, better yet, "deaf". We denote these protocols sBR and sBN. Figure 3.31 through 3.35 show that even for $r = 0$ the power is not better than "perfect". Queues $M/D_qM(sB,r)$ do not have any of the $M/D_qM(B)$ queues as a special case.

In the following sections we analyze some queues that are even more general than the ones presented in this section.

3.3.3 Analysis

As for the simple Poisson winner queues, we also define d_k as $d_k \stackrel{\text{def}}{=} P[k$ customers left in the system (by a departure)].

We use Equations (3.1) and (3.2) to calculate the normalized average system

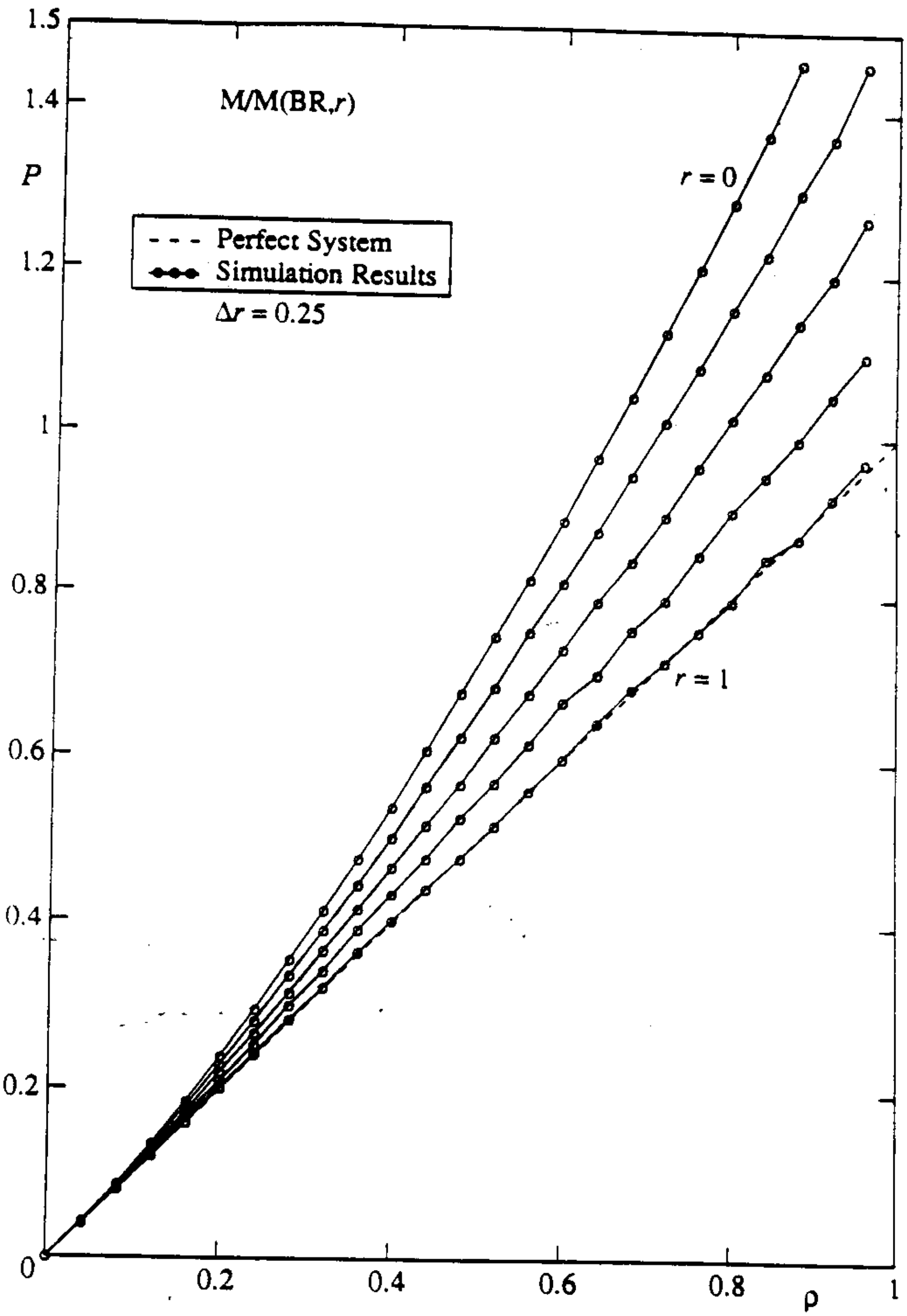


Figure 3.31: Simulation Results for M/M(BR,r)

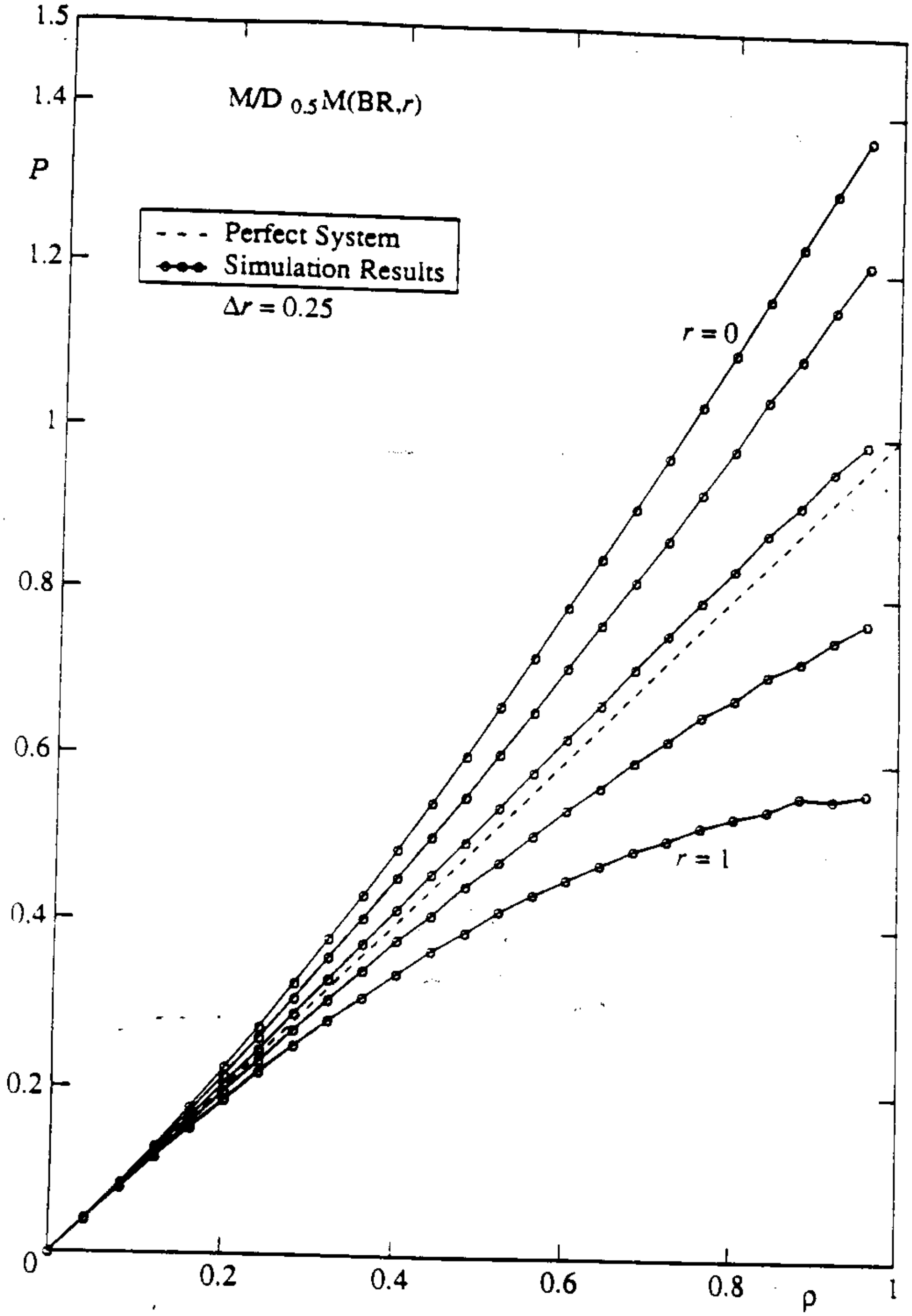


Figure 3.32: Simulation Results for $M/D_{0.5}M(BR,r)$

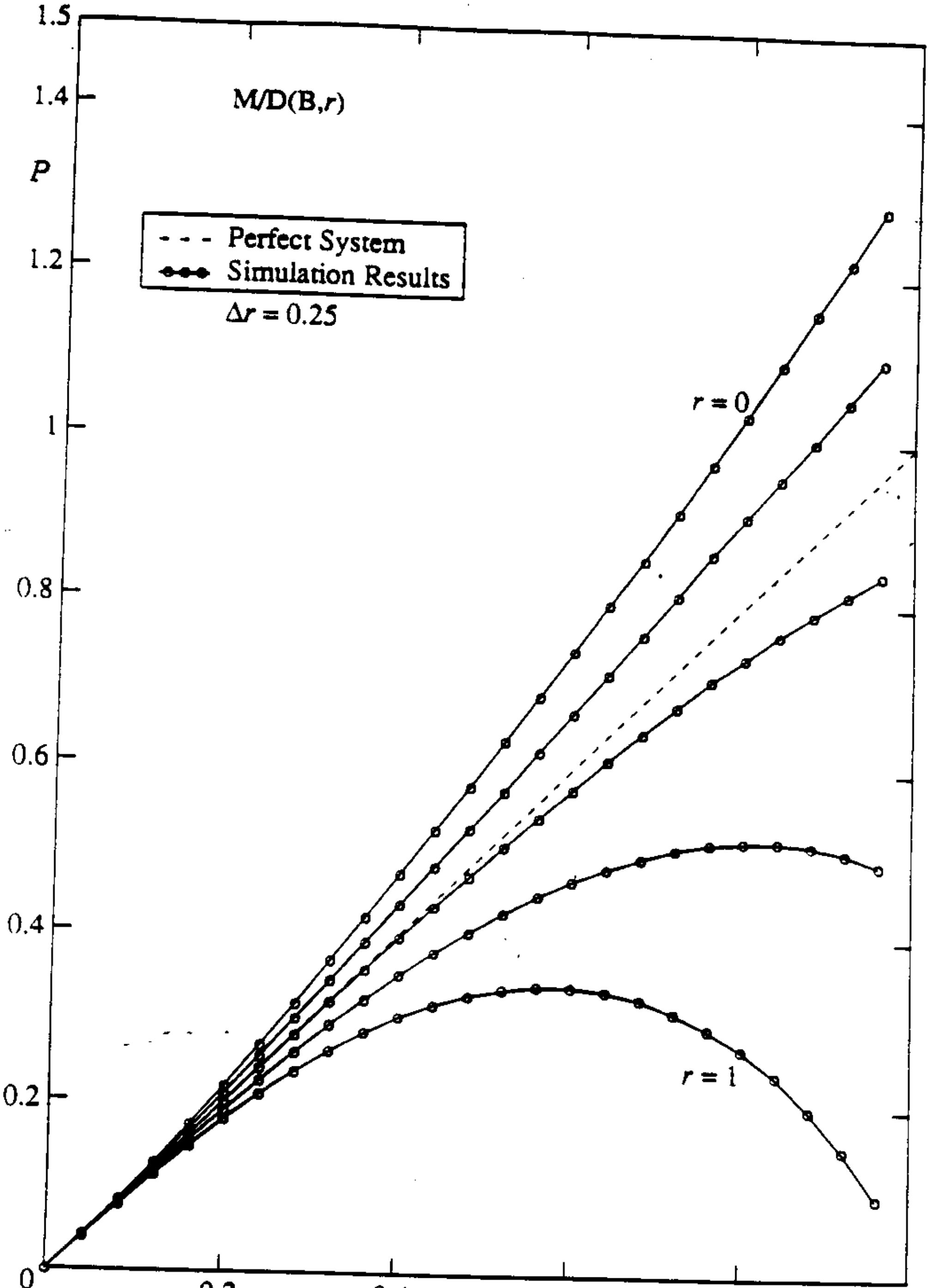


Figure 3.33: Simulation Results for M/D(B,r)

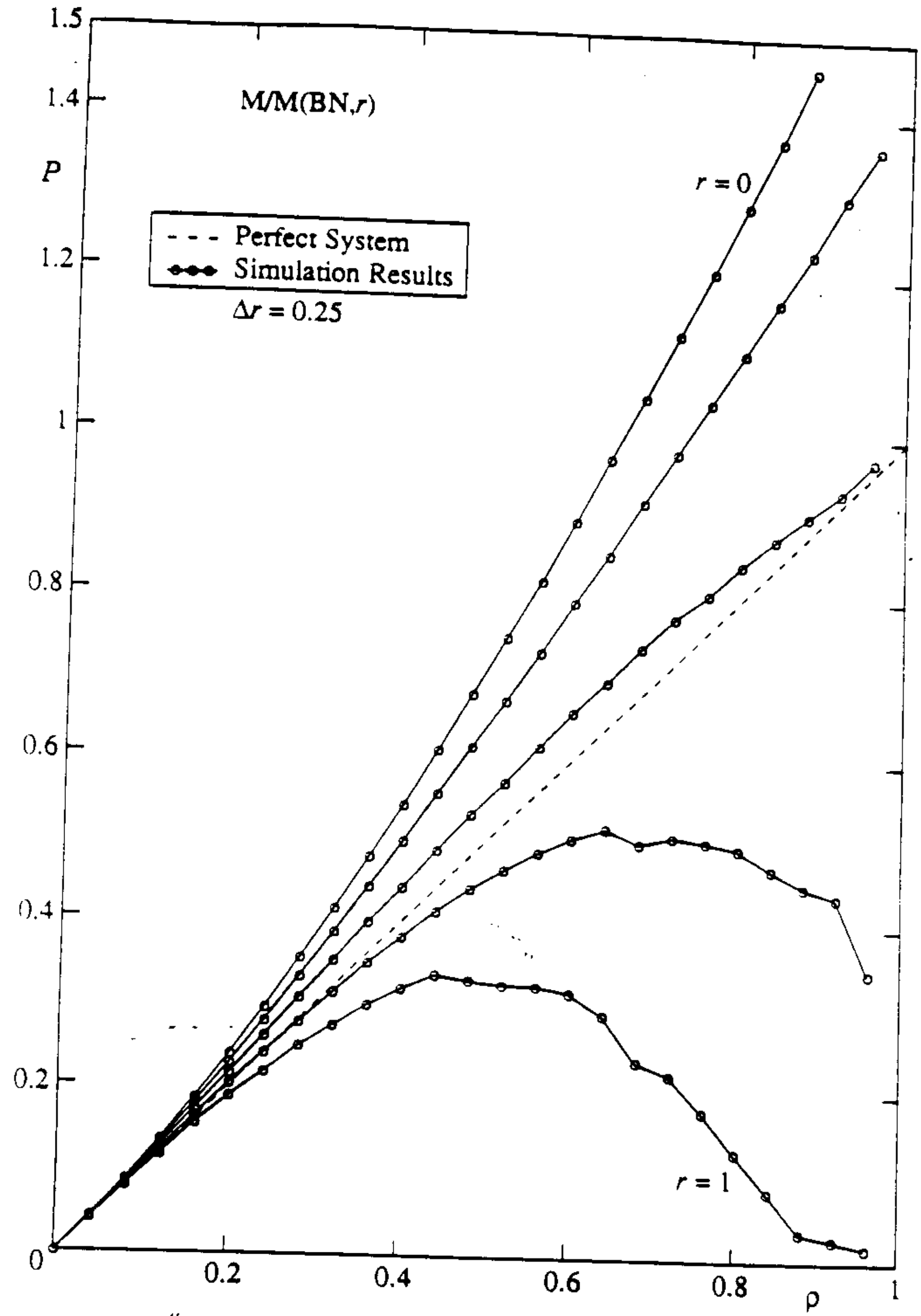


Figure 3.34: Simulation Results for M/M(BN,r)

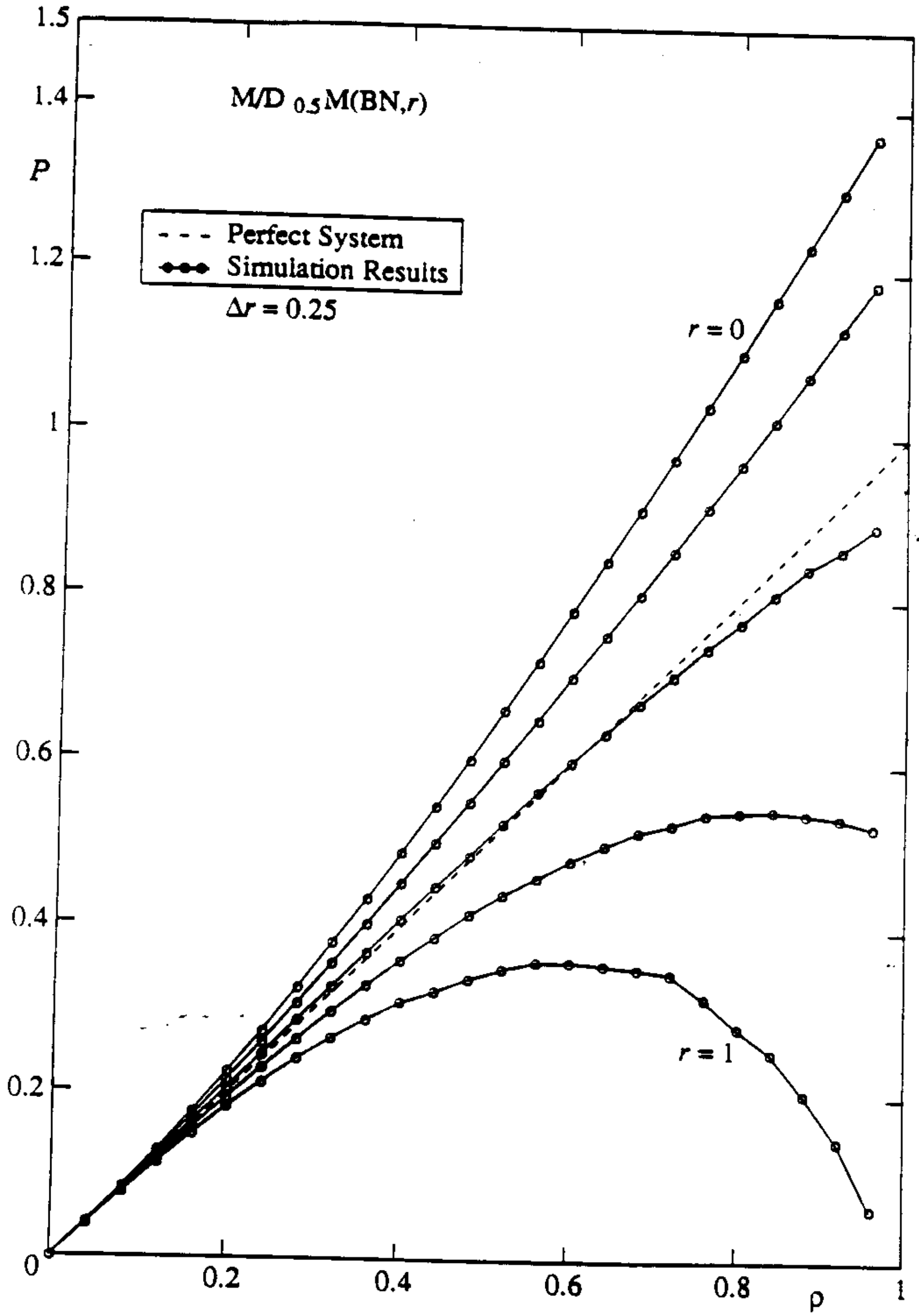


Figure 3.35: Simulation Results for $M/D_{0.5}M(BN,r)$

time T_n and the normalized power P. Definitions of r.v.'s X, X_r, X_s, V, and U from Table 3.2 are used here again. The variables are shown in Figure 3.6.

The first service time in Poisson winner queues with partial restarts has a different distribution from the service times upon restarts. Therefore, we distinguish between two types of customers with respect to a target departure D. *Initial* customers are those being served for the first time when departure D occurs. *Old* customers are those being served upon restart when departure D occurs.

We now define $d_{m,n}$ to be the probability of m old and n initial customers left in the system by a departure. i.e., $d_{m,n} \stackrel{\text{def}}{=} P[m$ old and n initial customers left in the system]. The following holds.

$$d_k = \sum_{m=0}^{k} d_{m,k-m}, \quad k = 0, 1, 2, \ldots \tag{3.44}$$

In order to find $d_{m,n}$ we use a two dimensional embedded Markov chain, where one dimension is the number of old customers left in the system, and the other dimension is the number of initial customers left in the system. In Table 3.9 we further define $p_{i,k,j,l}$'s, the transition probabilities between the states.

In the sections below we find expressions for $p_{i,k,j,l}$, from which we can calculate probabilities $d_{m,n}$ by solving the following equations.

$$d_{m,n} = \sum_{i=0}^{\infty} \sum_{k=0}^{\infty} d_{i,k} p_{i,k,m,n}, \quad m, n = 0, 1, 2, \ldots \tag{3.45}$$

Symbol	Definition
$p_{i,k,j,l}$	P[a departure leaves j old and l initial customers in the system, given the previous departure left i old and k initial customers in the system], $i,k,j,l = 0,1,2,\ldots$

Table 3.9: Definition of $p_{i,k,j,l}$

$$1 = \sum_{m=0}^{\infty}\sum_{n=0}^{\infty} d_{m,n} \tag{3.46}$$

The numerical procedure for finding the normalized response time and power is as follows. For reasonably large M (i.e., $M >> 1$) we calculate all the transition probabilities $p_{i,k,j,l}$, $i,k,j,l \geq 0, 0 \leq i+k \leq M$, $0 \leq j+l \leq M$. To preserve the conservation of the probabilities, we assign the following value to the $p_{i,k,M,0}$, $0 \leq i+k \leq M$:

$$p_{i,k,M,0} = 1 - \sum_{j=0}^{M-1}\sum_{l=0}^{M-j} p_{i,k,j,l}, \quad i,k \geq 0,\ 0 \leq i+k \leq M \tag{3.47}$$

We use $(M+1)(M+2)/2$ Equations (3.45) where $m+n \leq M$. Thus, from Equation (3.45) and (3.46) we find $d_{m,n}$ for $0 \leq m+n \leq M$, and, using Equation 3.44, we calculate d_k for $k = 0,1,\ldots,M$. From the d_k's we find N, the average number of customers in the system. We refer to the number M below as the *precision* of the numerical solution. Through Equation (3.1) and (3.2) we use N to obtain the normalized average response time T_n and normalized power

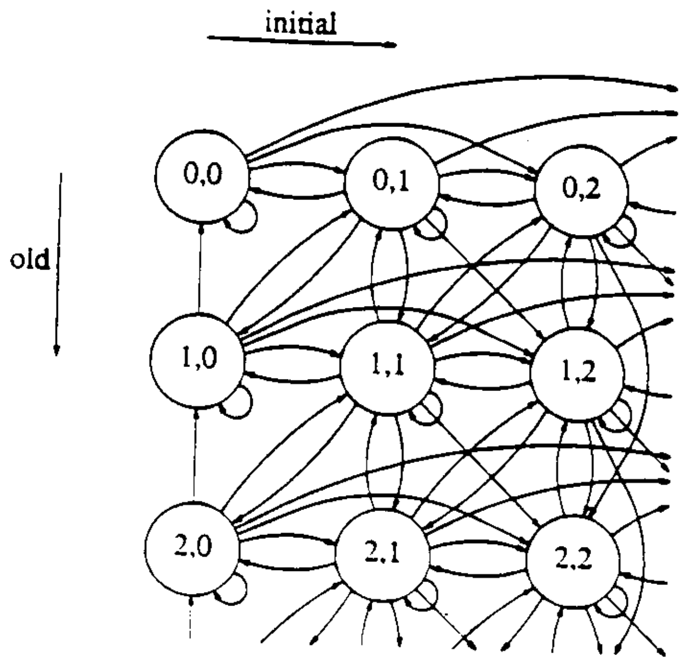


Figure 3.36: Poisson Winner Queue with Partial Restarts State Transition Diagram

P. The numerical procedure from the transition probabilities to the performance measures is shown later as the top part of Figure 3.39.

In the sections below we concentrate on finding the transition probabilities.

3.3.3.1 Finding Transition Probabilities

The embedded Markov chain, with the arcs representing transition probabilities, is shown in Figure 3.36.

There are several situations where transition probabilities equal zero. We know that old customers cannot become initial, while initial customers can become old between two successive departures. The number of old customers cannot decrease by more than one, and the same holds for the total number of customers.

Thus we have

$$p_{i,k,j,l} = 0, \quad (j < i-1) \text{ or } (j+l < i+k-1)$$

On the other hand, new customers can only become initial, not old. Thus we have

$$p_{i,k,j,l} = 0, \quad j > i+k$$

Now we can write that for given $j, l \geq 0$ we have

$$p_{i,k,j,l} = 0, \quad (i > j+1) \text{ or } (i+k > j+l+1) \text{ or } (i+k < j) \tag{3.48}$$

In Figure 3.37 we show, in the shaded closed region, those pairs of values (i, k) for which there *may exist* a transition to state (j, l).

In our calculations for $d_{m,n}$ we will use Equations (3.45) and (3.46), where appropriate transition probabilities $p_{i,k,m,n}$ will take on zero values according to their formulas given in the sections below. However, we here give a more restrictive formula for $d_{m,n}$ based on Equation (3.48).

$$d_{m,n} = \sum_{i=0}^{m} \sum_{k=Mi}^{m+n+1-i} d_{i,k} p_{i,k,m,n} + \sum_{k=0}^{n} d_{m+1,k} p_{m+1,k,m,n}, \quad m, n = 0, 1, 2, \ldots \tag{3.49}$$

Similar to the probability $p_{i,j}(v)$ defined for the simple Poisson winner queues in Table 3.4, we define probability density $p_{i,k,j,l}(v)$ in Table 3.10.

Knowing the probability density $p_{i,k,j,l}(v)$, we find $p_{i,k,j,l}$ as

$$p_{i,k,j,l} = \int_0^\infty p_{i,k,j,l}(v)\, dv \tag{3.50}$$

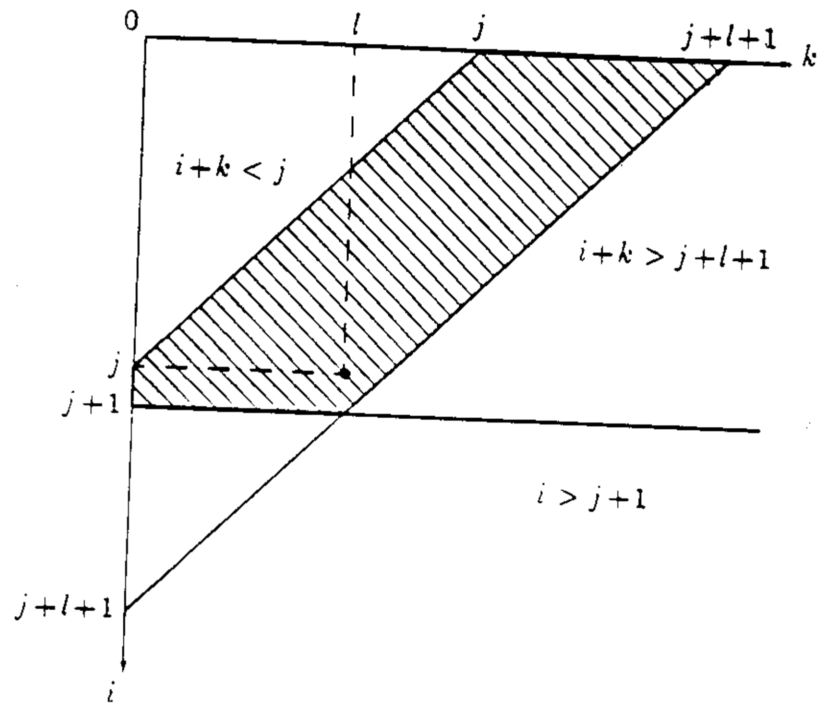


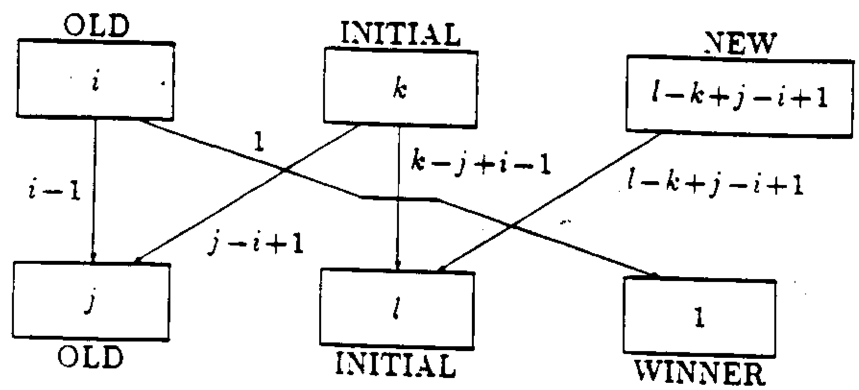
Figure 3.37: Transitions to State (j, l)

Symbol	Definition
$p_{i,k,j,l}(v)$	$\frac{1}{dv}$ P[a departure leaves j old and l initial customers in the system, given the previous departure left i old and k initial customers in the system, and the corresponding interdeparture time $v < V < v+dv$]. $i, k, j, l = 0, 1, 2, \ldots,\ v \geq 0$

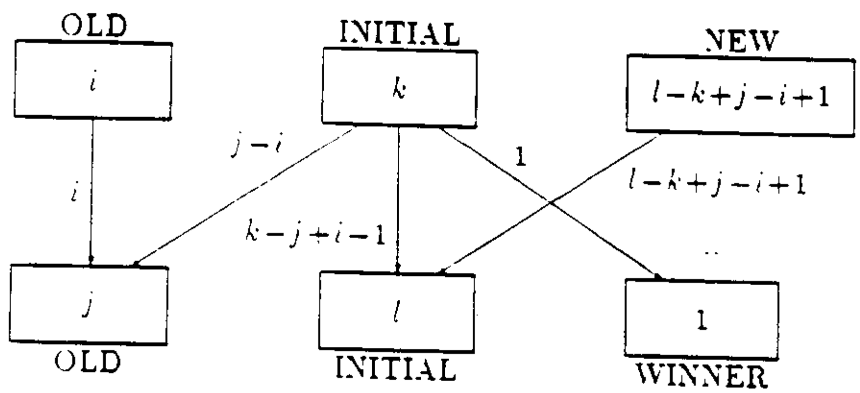
Table 3.10: Definition of $p_{i,k,j,l}(v)$

A transition in the Poisson winner queues with partial restarts occurs when an old customer, an initial customer, or a new customer wins. Figure 3.38 shows the transition graphs for the three different cases.

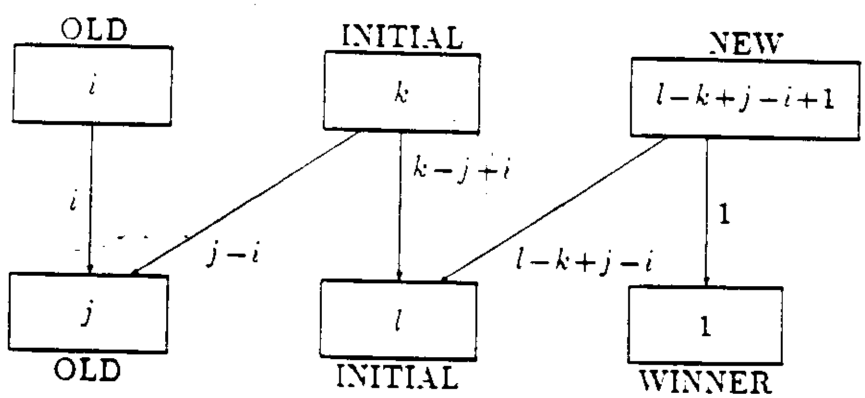
We draw the transition graph a) in Figure 3.38 as follows. The graph represents a transition from state (i,k) to state (j,l). We draw two rows of boxes. The first row is associated with the state (i,k). The first box represents old customers, and we write "i" in it. The second box represents initial customers, and we write "k" in it. The last box represents new customers that arrived after we entered state (i,k) and before the transition to the state (j,l). We leave that box empty for now. The second row of boxes is associated with the state (j,l). In the first box, which represents old customers, we write "j", and in the second box representing initial customers we write "l". In the last box, which represents a winner, we write "1". We now draw arcs from the boxes in the first row to the boxes in the second row. The labels on the arcs represent the number of customers that are transferred from one box to another. Since we know that the winner is an old customer (for the case (a)) we draw an arc labeled "1" from the top old customer box to the winner box. All other old customers remain old, since they cannot become initial, so we draw an arc labeled "$i-1$" from the top old customer box to the bottom old customer box. Next, all other $j-i+1$ old customers in state (j,l) must have come from initial customers which have finished their (first) service before the transition occurs. (Note that new arriving customers cannot become old in the first transition.) Thus we draw an arc la-



a) Old Customer Winner



b) Initial Customer Winner



c) New Customer Winner

Figure 3.38: Poisson Winner Queue with Partial Restarts Transition Graphs

$$(i,k) \longrightarrow (j,l)$$

beled "$j-i+1$" from the top initial customer box to the bottom old customer box. The rest of the initial customers remain initial, and so we draw an arc labeled "$k-j+i-1$" from the top initial customer box to the bottom initial customer box. All other initial customers in state (j,l) must have newly arrived. We thus draw an arc labeled "$l-k+j-i+1$" from the new customer box to the bottom initial customer box. We have now completed drawing arcs since the sum of the labels on the arcs equals the sum of the bottom row of boxes. Now we take the sum of the labels of all the arcs that leave the new customer box, and we write that number in the box, i.e., we write "$l-k+j-i+1$". In the similar way we draw the transition graphs for the cases (b) and (c) in Figure 3.38.

Let us consider again a departure D from the system. Let it leave i old and k initial customers in the system and let the time be equal zero when departure D takes place. In addition to the probabilities P_{OL}, P_{OW}, P_{NL}, and P_{NW} defined in Table 3.5, we define two other probabilities, P_{IL} and P_{IW}, in Table 3.11.

The probability density that an old customer won, all the initial and new customers lost, and the interdeparture time equals v, is given by $W_O(v)=P_{OW}(i,v)\cdot P_{IL}(k,j-i+1,v)P_{NL}(l-k+j-i+1,v)$, as can be clearly observed from part (a) of Figure 3.38. From part (b) of the same figure we see that the probability density that an initial customer won, all the old and new customers lost, and the interdeparture time equals v, is given by $W_I(v)=P_{OL}(i,v)P_{IW}(k,j-i+1,v)P_{NL}(l-k+j-i+1,v)$. From part (c) of Figure 3.38 we see that the probability density that a new customer won, all the old and initial customers lost, and the interdeparture time

Symbol	Definition
$P_{IL}(k, m, v)$	P[m initial customers come before v and finish their next service after v, and $k-m$ initial customers come after v, given k initial customers in the system]
$P_{IW}(k, m, v)$	$\frac{1}{dv}$ P[$m-1$ initial customers come before v and finish their next service after v, and $k-m$ initial customers come after v, and one initial customer comes before v and finishes in the interval $(v, v+dv)$, given k initial customers in the system]

Table 3.11: Definition of P_{IW} and P_{IL}

equals v, is given by $W_N(v) = P_{OL}(i,v)P_{IL}(k,j-i,v)P_{NW}(l-k+j-i+1,v)$. We can now write the following.

$$\begin{aligned} p_{i,k,j,l}(v) &= W_O(v) + W_I(v) + W_N(v) \\ &= P_{OW}(i,v)P_{IL}(k,j-i+1,v)P_{NL}(l-k+j-i+1,v) + \qquad (3.51) \\ &\quad P_{OL}(i,v)P_{IW}(k,j-i+1,v)P_{NL}(l-k+j-i+1,v) + \\ &\quad P_{OL}(i,v)P_{IL}(k,j-i,v)P_{NW}(l-k+j-i+1,v), \\ &\quad i,k,j,l \geq 0 \end{aligned}$$

The expressions for $P_{OL}(i,v)$, $P_{OW}(i,v)$, $P_{NL}(k,v)$, and $P_{NW}(k,v)$ are already given in Section 3.2.3.2, 3.2.3.3, and 3.2.3.4. However, since the distribution of the service times upon restarts differs from the distribution of the initial service times, in the following section we rewrite the formulas obtained for $P_{OL}(i,v)$ and $P_{OW}(i,v)$ with q_r replacing q, p_r replacing p, $r\bar{x}$ replacing $\bar{x}$, and μ/r replacing μ.

In the subsequent sections, the three different parts of the right hand side of the Equation (3.52), $W_O(v)$, $W_I(v)$, and $W_N(v)$ are found separately. We then calculate $p_{i,k,j,l}$ as

$$\begin{aligned} p_{i,k,j,l} &= \int_0^\infty W_O(v)\,dv + \int_0^\infty W_I(v)\,dv + \int_0^\infty W_N(v)\,dv \qquad (3.52) \\ &= W_O + W_I + W_N \end{aligned}$$

3.3.3.2 Rewriting the $P_{OL}(i,v)$ and $P_{OW}(i,v)$

Equations (3.14) and (3.15) give us, for the broadcast redraw systems with a $D_{q_r}M$ distribution of the service times upon restarts, and $\overline{X_r} = r\overline{x}$, the following.

$$P_{OL}(i,v) = \begin{cases} 1, & v < rq_r\overline{x} \text{ or } i=0 \\ e^{-i(\mu v - rq_r)/(rp_r)}, & v \ge rq_r\overline{x},\ i \ge 1 \end{cases} \tag{3.53}$$

$$P_{OW}(i,v) = \begin{cases} 0, & v < rq_r\overline{x} \text{ or } i=0 \\ i\frac{\mu}{rp_r}e^{-i(\mu v/r - q_r)/p_r}, & v \ge rq_r\overline{x},\ i \ge 1 \end{cases} \tag{3.54}$$

Formula (3.17) and (3.18) give us, for the silent redraw systems with memoryless service times upon restarts, and $\overline{X_r} = r\overline{x}$, the following.

$$P_{OL}(i,v) = (1+\mu v/r)^i e^{-i\mu v/r}, \quad i \ge 0,\ v \ge 0 \tag{3.55}$$

$$P_{OW}(i,v) = i(\mu/r)^2 v(1+\mu v/r)^{i-1} e^{-i\mu v/r}, \quad i \ge 0,\ v \ge 0 \tag{3.56}$$

For the silent winner queues with deterministic restarted service times, we will make the assumption analogous to the one in Section 3.2.3.2, i.e., that the arrivals of old customers are memoryless within the time interval $[0, r\overline{x}]$, i.e., they are exponentially distributed but also forced to arrive in $[0, r\overline{x}]$.

$$P_{OL}(i,v) = \begin{cases} 1, & 0 \le v \le \overline{x} \text{ or } i=0 \\ \left(\frac{e^{1-\mu v}-1/e}{1-1/e}\right)^i, & \overline{x} < v \le 2\overline{x}, i \ge 1 \\ 0, & v > 2\overline{x}, i \ge 1 \end{cases} \tag{3.57}$$

$$P_{OW}(i,v) = \begin{cases} \frac{i\mu e^{1-\mu v}}{1-1/e}\left(\frac{e^{1-\mu v}-1/e}{1-1/e}\right)^{i-1}, & \overline{x} < v \le 2\overline{x} \\ 0, & \text{otherwise} \end{cases} \tag{3.58}$$

The probabilities specified in this section can be used for calculating exact transition probabilities in queues M/G-D_{q_r}M(BR,r), M/G-D_{q_r}M(sBR,r), and M/G-M(SR,r), and an approximation for M/G-D(SR,r).

3.3.3.3 Finding the $P_{IL}(k,m,v)$ and $P_{IW}(k,m,v)$

Consider an initial customer left in the system by a departure at time zero. We define U_{in} to be a random variable representing the end of his present unsuccessful service, and $V_{in} = U_{in}+X_r$ to be a random variable representing the time until the end of his restarted service. Let $\mathcal{U}_{in}(v)$ represent the probability distribution function of the random variable U_{in}. Let $\mathcal{V}_{in}(v)$ represent the probability distribution function of the random variable V_{in}. The following holds.

$$
\begin{aligned}
P[V_{in} \le v] &\stackrel{\text{def}}{=} \mathcal{V}_{in}(v) \\
&= \mathcal{U}_{in}(v) \otimes b_r(v) \qquad (3.59)\\
&= \int_0^v \mathcal{U}_{in}(u) b_r(v-u)\, dv
\end{aligned}
$$

$$
\begin{aligned}
P_{IL}(k,m,v) &= \frac{1}{dv}\binom{k}{m} P[U_{in} \le v \wedge V_{in} > v]^m (P[U_{in} > v])^{k-m} \\
&= \binom{k}{m} \{P[V_{in} > v] - P[U_{in} > v]\}^m (P[U_{in} > v])^{k-m} \\
&= \binom{k}{m} [\mathcal{U}_{in}(v) - \mathcal{V}_{in}(v)]^m [1 - \mathcal{U}_{in}(v)]^{k-m} \qquad (3.60)
\end{aligned}
$$

$$
\begin{aligned}
P_{IW}(k,m,v) &= m\binom{k}{m} \frac{1}{dv} P[U_{in} \le v \wedge v < V_{in} < v+dv] \\
&\quad P[U_{in} \le v \wedge V_{in} > v]^{m-1} (P[U_{in} > v])^{k-m} \\
&= m\binom{k}{m} \frac{1}{dv} P[v < V_{in} < v+dv] \{P[V_{in} > v] - P[U_{in} > v]\}^{m-1} \\
&\quad (P[U_{in} > v])^{k-m}
\end{aligned}
$$

$$= \quad m\binom{k}{m}\frac{d\mathcal{V}_{in}(v)}{dv}\left[\mathcal{U}_{in}(v) - \mathcal{V}_{in}(v)\right]^{m-1}\left[1 - \mathcal{U}_{in}(v)\right]^{k-m} \tag{3.61}$$

The restarted service time probability distribution function $B_r(x)$ is given as a parameter of the system. We now only need to find $\mathcal{U}_{in}(u)$ in order to find $P_{IL}(k, m, v)$. For pure broadcast systems $U_{in} = 0$, and thus we have

$$\mathcal{U}_{in}(u) = 1, \quad u \geq 0$$

which gives us, in the broadcast case

$$P_{IL}(k, m, v) = \begin{cases} 1, & k = 0 \\ (P[X_r > v])^m, & m = k \geq 1 \\ 0, & \text{otherwise} \end{cases} \tag{3.62}$$

$$= \begin{cases} 1, & k = 0 \\ [1 - B_r(v)]^m, & m = k \geq 1 \\ 0, & \text{otherwise} \end{cases} \tag{3.63}$$

$$P_{IW}(k, m, v) = \begin{cases} mb_r(v)\,[1 - B_r(v)]^{m-1}, & m = k \geq 1 \\ 0, & \text{otherwise} \end{cases} \tag{3.64}$$

For the broadcast redraw system with a $D_{q_r}M$ restarted service time distribution defined in Equation (2.1), we have

$$P_{IL}(k, m, v) = \begin{cases} 1, & (v \leq rq_r\overline{x}) \text{ or } (k = 0) \\ e^{-m(\mu v - rq_r)/(rp_r)}, & v > q\overline{x},\ m = k \geq 1 \\ 0, & \text{otherwise} \end{cases} \tag{3.65}$$

$$P_{IW}(k,m,v) = \begin{cases} m\dfrac{\mu}{rp_r}e^{-m(\mu v - rq_r)/(rp_r)}, & v > rq_r\overline{x},\ m = k \geq 1 \\ 0, & \text{otherwise} \end{cases} \tag{3.66}$$

where μ is defined as $\mu \stackrel{\text{def}}{=} 1/\overline{x}$.

For the silent and silent/broadcast systems with memoryless initial service times we have

$$\mathcal{U}_{in}(u) = B(u) = 1 - e^{-\mu u} \tag{3.67}$$

For the silent and silent/broadcast redraw systems with memoryless initial service times and $D_{q_r}M$ distributed restarted service times from Equation (2.1), (3.67), and (3.59) we get

$$\mathcal{V}_{in}(v) = \begin{cases} 0, & v \leq rq_r\overline{x} \\ 1 - \dfrac{e^{-(\mu v - rq_r)/(rp_r)} - rp_r e^{-(\mu v - rq_r)}}{rp_r - 1}, & v > rq_r\overline{x} \end{cases} \tag{3.68}$$

and thus, using Equation (3.60) and (3.61)

$$P_{IL}(k,m,v) = \begin{cases} 1, & k = 0 \\ \left\{\dfrac{rp_r e^{-(\mu v - rq_r)} - rp_r e^{-\mu v} - e^{-(\mu v - rq_r)/(rp_r)} - e^{-\mu v}}{rp_r - 1}\right\}^m & \\ \cdot \binom{k}{m} e^{-(k-m)\mu v}, \ v > rq_r\overline{x},\ k \geq 1 & \\ 0, & \text{otherwise} \end{cases}$$

(3.69)

$$P_{IW}(k,m,v) = \begin{cases} \binom{k}{m} \dfrac{\mu(rp_r)^2 e^{-(\mu v - rq_r)} - \mu e^{-(\mu v - rq_r)/(rp_r)}}{rp_r(rp_r - 1)} e^{-(k-m)\mu v} \\ \quad \cdot \left\{ \dfrac{rp_r e^{-(\mu v - rq_r)} - rp_r e^{-\mu v} - e^{-(\mu v - rq_r)/(rp_r)} - e^{-\mu v}}{rp_r - 1} \right\}^{m-1}, \\ \qquad v > rq_r\overline{x},\ k \geq 1 \\ 0, \qquad \text{otherwise} \end{cases} \tag{3.70}$$

For the silent winner queues with deterministic initial service times, we assume that the arrival of old customers is memoryless within the time interval $[0, \overline{x}]$, i.e., they are exponentially distributed but also forced to arrive in $[0, \overline{x}]$. This gives us the following approximate expression for $\mathcal{U}_{in}(u)$.

$$\mathcal{U}_{in}(u) = \begin{cases} \dfrac{1 - e^{-\mu u}}{1 - 1/e}, & 0 \leq u \leq q\overline{x} \\ 1, & u > q\overline{x} \end{cases} \tag{3.71}$$

For the silent and silent/broadcast redraw systems with deterministic initial service times and $D_{q_r}M$ distributed restarted service times from Equation (2.1), (3.67), and (3.59) we get the following approximation for $\mathcal{V}_{in}(v)$.

$$\mathcal{V}_{in}(v) = \begin{cases} 0, & 0 \leq v \leq rq_r\overline{x} \\ \dfrac{1}{1-1/e} - \dfrac{e^{-(\mu v - rq_r)} - rp_r e^{-(\mu v - rq_r)/(rp_r)}}{(1-1/e)(1-rp_r)}, & rq_r\overline{x} < v \leq \overline{x} + rq_r\overline{x} \\ 1 - \left[e^{-1+1/(rp_r)} - 1\right] \dfrac{rp_r e^{-(\mu v - rq_r)/(rp_r)}}{(1-1/e)(1-rp_r)}, & v > \overline{x} + rq_r\overline{x} \end{cases} \tag{3.72}$$

and thus, from Equation (3.60) and (3.61) we have

$$P_{IL}(k,m,v)_{rq_r\leq q} = \begin{cases} \left(\dfrac{1-e^{-\mu v}}{1-1/e}\right)^k, \quad m=k,\ 0\leq v\leq rq_r\overline{x} \\ \left[\dfrac{e^{-\mu v}(e^{rq_r}-1+rp_r)-rp_r e^{-(\mu v-rq_r)/(rp_r)}}{(1-1/e)(1-rp_r)}\right]^k, \\ \qquad m=k,\ rq_r\overline{x}<v\leq\overline{x} \\ \left[\dfrac{e^{-\mu v-rq_r}-(1-rp_r)/e-rp_r e^{-(\mu v-rq_r)/(rp_r)}}{(1-1/e)(1-rp_r)}\right]^k, \\ \qquad m=k,\ \overline{x}<v\leq\overline{x}+rq_r\overline{x} \\ \left[\dfrac{rp_r\left(e^{(1-rp_r)/(rp_r)}-1\right)}{(1-1/e)(1-rp_r)}\right]^k e^{-k(\mu v-rq_r)/(rp_r)}, \\ \qquad m=k,\ v>\overline{x}+rq_r\overline{x} \\ 0, \qquad \text{otherwise} \end{cases} \tag{3.73}$$

$$P_{IL}(k,m,v)_{rq_r>1} = \begin{cases} \left(\dfrac{1-e^{-\mu v}}{1-1/e}\right)^k, \quad m=k,\ 0\leq v\leq\overline{x} \\ 1, \qquad \overline{x}<v\leq rq_r\overline{x} \\ \left[\dfrac{e^{-\mu v-rq_r}-(1-rp_r)/e-rp_r e^{-(\mu v-rq_r)/(rp_r)}}{(1-1/e)(1-rp_r)}\right]^k, \\ \qquad m=k,\ rq_r\overline{x}<v\leq\overline{x}+rq_r\overline{x} \\ \left[\dfrac{rp_r\left(e^{(1-rp_r)/(rp_r)}-1\right)}{(1-1/e)(1-rp_r)}\right]^k e^{-k(\mu v-rq_r)/(rp_r)}, \\ \qquad m=k,\ v>\overline{x}+rq_r\overline{x} \\ 0, \qquad \text{otherwise} \end{cases} \tag{3.74}$$

$$P_{IW}(k,m,v)_{rq_r\le 1} = \begin{cases} k\mu\,\dfrac{e^{-(\mu v-rq_r)}-e^{-(\mu v-rq_r)/(rp_r)}}{(1-1/e)^k(1-rp_r)^k} \\ \cdot\left[e^{-\mu v}(e^{rq_r}-1+rp_r)-rp_r e^{-(\mu v-rq_r)/(rp_r)}\right]^{k-1}, \\ \qquad m=k,\ rq_r\overline{x}<v\le\overline{x} \\ k\mu\,\dfrac{e^{-(\mu v-rq_r)}-e^{-(\mu v-rq_r)/(rp_r)}}{(1-1/e)^k(1-rp_r)^k} \\ \cdot\left[e^{-\mu v-rq_r}-(1-rp_r)/e-rp_r e^{-(\mu v-rq_r)/(rp_r)}\right]^{k-1}, \\ \qquad m=k,\ \overline{x}<v\le\overline{x}+rq_r\overline{x} \\ k\mu\,\dfrac{(rp_r)^{k-1}\left(e^{(1-rp_r)/(rp_r)}-1\right)^k}{(1-1/e)^k(1-rp_r)^k} \\ \cdot\, e^{-k(\mu v-rq_r)/(rp_r)},\quad m=k, v>\overline{x}+rq_r\overline{x} \\ 0, \qquad \text{otherwise} \end{cases} \tag{3.75}$$

$$P_{IW}(k,m,v)_{rq_r>1} = \begin{cases} \dfrac{k\mu\left[e^{-(\mu v-rq_r)}-e^{-(\mu v-rq_r)/(rp_r)}\right]}{(1-1/e)^k(1-rp_r)^k} \\ \cdot\left[e^{-\mu v-rq_r}-(1-rp_r)/e-rp_r e^{-(\mu v-rq_r)/(rp_r)}\right]^{k-1}, \\ \qquad m=k,\ rq_r\overline{x}<v\le\overline{x}+rq_r\overline{x} \\ k\mu\,\dfrac{(rp_r)^{k-1}\left(e^{(1-rp_r)/(rp_r)}-1\right)^k}{(1-1/e)^k(1-rp_r)^k} \\ \cdot\, e^{-k(\mu v-rq_r)/(rp_r)},\quad m=k,\ v>\overline{x}+rq_r\overline{x} \\ 0, \qquad \text{otherwise} \end{cases} \tag{3.76}$$

In this section we found the exact values for the probabilities $P_{IL}(k,m,v)$ and $P_{IW}(k,m,v)$ that can be used for M/D_qM-D_{q_r}M(BR,r), M/M-D_{q_r}M(sBR,r), M/M(SR,r), and M/M-D(SR,r) systems, and approximations which can be used for systems M/D-D_q,M(sBR,r), M/D-M(SR,r), and M/D(S,r).

We point out that for other winner queues with partial restarts we have not yet found the probability distribution of the random variable U_{in}, and so, we are unable to find $P_{IL}(k,m,v)$ and $P_{IW}(k,m,v)$. For all noredraw systems, except for the approximation M/D(S,r), not only don't have the distribution of U_{in}, but we also don't have the distribution of random variable X_r since, in general, it differs from the given initial service time distribution, i.e., $b_r(x) \neq B_r(x)$.

3.3.3.4 Use of the $P_{NL}(k,v)$ and $P_{NW}(k,v)$

Equation (3.26) through (3.37) are valid for Poisson winner queues with partial restarts as well as for the simple Poisson winner queues. For convenience, we rewrite them here as Equation (3.77) through (3.82). For the D_qM service time probability

$$P_{NL}(k,v) = \begin{cases} \dfrac{(\lambda v)^k}{k!}e^{-\lambda v}, & v \leq q\overline{x} \\ \dfrac{\rho^k[1-pe^{-(\mu v-q)/p}]^k}{k!}e^{-\lambda v}, & v > q\overline{x} \end{cases} \tag{3.77}$$

$$P_{NW}(k,v) = \begin{cases} \lambda[1-e^{-(\mu v-q)/p}]\dfrac{\rho^{k-1}[1-pe^{-(\mu v-q)/p}]^{k-1}}{(k-1)!}e^{-\lambda v}, \\ \qquad v > q\overline{x}, k \geq 1 \\ 0, \qquad \text{otherwise} \end{cases} \tag{3.78}$$

For memoryless initial service times

$$P_{NL}(k,v) = \frac{\rho^k(1-e^{-\mu v})^k}{k!}e^{-\lambda v}, \quad v \geq 0 \tag{3.79}$$

$$P_{NW}(k,v) = \begin{cases} \lambda \dfrac{\rho^{k-1}(1-e^{-\mu v})^k}{(k-1)!} e^{-\lambda v}, & v \geq 0, k \geq 1 \\ 0, & \text{otherwise} \end{cases} \quad (3.80)$$

For deterministic initial service times

$$P_{NL}(k,v) = \begin{cases} \dfrac{(\lambda v)^k}{k!} e^{-\lambda v}, & v \leq \overline{x} \\ \dfrac{\rho^k}{k!} e^{-\lambda v}, & v > \overline{x} \end{cases} \quad (3.81)$$

$$P_{NW}(k,v) = \begin{cases} \lambda \dfrac{\rho^{k-1}}{(k-1)!} e^{-\lambda v}, & v > \overline{x}, k \geq 1 \\ 0, & \text{otherwise} \end{cases} \quad (3.82)$$

3.3.3.5 Derivation of Results in Short

The summary of notation used in this analysis is given in Table 3.12.

Figure 3.39 shows the numerical process of obtaining the performance values T_n and P, starting with given densities $b(x)$ and $b_R(x)$ at the bottom and ending with T_n and P at the top of the *derivation graph.*

3.3.4 System P1: M/M(SR,r) with Exact $p_{i,k,j,l}$'s

We first substitute Equation (3.55), (3.56), (3.69), (3.70), (3.79), and (3.80) into (3.52). Then, after integration, according to Equation (3.52), we get the following expressions for the M/M(SR,r) system. [2]

[2] For convenience, a field in a domain table is left blank if it has the same contents as the field above it.

Function	Defined in
$b(x)$	Section 2.1
$b_R(x)$	Section 2.1
$\mathcal{U}_{in}(u)$	Section 3.3.3.3
$\mathcal{V}_{in}(u)$	Section 3.3.3.3
$\mathcal{U}_{old}(u)$	Section 3.2.3.2
$\mathcal{V}_{old}(u)$	Section 3.2.3.2
$\gamma(x)$	Equation (3.22)
$\gamma_0(x)$	Equation (3.25)
P_{OL}	Table 3.5
P_{OW}	Table 3.5
P_{IL}	Table 3.11
P_{IW}	Table 3.11
P_{NL}	Table 3.5
P_{NW}	Table 3.5
$p_{i,k,j,l}(v)$	Table 3.10
$p_{i,k,j,l}$	Table 3.9
$d_{m,n}$	Section 3.3.3
d_k	Section 3.2.3
T_n	Section 3.2.3
P	Section 3.2.3

Table 3.12: Notation for Poisson Winner Queues with Partial Restarts

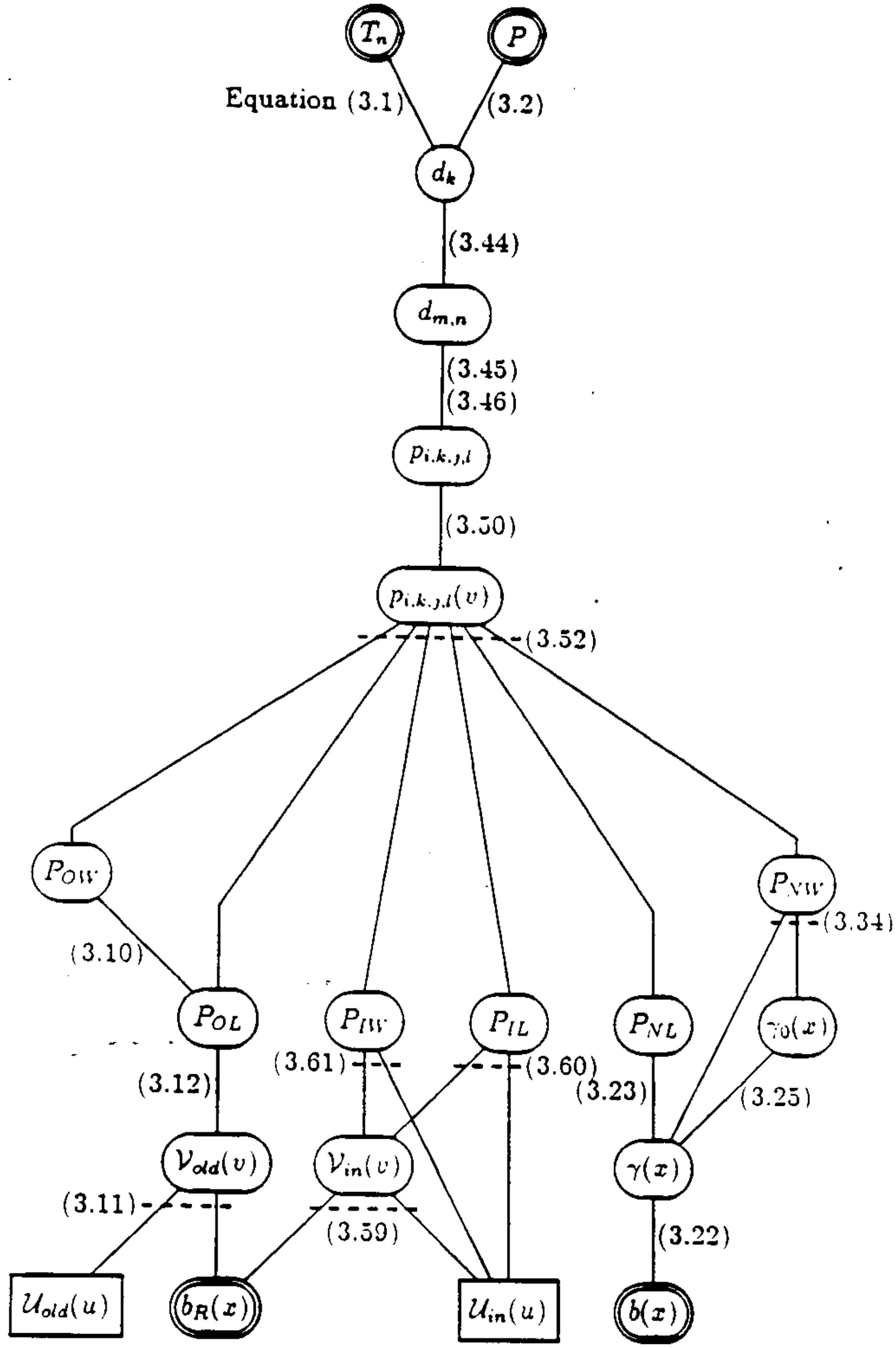


Figure 3.39: Derivation Graph for Winner Queues with Partial Restarts

Domains for W_O in Equation (3.83)					
1	$r=0$	$i \geq 1$	$k \geq j-i+1$	$j=i-1$	$l \geq k-j+i-1$
2	$0<r \neq 1$			$j \geq i-1$	
3	$r=1$				
4	otherwise				

<u>Domain</u>

$$1) \quad W_O = 1$$

$$2) \quad W_O = \frac{i \binom{k}{j-i+1} \rho^{l-k+j-i+1}}{r^2 (l-k+j-i+1)! (1/r-1)^{j-i+1}} \sum_{h=0}^{i-1} \binom{i-1}{h} (1/r)^h (h+1)! \cdot \sum_{n=0}^{j-i+1} \binom{j-i+1}{n} (-1)^n \sum_{m=0}^{l-k+j-i+1} \frac{\binom{l-k+j-i+1}{m} (-1)^m}{[\rho+k+m-n+(i+n)/r]^{h+2}}$$

$$3) \quad W_O = \frac{i \binom{k}{j-i+1} \rho^{l-k+j-i+1}}{(l-k+j-i+1)!} \sum_{h=0}^{i-1} \binom{i-1}{h} (h+j-i+2)! \cdot \sum_{m=0}^{l-k+j-i+1} \frac{\binom{l-k+j-i+1}{m} (-1)^m}{(\rho+k+m+i)^{h+j-i+3}}$$

$$4) \quad W_O = 0 \tag{3.83}$$

Domains for W_I in Equation (3.84)					
1	$r=0$	$i=0$	$k \geq j-i+1$	$j=i$	$l \geq k-j+i-1$
2	$0<r \neq 1$	$i \geq 0$		$j \geq i-1$	
3	$r=1$				
4	otherwise				

Domain

1) $W_I = \frac{k\,\rho^{l-k+1}}{(l-k+1)!} \sum_{m=0}^{l-k+1} \frac{\binom{l-k+1}{m}(-1)^m}{\rho+k+m}$

2) $W_I = \frac{(j-i+1)\binom{k}{j-i+1}\rho^{l-k+j-i+1}}{r\,(l-k+j-i+1)!\,(1/r-1)^{j-i+1}} \sum_{h=0}^{i} \binom{i}{h}(1/r)^h\,h!$
$\cdot \sum_{n=0}^{j-i+1} \binom{j-i+1}{n}(-1)^n \sum_{m=0}^{l-k+j-i+1} \frac{\binom{l-k+j-i+1}{m}(-1)^m}{[\rho+k+m-n+(i+n)/r]^{h+1}}$

3) $W_I = \frac{(j-i+1)\binom{k}{j-i+1}\rho^{l-k+j-i+1}}{(l-k+j-i+1)!} \sum_{h=0}^{i} \binom{i}{h}(h+j-i+1)!$
$\cdot \sum_{m=0}^{l-k+j-i+1} \frac{\binom{l-k+j-i+1}{m}(-1)^m}{(\rho+k+m+i)^{h+j-i+2}}$

4) $W_I = 0$ (3.84)

Domains for W_N in Equation (3.85)					
1	$r=0$	$i=0$	$k\geq j-i$	$j=i$	$l\geq k-j+i$
2	$0<r\neq 1$	$i\geq 0$		$j\geq i$	
3	$r=1$				
4	otherwise				

Domain

1) $W_N = \frac{\rho^{l-k+1}}{(l-k)!} \sum_{m=0}^{l-k+1} \frac{\binom{l-k+1}{m}(-1)^m}{\rho+k+m}$

2) $W_N = \frac{\binom{k}{j-i}\rho^{l-k+j-i+1}}{(l-k+j-i)!\,(1/r-1)^{j-i}} \sum_{h=0}^{i} \binom{i}{h}(1/r)^h\,h!$
$\cdot \sum_{n=0}^{j-i} \binom{j-i}{n}(-1)^n \sum_{m=0}^{l-k+j-i+1} \frac{\binom{l-k+j-i+1}{m}(-1)^m}{[\rho+k+m-n+(i+n)/r]^{h+1}}$

$$
\begin{aligned}
3) \quad W_N &= \frac{\binom{k}{j-i}\,\rho^{l-k+j-i+1}}{(l-k+j-i)!} \sum_{h=0}^{i} \binom{i}{h}\,(h+j-i)! \\
&\quad \cdot \sum_{m=0}^{l-k+j-i+1} \frac{\binom{l-k+j-i+1}{m}\,(-1)^m}{(\rho+k+m+i)^{h+j-i+1}} \\
4) \quad W_N &= 0
\end{aligned}
\tag{3.85}
$$

Following the process of numerical calculation depicted in Figure 3.39, we calculate the normalized power P and normalized response time T_n versus load ρ in Figure 3.40 and 3.41. In the same figures we also show the simulation results for M/M(SR,r) given previously as one of the curves in Figure 3.22.

The performance of the system gets worse with an increase of the restart-to-initial ratio r. Note that the results match the simulation results very closely. The numerical error causes the artifact of rising power curves and falling response time curves for greater values of r and ρ.

3.3.5 System P2: M/M-D(SR,r) with Approximate $p_{i,k,j,l}$'s

We first substitute Equation (3.57), (3.58), (3.69), (3.79), and (3.80) into (3.52). Then, after integration according to Equation (3.52) we get the following expressions for the M/M-D(SR,r) system. [3]

[3] For convenience, a field in a domain table is left blank if it has the same contents as the field above it.

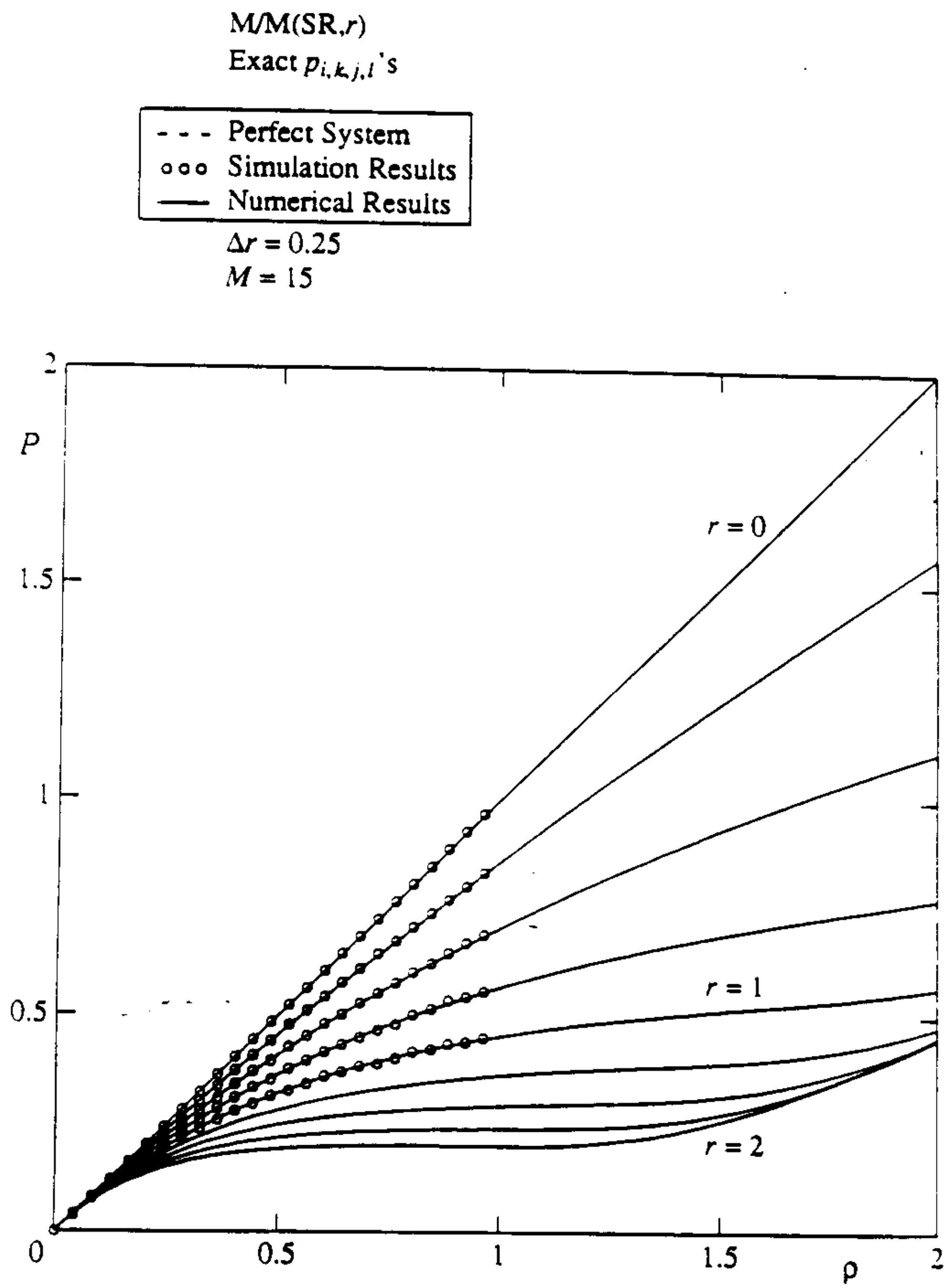


Figure 3.40: Normalized Power for M/M(SR,r)

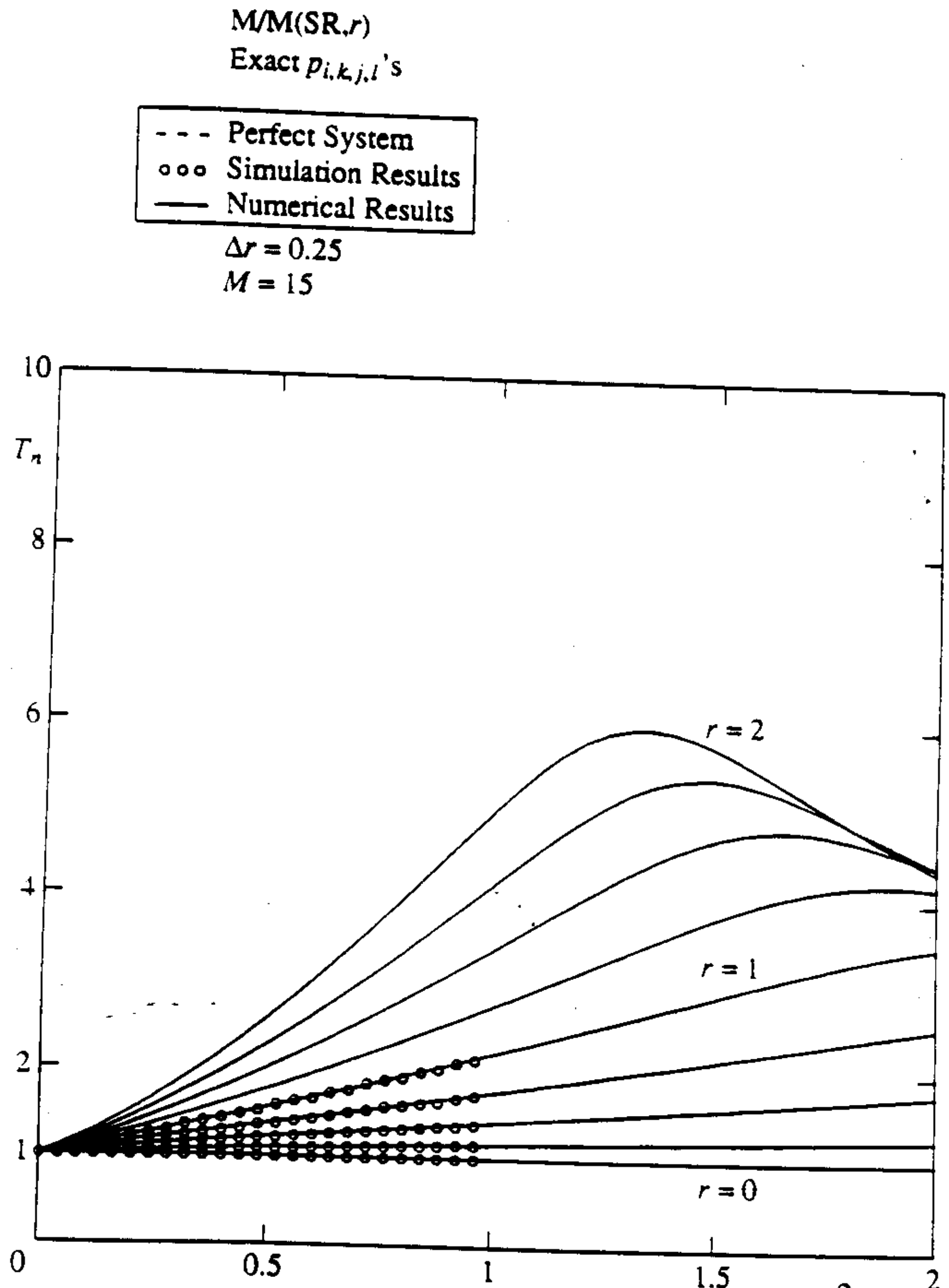


Figure 3.41: Normalized Average Response Time for M/M(SR,r)

Domains for W_O in Equation (3.86)					
1	$r=0$	$i \geq 1$	$k \geq j-i+1$	$j=i-1$	$l=k$
2	$r>0$			$j \geq i-1$	$l \geq k-j+i-1$
3	otherwise				

Domain

$$1) \quad W_O = 1$$

$$2) \quad W_O = i \binom{k}{j-i+1} \frac{\rho^{l-k+j-i+1} (e^r-1)^{j-i+1}}{r\,(l-k+j-i+1)!\,(1-e)^i} \cdot \sum_{h=0}^{i-1} \binom{i-1}{h} (-e^2)^{h+1} \sum_{m=0}^{l-k+j-i+1} \binom{l-k+j-i+1}{m} (-1)^m \cdot \frac{e^{-r(\rho+k+m+(h+1)/r)} - e^{-2r(\rho+k+m+(h+1)/r)}}{\rho+k+m+(h+1)/r}$$

$$3) \quad W_O = 0 \tag{3.86}$$

Domains for W_I in Equation (3.87)					
1	$r=0$	$i=0$	$k \geq j-i+1$	$j=i$	$l \geq k-j+i-1$
2	$r>0$			$j \geq i-1$	
3		$i \geq 1$			
4	otherwise				

Domain

$$1) \quad W_I = \frac{k\,\rho^{l-k+1}}{(l-k+1)!} \sum_{m=0}^{l-k+1} \frac{\binom{l-k+1}{m} (-1)^m}{\rho+k+m}$$

2) $$W_I = (j-i+1)\binom{k}{j-i+1}\frac{\rho^{l-k+j-i+1}\, e^r(e^r-1)^{j-i}}{(l-k+j-i+1)!} \cdot \sum_{m=0}^{l-k+j-i+1}\binom{l-k+j-i+1}{m}(-1)^m\frac{e^{-r(\rho+k+m)}}{\rho+k+m}$$

3) $$W_I = (j-i+1)\binom{k}{j-i+1}\frac{\rho^{l-k+j-i+1}\, e^r(e^r-1)^{j-i}}{(l-k+j-i+1)!\,(1-e)^i} \cdot \sum_{h=0}^{i}\binom{i}{h}(-e^2)^h \sum_{m=0}^{l-k+j-i+1}\binom{l-k+j-i+1}{m}(-1)^m \cdot \frac{e^{-r(\rho+k+m+h/r)} - e^{-2r(\rho+k+m+h/r)}}{\rho+k+m+h/r}$$

4) $$W_I = 0 \tag{3.87}$$

Domains for W_N in Equation (3.89)					
1	$r=0$	$i=0$	$k \geq j-i$	$j=i$	$l \geq k-j+i$
2	$r>0$			$j \geq i$	
3		$i \geq 1$			
4	otherwise				

Domain

1) $$W_N = \frac{\rho^{l-k+1}}{(l-k)!}\sum_{m=0}^{l-k+1}\frac{\binom{l-k+1}{m}(-1)^m}{\rho+k+m}$$

2) $$W_N = \binom{k}{j-i}\frac{\rho^{l-k+j-i+1}}{(l-k+j-i)!}\left\{\sum_{g=0}^{l-k+2j-2i+1}\binom{l-k+2j-2i+1}{g}(-1)^g \cdot \frac{1-e^{-r(\rho+k-j+i+g)}}{\rho+k-j+i+g} + (e^r-1)^{j-i} \cdot \sum_{m=0}^{l-k+j-i+1}\binom{l-k+j-i+1}{m}(-1)^m\frac{e^{-r(\rho+k+m)}}{\rho+k+m}\right\}$$

$$
\begin{aligned}
3)\quad W_N &= \binom{k}{j-i} \frac{\rho^{l-k+j-i+1}}{(l-k+j-i)!} \left\{ \sum_{g=0}^{l-k+2j-2i+1} \binom{l-k+2j-2i+1}{g} (-1)^g \right. \\
&\quad \cdot \frac{1-e^{-r(\rho+k-j+i+g)}}{\rho+k-j+i+g} + \frac{(e^r-1)^{j-i}}{(1-e)^i} \sum_{h=0}^{i} \binom{i}{h} (-e^2)^h \\
&\quad \cdot \left. \sum_{m=0}^{l-k+j-i+1} \binom{l-k+j-i+1}{m} (-1)^m \frac{e^{-r(\rho+k+m+h/r)} - e^{-2r(\rho+k+m+h/r)}}{\rho+k+m+h/r} \right\} \\
4)\quad W_N &= 0
\end{aligned}
\tag{3.88}
$$

Following the process of numerical calculation depicted in Figure 3.39, we calculate the normalized power P and normalized response time T_n versus load ρ in Figure 3.42 and 3.43.

The performance of the system gets worse with an increase of the restart-to-initial ratio r. The system performs better than M/M(SR,r). Note that the results match the simulation results very closely. The numerical error causes the artifact of rising power curves and falling response time curves for greater values of r and ρ.

3.3.6 System P3: M/D-M(SR,r) with Approximate $p_{i,k,j,l}$'s

We first substitute Equation (3.55), (3.56), (3.73) or (3.74), (3.75) or (3.76), (3.81), and (3.82) into (3.52). Then, after the integration according to Equation (3.52) we get the following expressions for the M/D-M(SR,r) system. [4]

[4] For convenience, a field in a domain table is left blank if it has the same contents as the field above it.

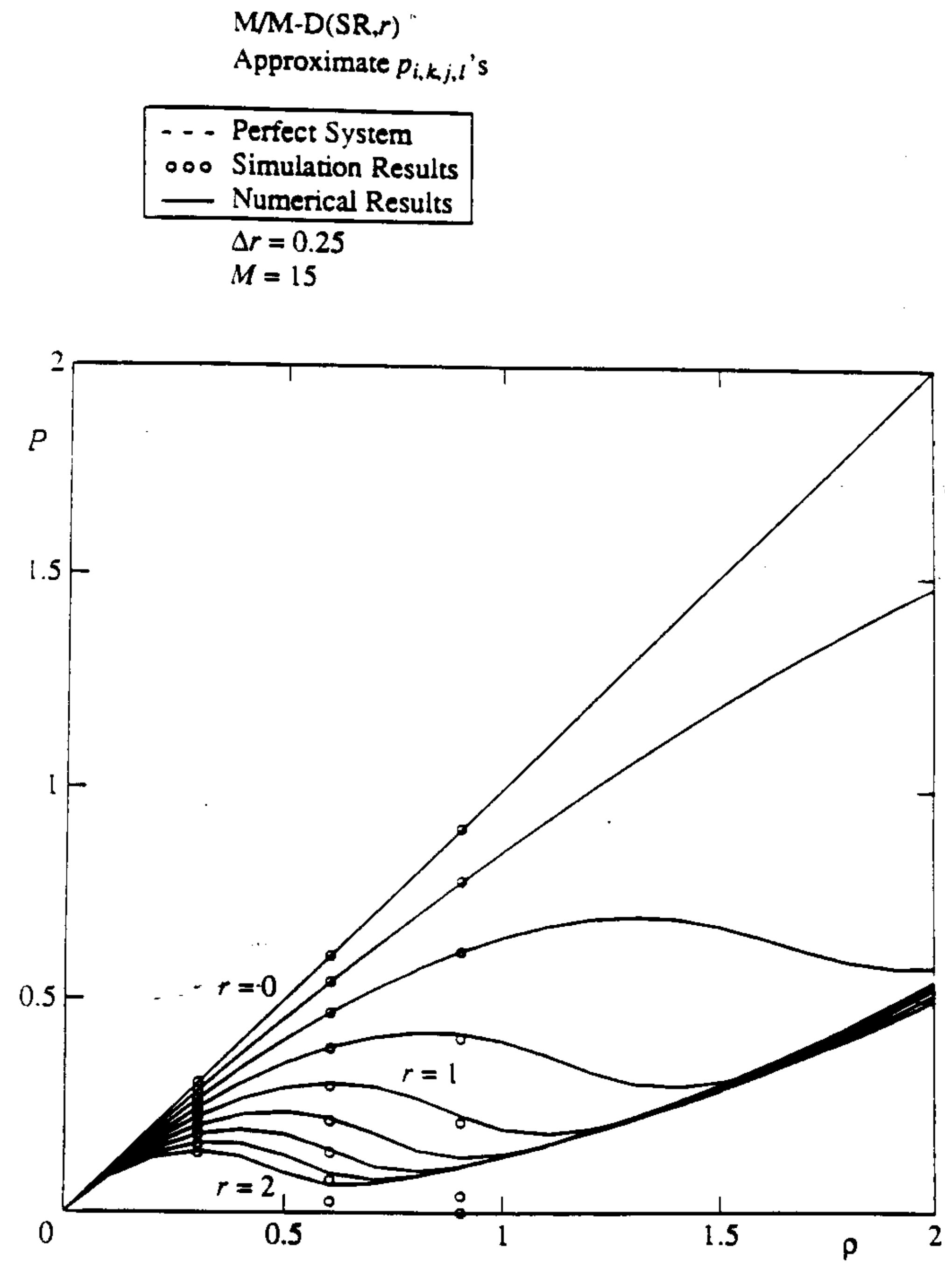


Figure 3.42: Normalized Power for M/M-D(SR,r)

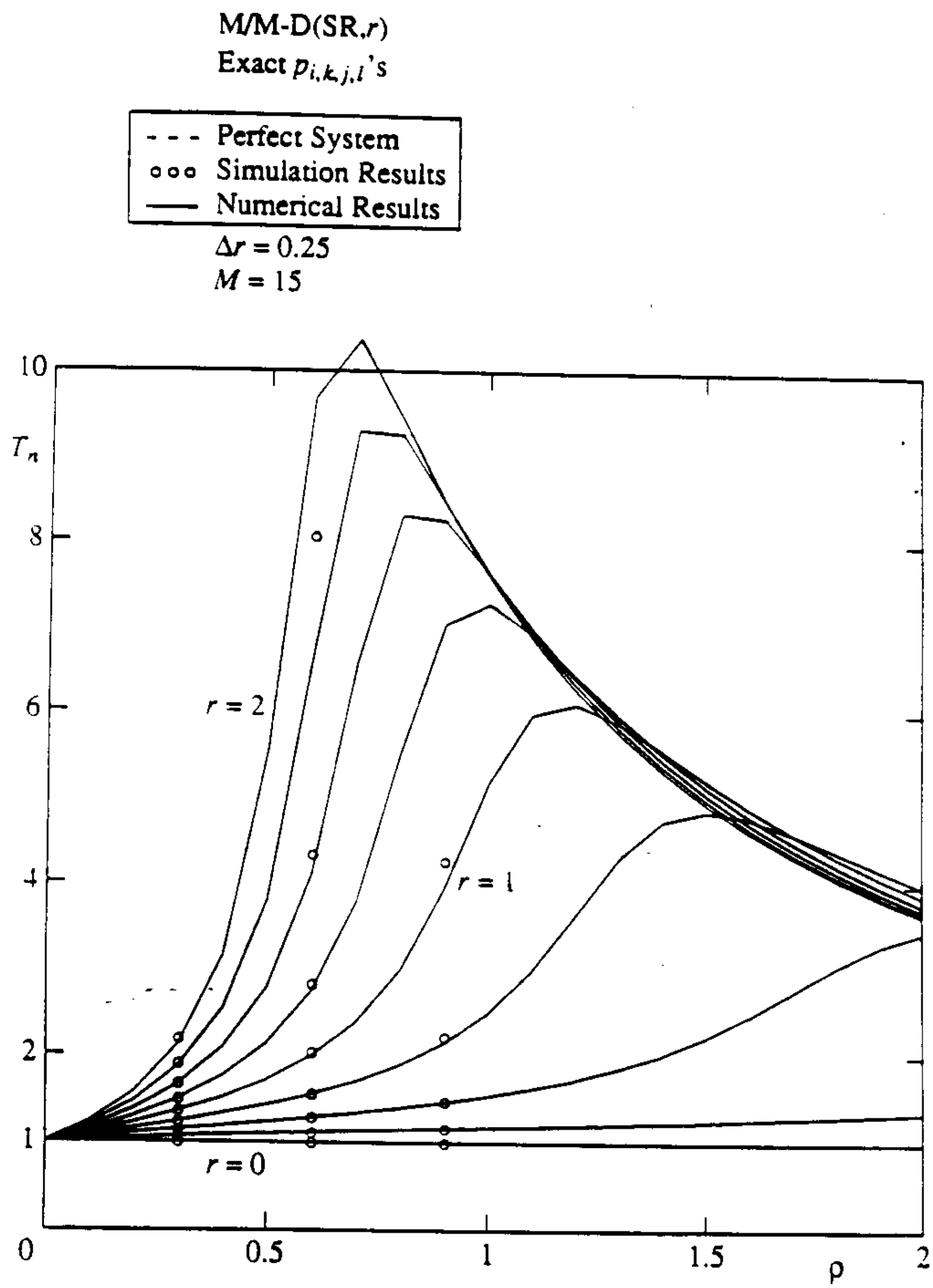


Figure 3.43: Normalized Average Response Time for M/M-D(SR,r)

Domains for W_O in Equation (3.89)					
1	$r=0$	$i\geq 1$	$k\geq j-i+1$	$j=i-1$	$l=k-j+i-1$
2	$0<r\neq 1$		$k=j-i+1$	$j\geq i-1$	$l\geq k-j+i-1$
3			$k>j-i+1$		
4	$r=1$		$k=j-i+1$		
5			$k>j-i+1$		
6	otherwise				

Domain

1) $W_O = 1$

2)
$$W_O = \frac{i\ \rho^l}{l!\ (1-1/e)^k (1-1/r)^k} \sum_{g=0}^{i-1} \frac{\binom{i-1}{g}}{r^{g+2}} \left\{ \sum_{n=0}^{k} \binom{k}{n} (-1)^{k-n}\ (l+g+1)! \right.$$
$$\cdot \left[\frac{1}{[\rho+k-n+(i+n)/r]^{l+g+2}} - e^{-[\rho+k-n+(i+n)/r]} \right.$$
$$\left. \cdot \sum_{b=0}^{l+g+1} \frac{1}{b!\ [\rho+k-n+(i+n)/r]^{l+g+2-b}} \right] + (1-e^{1/r-1})^k\ (g+1)!$$
$$\left. \cdot\ e^{-[\rho+(j+1)/r]} \sum_{b=0}^{g+1} \frac{1}{b!\ [\rho+(j+1)/r]^{g+2-b}} \right\}$$

3)
$$W_O = \frac{i\ \binom{k}{j-i+1}\ \rho^{l-k+j-i+1}}{(l-k+j-i+1)!\ (e-1)^k (1-1/r)^{j-i+1}} \sum_{g=0}^{i-1} \frac{\binom{i-1}{g}}{r^{g+2}} \sum_{n=0}^{j-i+1} \binom{j-i+1}{n}$$
$$\cdot \sum_{m=0}^{k-j+i-1} \binom{k-j+i-1}{m} (-1)^{k-n-m}\ e^{j-i+1+m}\ (l-k+j-i+g+2)!$$
$$\cdot \left\{ \frac{1}{[\rho+j-i+1-n+m+(i+n)/r]^{l-k+j-i+g+3}} \right.$$
$$-\ e^{-[\rho+j-i+1-n+m+(i+n)/r]}$$
$$\left. \cdot \sum_{b=0}^{l-k+j-i+g+2} \frac{1}{b!\ [\rho+j-i+1-n+m+(i+n)/r]^{l-k+j-i+g+3-b}} \right\}$$

4) $$W_O = \frac{i\ \rho^l}{l!\ (1-1/e)^k} \sum_{g=0}^{i-1} \binom{i-1}{g} \left\{ (l+k+g+1)! \left[\frac{1}{(\rho+j+1)^{l+k+g+2}} - e^{-(\rho+j+1)} \sum_{b=0}^{l+k+g+1} \frac{1}{b!\ (\rho+j+1)^{l+k+g+2-b}} \right] + e^{-(\rho+j+1)}\ (g+1)! \sum_{b=0}^{g+1} \frac{1}{b!\ (\rho+j+1)^{g+2-b}} \right\}$$

5) $$W_O = \frac{i \binom{k}{j-i+1} \rho^{l-k+j-i+1}}{(l-k+j-i+1)!\ (1-1/e)^k} \sum_{g=0}^{i-1} \binom{i-1}{g} \sum_{n=0}^{k-j+i-1} \binom{k-j+i-1}{n} \cdot (-1)^{k-j+i-1-n}\ (l-k+2j-2i+g+3)! \cdot \left\{ \frac{1}{(\rho+n+j+1)^{l-k+2j-2i+g+4}} - e^{-(\rho+n+j+1)} \cdot \sum_{b=0}^{l-k+2j-2i+g+3} \frac{1}{b!\ (\rho+n+j+1)^{l-k+2j-2i+g+4-b}} \right\}$$

6) $$W_O = 0 \qquad (3.89)$$

Domains for W_I in Equation (3.90)					
1	$r=0$	$i=0$	$k \geq j-i+1$	$j=i$	$l \geq k-j+i-1$
2	$0<r\neq 1$	$i \geq 0$	$k=j-i+1$	$j \geq i$	
3			$k>j-i+1$		
4	$r=1$		$k=j-i+1$		
5			$k>j-i+1$		
6	otherwise				

<u>Domain</u>

1) $$W_I = \frac{k\ \rho^l}{l!\ (1-1/e)^k} \sum_{g=0}^{k-1} \binom{k-1}{g} (-1/e)^{k-1-g} \frac{1-e^{-(\rho+g+1)}}{\rho+g+1}$$

$$
\begin{aligned}
2)\quad W_I \;=\;& \frac{k\,\rho^l}{l!\,(1-1/e)^k(1-1/r)^k}\left\{\sum_{g=0}^{i}\frac{\binom{i}{g}}{r^{g+1}}\sum_{n=0}^{k}\binom{k}{n}(-1)^{k-n}\,(l+g)!\right.\\
&\cdot\left[\frac{1}{[\rho+k-n+(i+n)/r]^{l+g+1}} - e^{-[\rho+k-n+(i+n)/r]}\right.\\
&\left.\cdot\sum_{b=0}^{l+g}\frac{1}{b!\,[\rho+k-n+(i+n)/r]^{l+g+1-b}}\right] + (1-e^{1/r-1})^k\\
&\left.\cdot\sum_{g=0}^{i}\frac{\binom{i}{g}}{r^{g+1}}\,e^{-[\rho+(j+1)/r]}\sum_{b=0}^{g}\frac{1}{b!\,[\rho+(j+1)/r]^{g+1-b}}\right\}
\end{aligned}
$$

$$
\begin{aligned}
3)\quad W_I \;=\;& \frac{(j-i+1)\,\binom{k}{j-i+1}\,\rho^{l-k+j-i+1}}{(l-k+j-i+1)!\,(e-1)^k(1-1/r)^{j-i+1}}\sum_{g=0}^{i}\frac{\binom{i}{g}}{r^{g+1}}\sum_{n=0}^{j-i+1}\binom{j-i+1}{n}\\
&\cdot\sum_{m=0}^{k-j+i-1}\binom{k-j+i-1}{m}(-1)^{k-n-m}\,e^{j-i+1+m}\,(l-k+j-i+g+1)!\\
&\cdot\left\{\frac{1}{[\rho+j-i+1-n+m+(i+n)/r]^{l-k+j-i+g+2}}\right.\\
&- e^{-[\rho+j-i+1-n+m+(i+n)/r]}\\
&\left.\cdot\sum_{b=0}^{l-k+j-i+g+1}\frac{1}{b!\,[\rho+j-i+1-n+m+(i+n)/r]^{l-k+j-i+g+2-b}}\right\}
\end{aligned}
$$

$$
\begin{aligned}
4)\quad W_I \;=\;& \frac{k\,\rho^l}{l!\,(1-1/e)^k}\sum_{g=0}^{i}\binom{i}{g}\left\{(l+k+g)!\left[\frac{1}{(\rho+j+1)^{l+k+g+1}}\right.\right.\\
&\left.- e^{-(\rho+j+1)}\sum_{b=0}^{l+k+g}\frac{1}{b!\,(\rho+j+1)^{l+k+g+1-b}}\right]\\
&\left.+ e^{-(\rho+j+1)}\sum_{b=0}^{g}\frac{1}{b!\,(\rho+j+1)^{g+1-b}}\right\}
\end{aligned}
$$

$$
\begin{aligned}
5)\quad W_I \;=\;& \frac{(j-i+1)\,\binom{k}{j-i+1}\,\rho^{l-k+j-i+1}}{(l-k+j-i+1)!\,(1-1/e)^k}\sum_{g=0}^{i}\binom{i}{g}\sum_{n=0}^{k-j+i-1}\binom{k-j+i-1}{n}\\
&\cdot(-1)^{k-j+i-1-n}\,(l-k+2j-2i+g+2)!\\
&\cdot\left\{\frac{1}{(\rho+n+j+1)^{l-k+2j-2i+g+3}} - e^{-(\rho+n+j+1)}\right.\\
&\left.\cdot\sum_{b=0}^{l-k+2j-2i+g+2}\frac{1}{b!\,(\rho+n+j+1)^{l-k+2j-2i+g+3-b}}\right\}
\end{aligned}
$$

$$
6)\quad W_I \;=\; 0 \tag{3.90}
$$

Domains for W_N in Equation (3.92)					
1	$r=0$	$i=0$	$k=j-i$	$j=i$	$l\geq k-j+i$
2	$0<r\neq 1$	$i\geq 0$		$j\geq i$	
3	$r=1$				
4	otherwise				

Domain

$$
\begin{aligned}
1)\quad W_N &= \frac{\rho^l}{l!}\, e^{-\rho} \\
2)\quad W_N &= \frac{\rho^{l+1}}{l!} \left[\frac{1-e^{1/r-1}}{(1-1/e)(1-1/r)}\right]^k \sum_{g=0}^{i} \frac{\binom{i}{g}}{r^g}\, g!\, e^{-(\rho+j/r)} \\
&\quad \cdot \sum_{b=0}^{g} \frac{1}{b!\,(\rho+j/r)^{g+1-b}} \\
3)\quad W_N &= \frac{\rho^{l+1}}{l!\,(1-1/e)^k} \sum_{g=0}^{i} \frac{\binom{i}{g}}{r^g}\, g!\, e^{-(\rho+j)} \sum_{b=0}^{g} \frac{1}{b!\,(\rho+j)^{g+1-b}} \\
4)\quad W_N &= 0
\end{aligned}
\tag{3.91}
$$

Following the process of numerical calculation depicted in Figure 3.39, we calculate the normalized power P and normalized response time T_n versus load ρ in Figure 3.44 and 3.45.

The performance of the system gets worse with an increase of the restart-to-initial ratio r. For higher r the distribution of the restarts is more significant to the system performance than is the distribution of the initial service times, since restarts are more frequent. Thus, the system M/D-M(SR,r) performs better than M/M-D(SR,r). Note that the results match the simulation results well. The fit is not as good as for the M/M(SR,r) and M/M-D(SR,r) systems due to the approximation of the arrivals of initial customers. The numerical error causes the artifact of rising power curves and falling response time curves for greater values of r and ρ.

3.3.7 System P4: M/D(S,r) with Approximate $p_{i,k,j,l}$'s

We first substitute Equation (3.57), (3.58), (3.73) or (3.74), (3.75) or (3.76), (3.81), and (3.82) into (3.52). Then, after integration according to Equation (3.52) we get the following expressions for the M/D(S,r) system. [5]

Domains for W_O in Equation (3.92)					
1	$r=0$	$i\geq 1$	$k\geq j-i+1$	$j=i-1$	$l=k-j+i-1$
2	$0<r\leq 1/2$			$j\geq i-1$	$l\geq k-j+i-1$
3	$1/2<r\leq 1$		$k=j-i+1$		
4	$r>1$				
5	$1/2<r\leq 1$		$k>j-i+1$		
6	otherwise				

[5] For convenience, a field in a domain table is left blank if it has the same contents as the field above it.

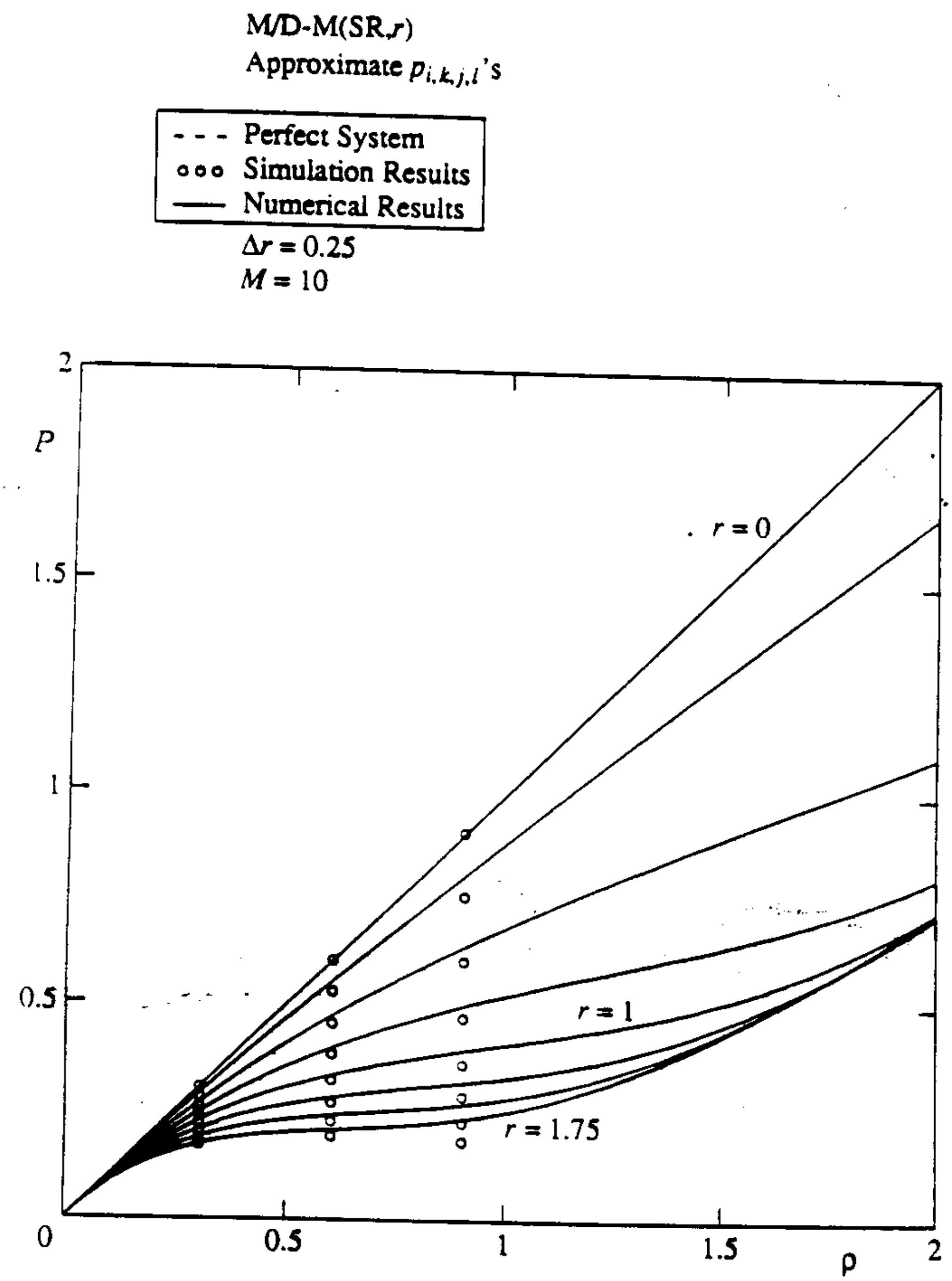


Figure 3.44: Normalized Power for M/D-M(SR,r)

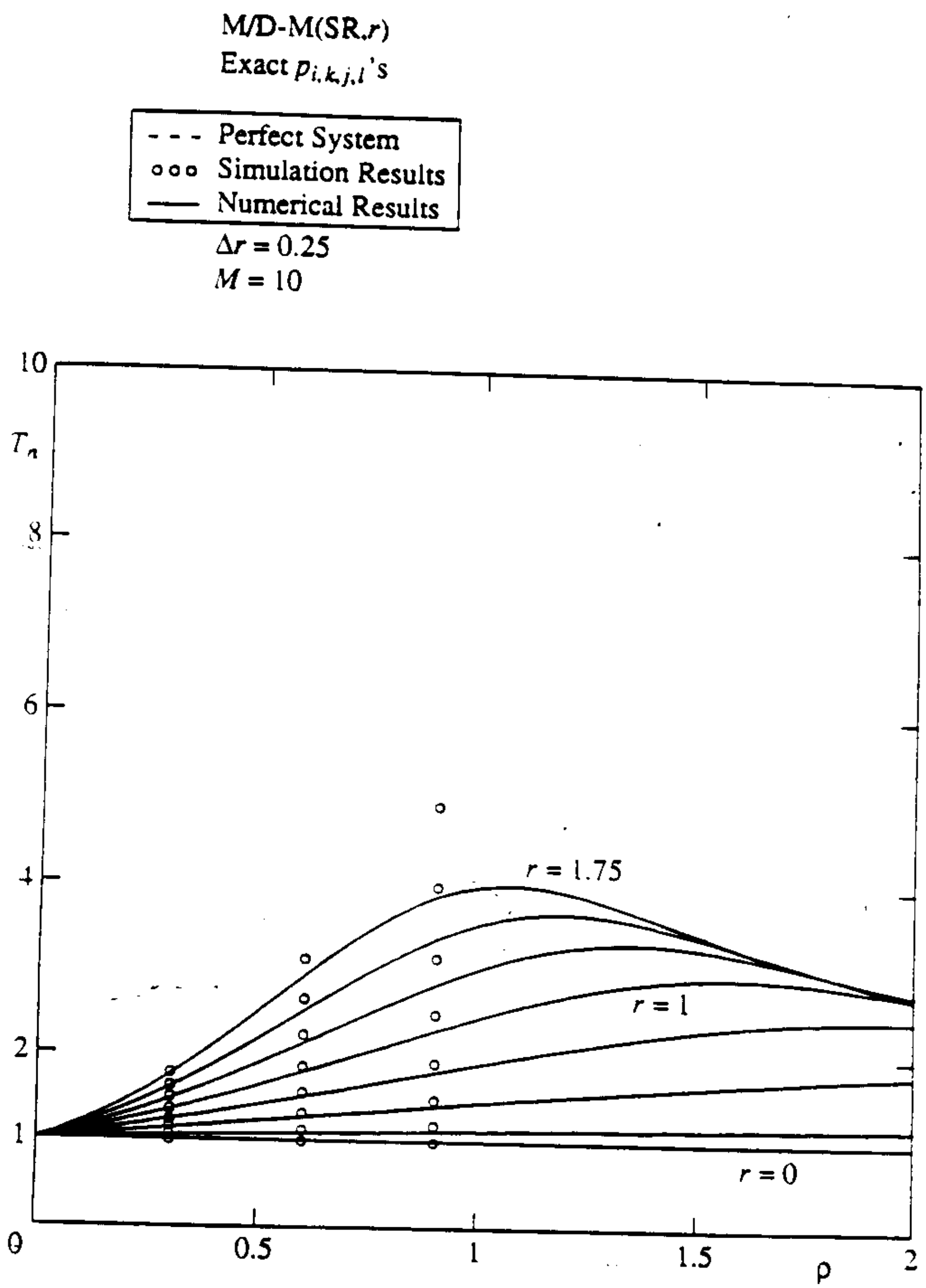


Figure 3.45: Normalized Average Response Time for M/D-M(SR.r)

Domain

$$1)\quad W_O = 1$$

$$2)\quad W_O = \frac{i\,\rho^{l-k+j-i+1}\,(e^r-1)^{j-i+1}}{r\,(1-1/e)^{i+k}} \sum_{h=0}^{i-1} \binom{i-1}{h} \sum_{n=0}^{k-j+i-1} \binom{k-j+i-1}{n}$$
$$\cdot (-1/e)^{2i+k-j-2-h-n} \Big\{ e^{-r(\rho+j-i+1+n)}$$
$$\cdot \sum_{m=0}^{l-k+j-i+1} \frac{r^m}{m!\,[\rho+j-i+1+n+(h+1)/r]^{l-k+j-i+2-m}}$$
$$- e^{-[2r(\rho+j-i+1+n)+h+1]}$$
$$\cdot \sum_{m=0}^{l-k+j-i+1} \frac{(2r)^m}{m!\,[\rho+j-i+1+n+(h+1)/r]^{l-k+j-i+2-m}} \Big\}$$

$$3)\quad W_O = \frac{i\,\rho^l}{r\,(1-1/e)^{i+k}} \Big\{ (e^r-1)^k \sum_{h=0}^{i-1} \binom{i-1}{h} (-1/e)^{i-1-h}$$
$$\cdot \Big[e^{-r(\rho+k)} \sum_{m=0}^{l} \frac{r^m}{m!\,[\rho+k+(h+1)/r]^{l-m+1}}$$
$$- e^{-[\rho+k+(h+1)(1/r-1)]} \sum_{m=0}^{l} \frac{1}{m!\,[\rho+k+(h+1)/r]^{l-m+1}} \Big]$$
$$+ \frac{1}{l!} \sum_{h=0}^{i-1} \binom{i-1}{h} \sum_{g=0}^{k} \binom{k}{g} (-1/e)^{i+k-1-h-g}$$
$$\cdot \frac{e^{-[\rho+(h+1)(1/r-1)+g(1-r)]} - e^{-(2r\rho+h+1+gr)}}{\rho+(h+1)/r+g} \Big\}$$

$$4)\quad W_O = \frac{i\,\rho^l}{r\,l!\,(1-1/e)^{i+k}} \sum_{h=0}^{i-1} \binom{i-1}{h} \sum_{g=0}^{k} \binom{k}{g} (-1/e)^{i+k-1-h-g}$$
$$\cdot \frac{e^{-r\rho} - e^{-(2r\rho+h+1+gr)}}{\rho+(h+1)/r+g}$$

(3.92)

5) $$W_O = \frac{i\,\rho^{l-k+j-i+1}\,(e^r-1)^{j-i+1}}{r\,(1-1/e)^{i+k}} \sum_{h=0}^{i-1} \binom{i-1}{h} \sum_{n=0}^{k-j+i-1} \binom{k-j+i-1}{n}$$

$$\cdot\,(-1/e)^{2i+k-j-2-h-n} \left\{ e^{-r(\rho+j-i+1+n)} \right.$$

$$\cdot \sum_{m=0}^{l-k+j-i+1} \frac{r^m}{m!\,[\rho+j-i+1+n+(h+1)/r]^{l-k+j-i+2-m}}$$

$$-\,e^{-[\rho+j-i+1+n+(h+1)(1/r-1)]}$$

$$\left. \cdot \sum_{m=0}^{l-k+j-i+1} \frac{1}{m!\,[\rho+j-i+1+n+(h+1)/r]^{l-k+j-i+2-m}} \right\}$$

6) $$W_O = 0$$

(3.93)

Domains for W_I in Equation (3.93)					
1	$r=0$	$i=0$	$k\geq j-i+1$	$j=i$	$l\geq k-j+i-1$
2	$0<r\leq 1$		$k=j-i+1$	$j\geq i$	
3	$1/2<r\leq 1$	$i>0$			
4	$r>1$	$i\geq 0$			
5	$0<r\leq 1$	$i=0$	$k>j-i+1$		
	$1/2<r\leq 1$	$i>0$			
6	$0<r\leq 1/2$		$k\geq j-i+1$		
7	otherwise				

<u>Domain</u>

1) $$W_I = \frac{k\,\rho^l}{l!\,(1-1/e)^k} \sum_{g=0}^{k-1} \binom{k-1}{g} (-1/e)^{k-1-g} \frac{1-e^{-(\rho+g+1)}}{\rho+g+1}$$

2) $$W_I = \frac{k\ \rho^l}{(1-1/e)^k} \left\{ (e^r-1)^{k-1} \cdot \left[e^{-r(\rho+k-1)} \sum_{m=0}^{l} \frac{r^m}{m!\ (\rho+k)^{l-m+1}} - e^{-(\rho+k-r)} \sum_{m=0}^{l} \frac{1}{m!\ (\rho+k)^{l-m+1}} \right] + \frac{1}{l!} \sum_{g=0}^{k-1} \binom{k-1}{g} (-1/e)^{k-1-g} \cdot \frac{e^{-(\rho+(g+1)(1-r))r} - e^{-[(1+r)\rho+r(g+1)]}}{\rho+g+1} \right\}$$

3) $$W_I = \frac{k\ \rho^l}{(1-1/e)^{i+k}} \left\{ (e^r-1)^{k-1} \sum_{h=0}^{i} \binom{i}{h} (-1/e)^{i-h} \cdot \left[e^{-r(\rho+k-1)} \sum_{m=0}^{l} \frac{r^m}{m!\ (\rho+k+h/r)^{l-m+1}} - e^{-(\rho+k+h/r-h-r)} \sum_{m=0}^{l} \frac{1}{m!\ (\rho+k+h/r)^{l-m+1}} \right] + \frac{1}{l!} \sum_{h=0}^{i} \binom{i}{h} \sum_{g=0}^{k-1} \binom{k-1}{g} (-1/e)^{i+k-1-h-g} \cdot \frac{e^{-[\rho+(g+1)(1-r)+h/r-h]} - e^{-[r(2\rho+g+1)+h]}}{\rho+g+1+h/r} \right\}$$

4) $$W_I = \frac{k\ \rho^l}{l!\ (1-1/e)^{i+k}} \sum_{h=0}^{i} \binom{i}{h} \sum_{g=0}^{k-1} \binom{k-1}{g} (-1/e)^{i+k-1-h-g} \frac{e^{-r\rho} - e^{-[(1+r)\rho+g+1+h/r]}}{\rho+g+1+h/r}$$

$$5)\quad W_I = \frac{(j-i+1)\,\rho^{l-k+j-i+1}\,\binom{k}{j-i+1}\,(e^r-1)^{j-i}}{(1-1/e)^{i+k}}$$
$$\cdot \sum_{h=0}^{i} \binom{i}{h} \sum_{n=0}^{k-j+i-1} \binom{k-j+i-1}{n} (-1/e)^{2i+k-j-1-h-n}$$
$$\cdot \Bigg\{ e^{-r(\rho+j-i+n)}$$
$$\cdot \sum_{m=0}^{l-k+j-i+1} \frac{r^m}{m!\,(\rho+j-i+1+n+h/r)^{l-k+j-i+2-m}}$$
$$- e^{-(\rho+j-i+1+n+h/r-h-r)}$$
$$\cdot \sum_{m=0}^{l-k+j-i+1} \frac{1}{m!\,(\rho+j-i+1+n+h/r)^{l-k+j-i+2-m}} \Bigg\}.$$

$$6)\quad W_I = \frac{(j-i+1)\,\rho^{l-k+j-i+1}\,\binom{k}{j-i+1}\,(e^r-1)^{j-i}}{(1-1/e)^{i+k}}$$
$$\cdot \sum_{h=0}^{i} \binom{i}{h} \sum_{n=0}^{k-j+i-1} \binom{k-j+i-1}{n} (-1/e)^{2i+k-j-1-h-n}$$
$$\cdot \Bigg\{ e^{-r(\rho+j-i+n)}$$
$$\cdot \sum_{m=0}^{l-k+j-i+1} \frac{r^m}{m!\,(\rho+j-i+1+n+h/r)^{l-k+j-i+2-m}}$$
$$- e^{-[2r(\rho+j-i+n)+h+r]}$$
$$\cdot \sum_{m=0}^{l-k+j-i+1} \frac{(2r)^m}{m!\,(\rho+j-i+1+n+h/r)^{l-k+j-i+2-m}} \Bigg\}$$

$$7)\quad W_I = 0 \qquad (3.94)$$

Domains for W_N in Equation (3.95)					
1	$r \geq 0$	$i=0$	$k=j-i$	$j=i$	$l \geq 0$
2	$0<r\leq 1$			$j>i$	
3	$r>1$	$i\geq 0$			
4	$1/2<r\leq 1$	$i>0$		$j\geq i$	
5	$r>1$			$j=i$	
6	otherwise				

Domain

$$
\begin{aligned}
1)\quad W_N &= \frac{\rho^l}{l!}e^{-\rho}\\
2)\quad W_N &= \frac{\rho^{l+1}}{l!\,(1-1/e)^k}\sum_{g=0}^{k}\binom{k}{g}(-1/e)^{k-g}\\
&\quad\cdot\frac{e^{-[\rho+g(1-r)]}-e^{-[(1+r)\rho+g]}}{\rho+g}\\
3)\quad W_N &= \frac{\rho^l}{l!}(e^{-\rho}-e^{-r\rho})+\frac{\rho^{l+1}}{l!\,(1-1/e)^{i+k}}\sum_{h=0}^{i}\binom{i}{h}\\
&\quad\cdot\sum_{g=0}^{k}\binom{k}{g}(-1/e)^{i+k-h-g}\,\frac{e^{-r\rho}-e^{-[(1+r)\rho+h/r+g]}}{\rho+h/r+g}\\
4)\quad W_N &= \frac{\rho^{l+1}}{l!\,(1-1/e)^{i+k}}\sum_{h=0}^{i}\binom{i}{h}\sum_{g=0}^{k}\binom{k}{g}(-1/e)^{i+k-h-g}\\
&\quad\cdot\frac{e^{-[\rho+h/r-h+g(1-r)]}-e^{-(2r\rho+h+gr)}}{\rho+h/r+g}\\
5)\quad W_N &= \frac{\rho^l}{l!}(e^{-\rho}-e^{-r\rho})+\frac{\rho^{l+1}}{l!\,(1-1/e)^{i}}\sum_{h=0}^{i}\binom{i}{h}(-1/e)^{i-h}\\
&\quad\cdot\frac{e^{-r\rho}-e^{-(2r\rho+h)}}{\rho+h/r}\\
6)\quad W_N &= 0
\end{aligned}
\qquad (3.95)
$$

Following the process of numerical calculation depicted in Figure 3.39, we calculate the normalized power P and normalized response time T_n versus load ρ in Figure 3.46 and 3.47. In the same figures we also show the simulation results for M/M(SR,r) given previously in Figure 3.23.

The performance of the system gets worse with an increase of the restart-to-initial ratio r. For $r=0$ the system behaves as an M/D(S,0). Every customer gets served once and leaves regardless of the eventual conflict with other customers. This is an M/D/∞ system, for which $P=\rho$ and $T_n=1$. Thus, results plotted for $r=0$ are trivially analytic (and were not obtained through the numerical computation).

Note that the results match the simulation results well. The fit is not as good as for the M/M(SR,r), M/M-D(SR,r), and M/D-M(SR,r) systems due to the approximation of the arrivals of both initial and old customers. The numerical error causes the artifact of rising power curves and falling response time curves for greater values of r and ρ.

3.3.8 System P5: M/M-D$_q$,M(sBR,r) with Exact $p_{i,k,j,l}$'s

We first substitute Equation (3.53), (3.54), (3.69), (3.70), (3.79), and (3.80) into (3.52). Then, after integration according to Equation (3.52) we get the following expressions for the M/M-D$_q$,M(sBR,r) system. [6]

[6] For convenience, a field in a domain table is left blank if it has the same contents as the field above it.

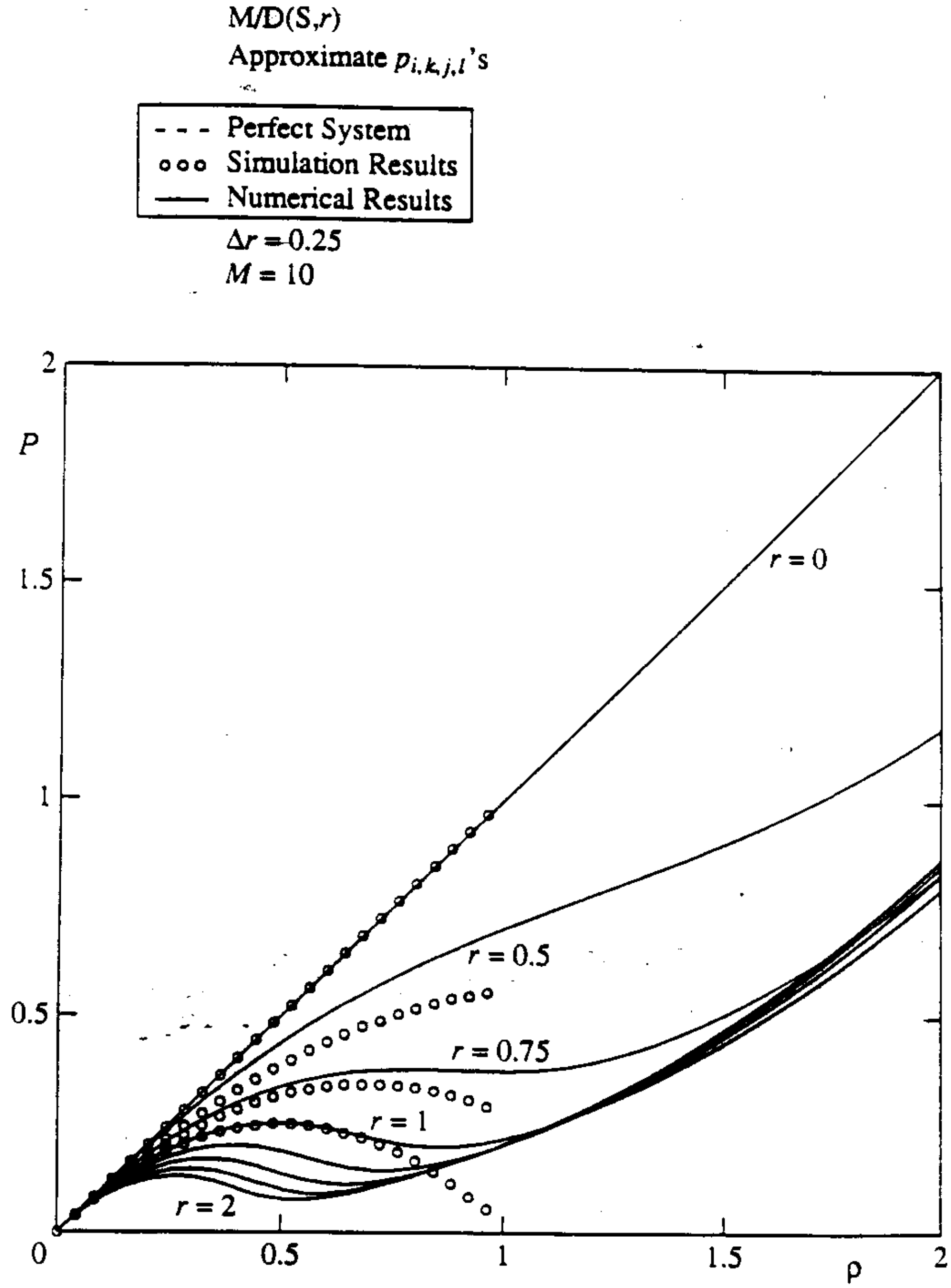


Figure 3.46: Normalized Power for M/D(S.r)

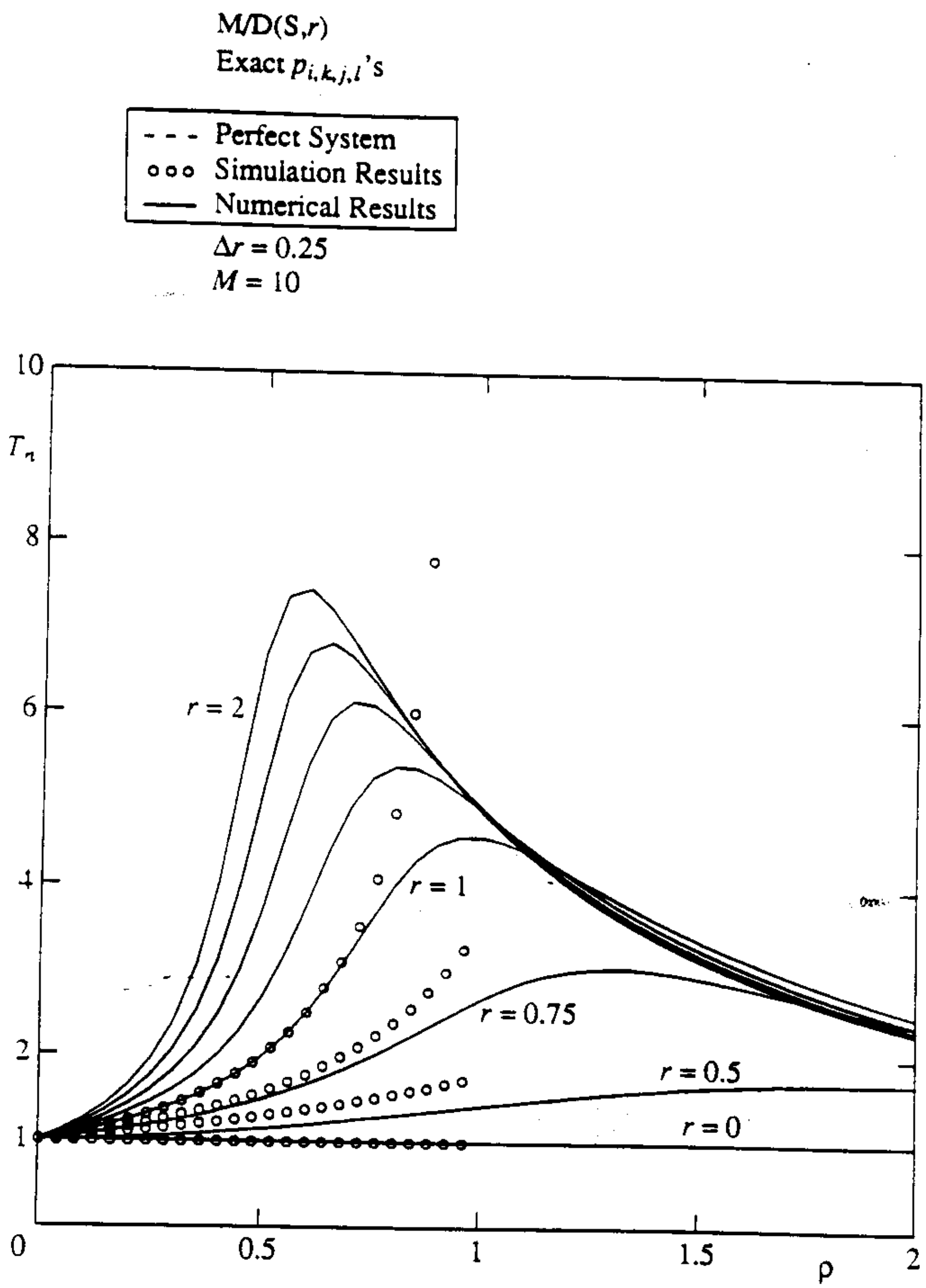


Figure 3.47: Normalized Average Response Time for M/D(S,r)

Domains for W_O in Equation (3.96)						
1	$0 \le q_r \le 1$	$r=0$	$i \ge 1$	$k \ge j-i+1$	$j=i-1$	$l \ge k-j+i-1$
2	$0 \le q_r < 1$	$r>0$			$j \ge i-1$	
3	$q_r=1$					
4	otherwise					

Domain

$$1)\quad W_O = 1$$

$$2)\quad W_O = \frac{i\binom{k}{j-i+1}\rho^{l-k+j-i+1}}{rp_r\,(l-k+j-i+1)!\,(1/r-1)^{j-i+1}}\sum_{n=0}^{j-i+1}\binom{j-i+1}{n}(-1)^n \cdot \sum_{m=0}^{l-k+j-i+1}\binom{l-k+j-i+1}{m}(-1)^m\frac{e^{-rq_r(\rho+k-n+m+n/r)}}{\rho+k-n+m+n/r+i/(rp_r)}$$

$$3)\quad W_O = \frac{\binom{k}{j-i+1}\rho^{l-k+j-i+1}}{(l-k+j-i+1)!}\,e^{-r(\rho+k-j+i-1)}\left(\frac{e^{-r}-1/e}{1/r-1}\right)^{j-i+1} \cdot (1-e^{-r})^{l-k+j-i+1}$$

$$4)\quad W_O = 0 \tag{3.96}$$

Domains for W_I in Equation (3.97)						
1	$0 \le q_r \le 1$	$r=0$	$i=0$	$k \ge j-i+1$	$j=i$	$l \ge k-j+i-1$
2	$0 \le q_r < 1$	$r>0$	$i \ge 0$		$j \ge i-1$	
3	$q_r=1$					
4	otherwise					

Domain

$$1) \quad W_I = \frac{k\,\rho^{l-k+1}}{(l-k+1)!} \sum_{m=0}^{l-k+1} \binom{l-k+1}{m} (-1)^m \left[\frac{1}{\rho+k+m-1} - \frac{1}{\rho+k+m}\right]$$

$$2) \quad W_I = \frac{(j-i+1)\binom{k}{j-i+1}\rho^{l-k+j-i+1}}{r\,(l-k+j-i+1)!\,(1/r-1)^{j-i+1}} \cdot \sum_{n=0}^{j-i+1} \binom{j-i+1}{n} (-1)^n \cdot \sum_{m=0}^{l-k+j-i+1} \binom{l-k+j-i+1}{m} (-1)^m \cdot \left[\frac{1-e^{-rq_r(\rho+k-n+m+n/r)}}{\rho+k-n+m+n/r} + \frac{e^{-rq_r(\rho+k-n+m+n/r)}}{\rho+k-n+m+n/r+i/(rp_r)}\right]$$

$$3) \quad W_I = \frac{(j-i+1)\binom{k}{j-i+1}\rho^{l-k+j-i+1}}{r\,(l-k+j-i+1)!\,(1/r-1)^{j-i+1}} \sum_{n=0}^{j-i+1} \binom{j-i+1}{n} (-1)^n \cdot \sum_{m=0}^{l-k+j-i+1} \binom{l-k+j-i+1}{m} (-1)^m \frac{1-e^{-r(\rho+k-n+m+n/r)}}{\rho+k-n+m+n/r}$$

$$4) \quad W_I = 0 \tag{3.97}$$

Domains for W_N in Equation (3.98)						
1	$0 \le q_r \le 1$	$r=0$	$i=0$	$k \ge j-i$	$j=i$	$l \ge k-j+i$
2	$0 \le q_r < 1$	$r>0$	$i \ge 0$		$j \ge i$	
3	$q_r = 1$					
4	otherwise					

Domain

$$1) \quad W_N = \frac{\rho^{l-k+1}}{(l-k)!} \sum_{m=0}^{l-k+1} \frac{\binom{l-k+1}{m}(-1)^m}{\rho+k+m}$$

$$
\begin{aligned}
2)\quad W_N &= \frac{\binom{k}{j-i}\,\rho^{l-k+j-i+1}}{(l-k+j-i)!\,(1/r-1)^{j-i}} \sum_{n=0}^{j-i} \binom{j-i}{n}(-1)^n \\
&\cdot \sum_{m=0}^{l-k+j-i+1} \binom{l-k+j-i+1}{m}(-1)^m \\
&\cdot \left[\frac{1-e^{-rq_r(\rho+k-n+m+n/r)}}{\rho+k-n+m+n/r} + \frac{e^{-rq_r(\rho+k-n+m+n/r)}}{\rho+k-n+m+n/r+i/(rp_r)}\right] \\
3)\quad W_N &= \frac{\binom{k}{j-i}\,\rho^{l-k+j-i+1}}{(l-k+j-i)!\,(1/r-1)^{j-i}} \sum_{n=0}^{j-i} \binom{j-i}{n}(-1)^n \\
&\cdot \sum_{m=0}^{l-k+j-i+1} \binom{l-k+j-i+1}{m}(-1)^m \frac{1-e^{-r(\rho+k-n+m+n/r)}}{\rho+k-n+m+n/r} \\
4)\quad W_N &= 0
\end{aligned}
\tag{3.98}
$$

Following the process of numerical calculation depicted in Figure 3.39, we calculate the normalized power P and normalized response time T_n versus load ρ in Figures 3.48 through 3.53.[7]

The performance of the system gets worse with an increase of the restart-to-initial ratio r. The system performs better than M/M-D(SR,r) due to the broadcast nature of restarts. The results match the simulation results well, especially for memoryless restart service times (system M/M(sBR,r)). For non-memoryless restart service times, as r increases, the truncation of the state space gives us results somewhat better than those obtained through simulation.

3.3.9 System P6: M/D-D$_{q_r}$M(sBR,r) with Approximate $p_{i,k,j,l}$'s

We first substitute Equation (3.53), (3.54), (3.73) or (3.74), (3.75) or (3.76), (3.81), and (3.82) into (3.52). Then, after integration according to Equation

[7]Curves for $r = 1$ are not plotted due to instability in the numerical calculations.

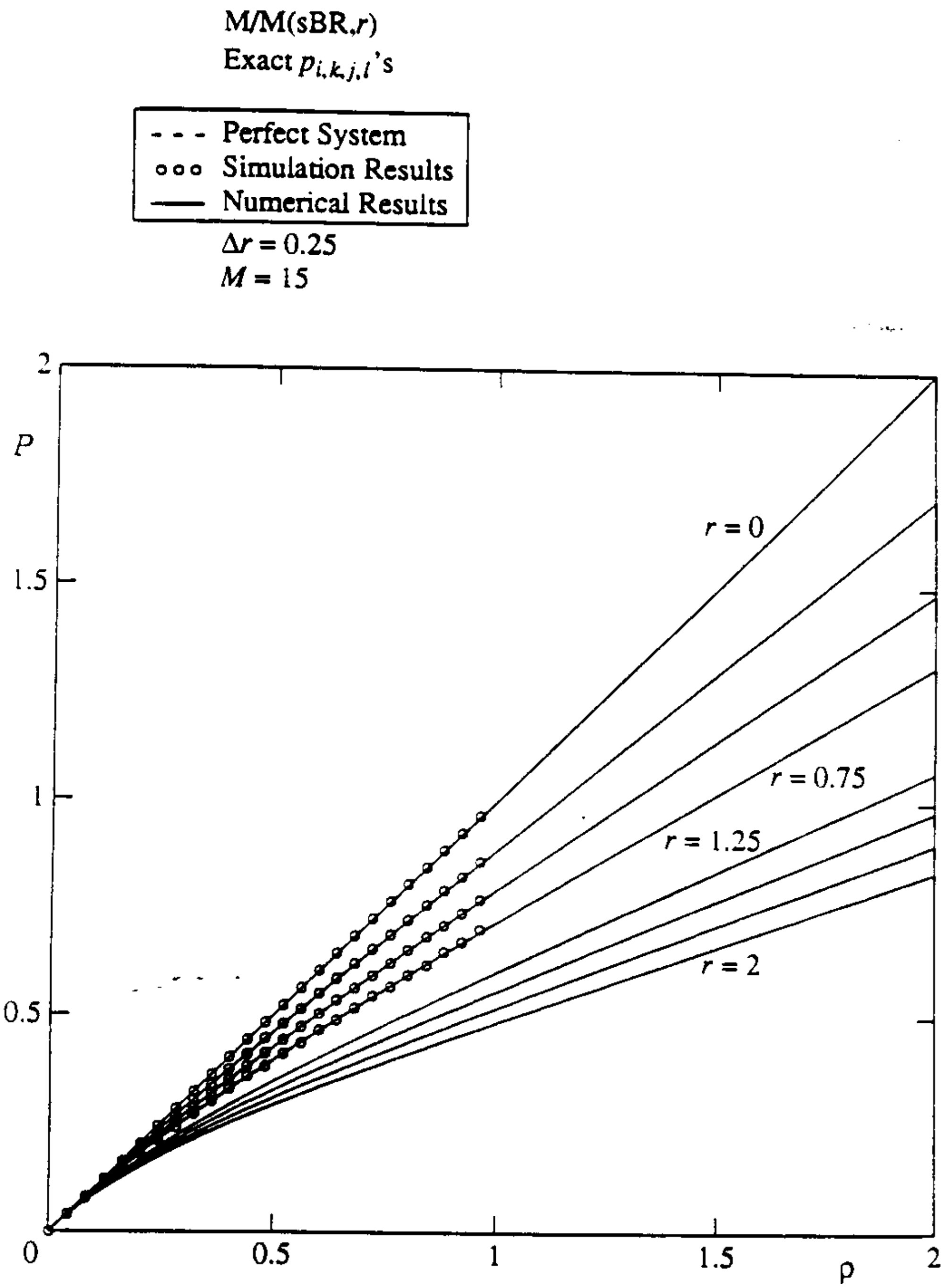


Figure 3.48: Normalized Power for M/M(sBR,r)

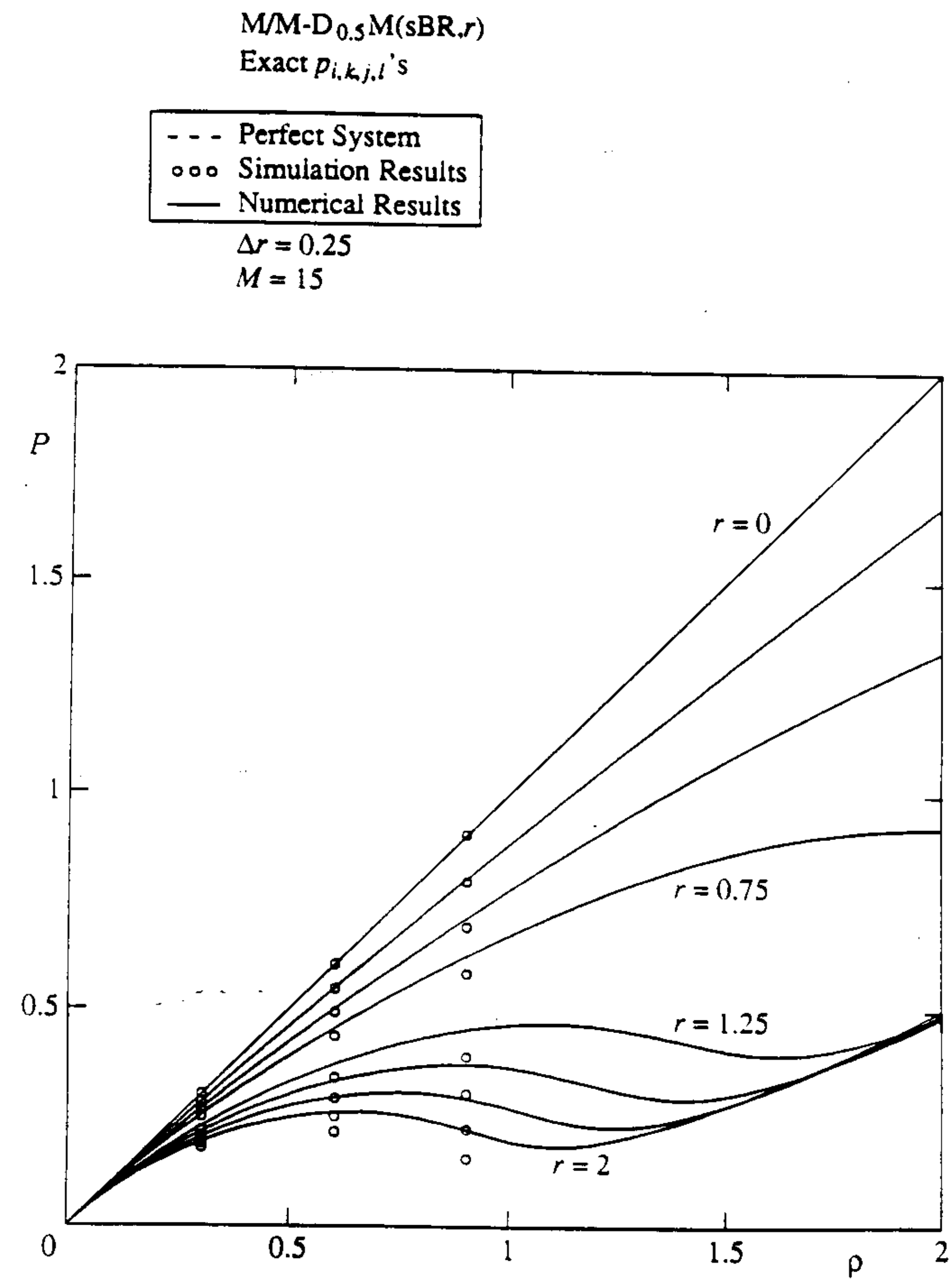


Figure 3.49: Normalized Power for M/M-D$_{0.5}$M(sBR.r)

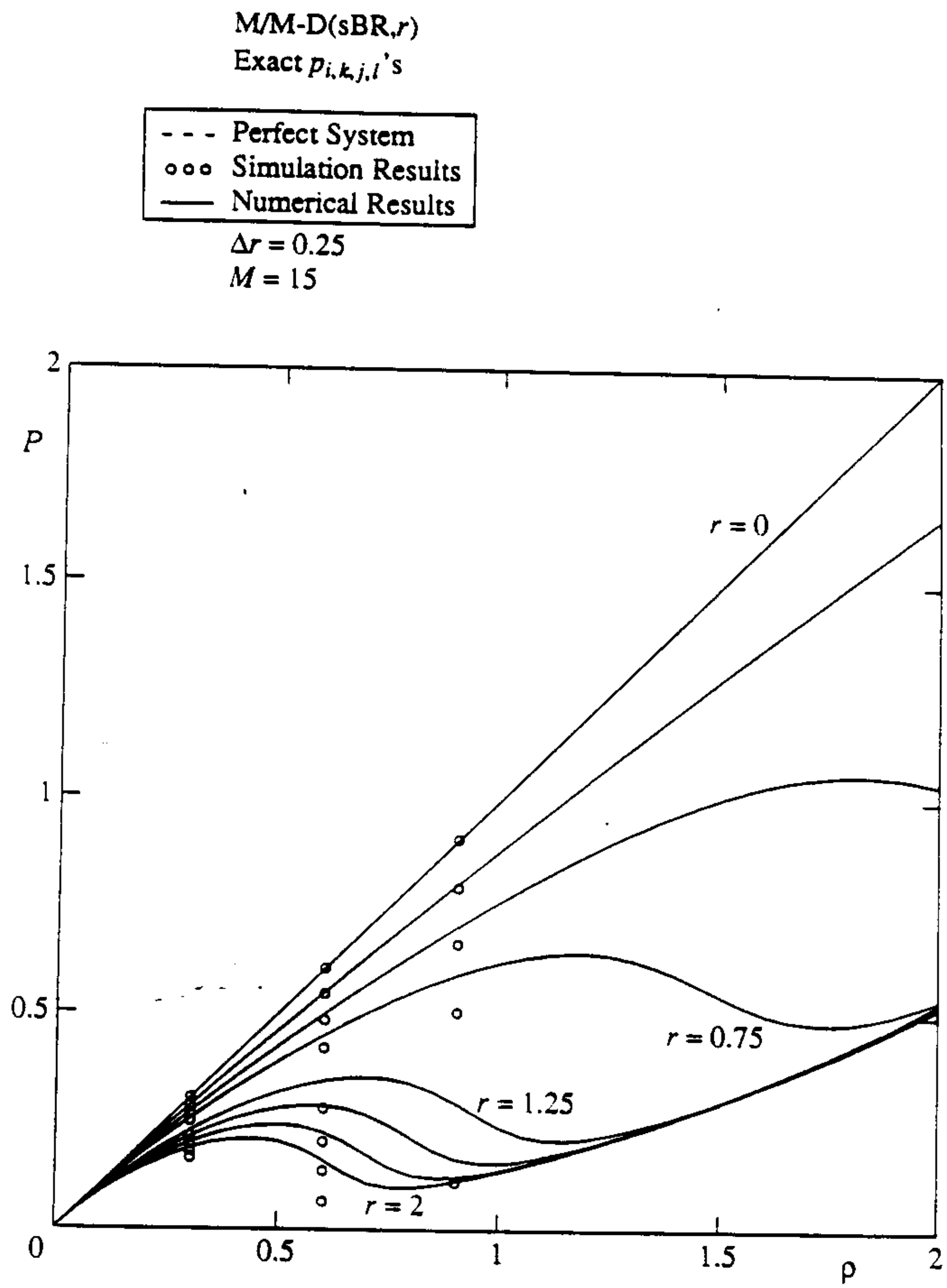


Figure 3.50: Normalized Power for M/M-D(sBR,r)

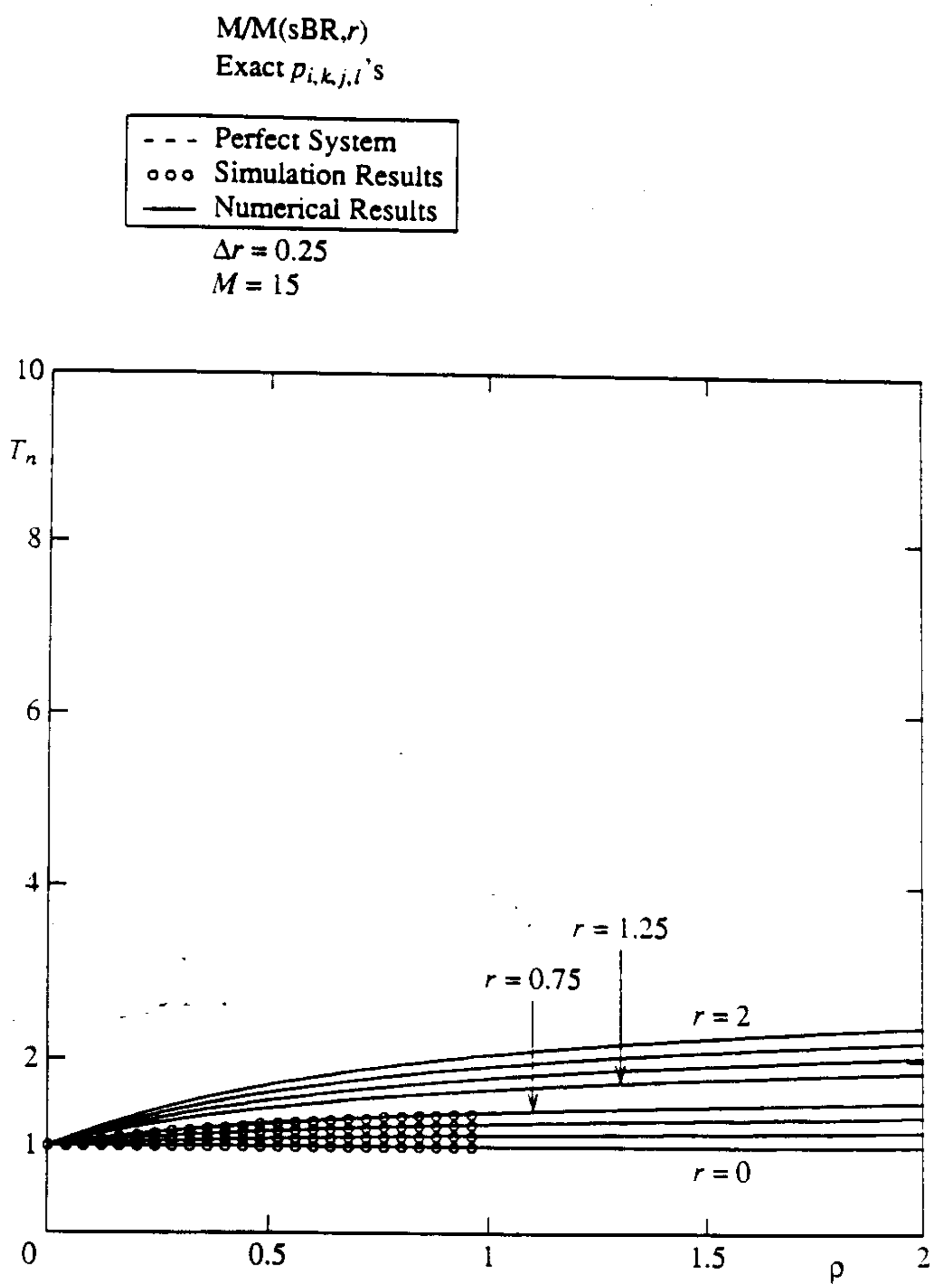


Figure 3.51: Normalized Average Response Time for M/M(sBR.r)

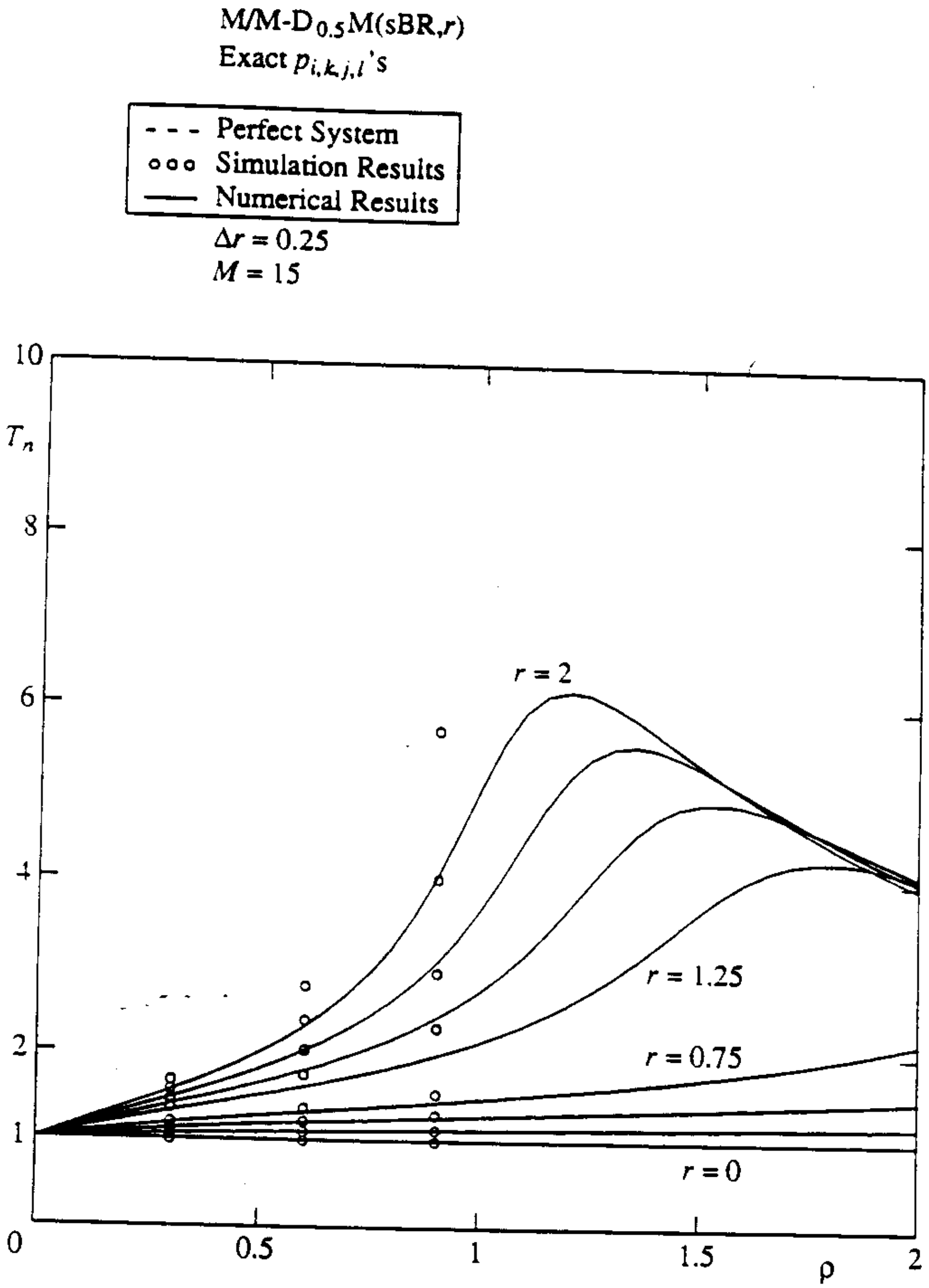


Figure 3.52: Normalized Average Response Time for M/M-$D_{0.5}$M(sBR,r)

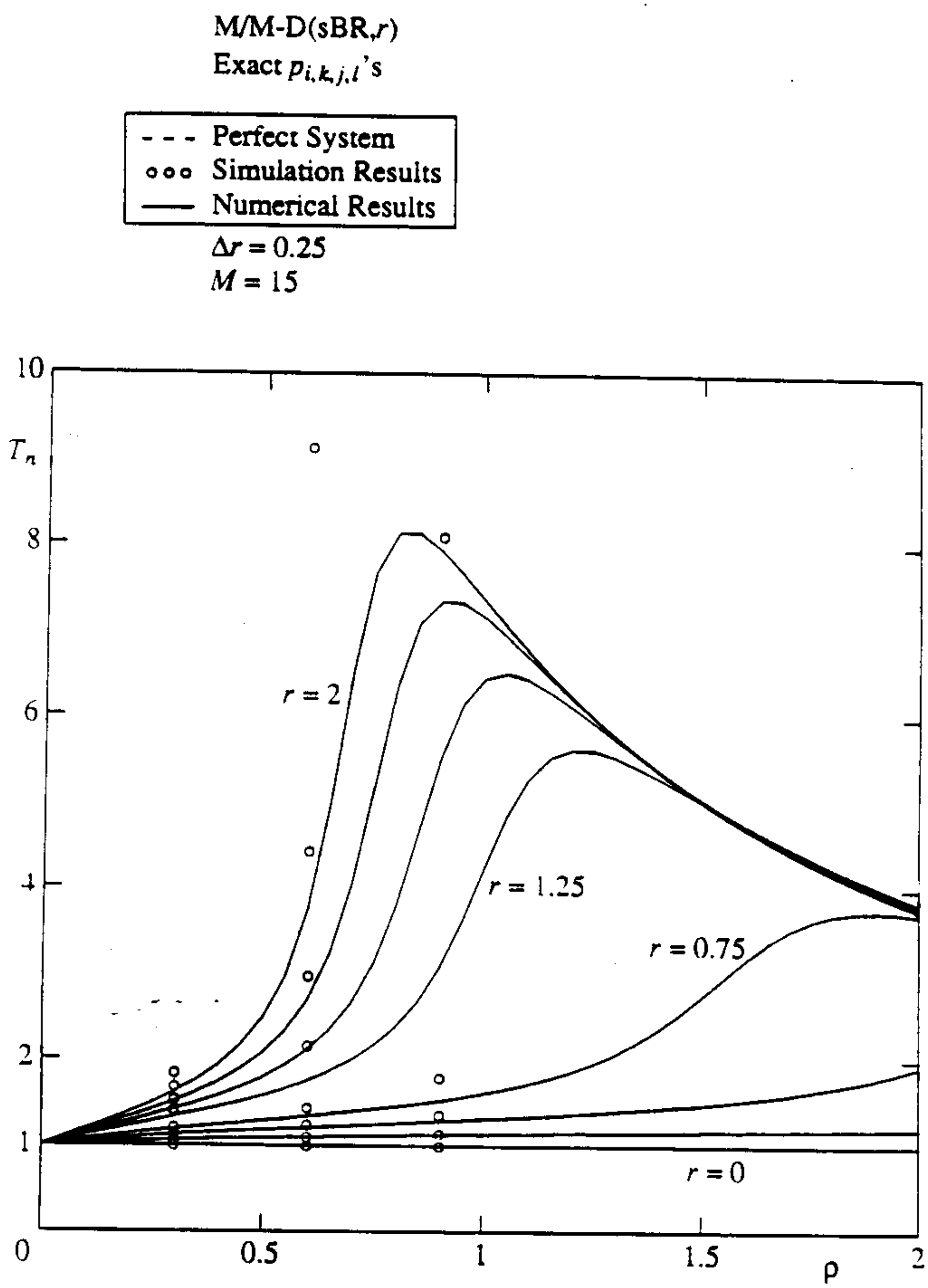


Figure 3.53: Normalized Average Response Time for M/M-D(sBR.r)

(3.52) we get the following expressions for the M/D-D$_{q_r}$M(sBR,r) system. [8]

Domains for W_O in Equation (3.99)								
1	$0 \le q_r \le 1$	$r=0$	$i \ge 1$	$k=j-i+1$	$j=i-1$	$l=0$		
2	$0 \le q_r < 1$	$r>0$			$j \ge i-1$	$l \ge 0$	$1 < rq_r$	$1 \ne rp_r$
3								$1 = rp_r$
4	$q_r=1$							
5	$0 \le q_r < 1$						$1 \ge rq_r$	$1 \ne rp_r$
6								$1 = rp_r$
7				$k > j-i+1$				$1 \ne rp_r$
8								$1 = rp_r$
9	$q_r=1$			$k=j-i+1$				
10				$k > j-i+1$				
11	otherwise							

Domain

1) $W_O = 1$

2)
$$W_O = \frac{i\,\rho^l}{l!\; rp_r(1-1/e)^k(1-rp_r)^k}\left\{\sum_{g=0}^{k}\binom{k}{g}(-rp_r)^{k-g}\sum_{c=0}^{g}\binom{g}{c}\right.$$
$$\cdot\,[(rp_r-1)/e]^{g-c}\,(-rp_r)^{k-g}\,\frac{e^{-rq_r\rho}-e^{-[(1+rq_r)\rho+c+(i+k-g)/(rp_r)]}}{\rho+c+(i+k-g)/(rp_r)}$$
$$\left.+\,\{rp_r[e^{-1+1/(rp_r)}-1]\}^k\,\frac{e^{-[(1+rq_r)\rho+(i+k)/(rp_r)]}}{\rho+(i+k)/(rp_r)}\right\}$$

[8] For convenience, a field in a domain table is left blank if it has the same contents as the field above it.

$$3)\quad W_O = \frac{i\,\rho^l}{l!\,(1-1/e)^k}\left\{\sum_{g=0}^{k}\binom{k}{g}(-1/e)^{k-g}\sum_{c=0}^{g}\binom{g}{c}(1-rq_r)^{g-c}\,c!\right.$$

$$\cdot\left[e^{-rq_r\rho}\sum_{d=0}^{c}\frac{(rq_r)^d}{d!\,(\rho+i+g)^{c-d+1}} - e^{-[(1+rq_r)\rho+i+g]}\right.$$

$$\left.\left.\cdot\sum_{d=0}^{c}\frac{(1+rq_r)^d}{d!\,(\rho+i+g)^{c-d+1}}\right] + \frac{e^{-[(1+rq_r)\rho+i+k]}}{\rho+i+k}\right\}$$

$$4)\quad W_O = \frac{\rho^l}{l!}\,e^{-rq_r\rho}$$

$$5)\quad W_O = \frac{i\,\rho^l}{rp_r\,(1-1/e)^k(1-rp_r)^k}\left\{\sum_{g=0}^{k}\binom{k}{g}(e^{rq_r}-1+rp_r)^g\,(-rp_r)^{k-g}\right.$$

$$\cdot\left[e^{-rq_r(\rho+g)}\sum_{m=0}^{l}\frac{(rq_r)^m}{m!\,[\rho+g+(i+k-g)/(rp_r)]^{l-m+1}}\right.$$

$$- e^{-[\rho+g+(i+k-g)(1-rq_r)/(rp_r)]}$$

$$\left.\cdot\sum_{m=0}^{l}\frac{1}{m!\,[\rho+g+(i+k-g)/(rp_r)]^{l-m+1}}\right]$$

$$+\frac{1}{l!}\sum_{g=0}^{k}\binom{k}{g}(-rp_r)^{k-g}\sum_{c=0}^{g}\binom{g}{c}[(rp_r-1)/e]^{g-c}$$

$$\cdot\frac{e^{-\{\rho+(1-rq_r)[c+(i+k-g)/(rp_r)]\}} - e^{-[(1+rq_r)\rho+c+(i+k-g)/(rp_r)]}}{\rho+c+(i+k-g)/(rp_r)}$$

$$\left.+\frac{1}{l!}\{rp_r[e^{-1+1/(rp_r)}-1]\}^k\,\frac{e^{-[(1+rq_r)\rho+(i+k)/(rp_r)]}}{\rho+(i+k)/(rp_r)}\right\}$$

$$6)\quad W_O = \frac{i\,\rho^l}{l!\,(1-1/e)^k}\left\{\sum_{g=0}^{k}\binom{k}{g}e^{grq_r}(-1)^{k-g}\sum_{c=0}^{g}\binom{g}{c}(1-rq_r)^{g-c}(1+c)!\right.$$

$$\cdot\left[e^{-rq_r\rho}\sum_{m=0}^{l+c}\frac{(rq_r)^m}{m!\,(\rho+i)^{l+c+1-m}}\right.$$

$$\left.- e^{-[\rho+i(1-rq_r)]}\sum_{m=0}^{l+c}\frac{1}{m!\,(\rho+i)^{l+c+1-m}}\right]$$

$$+\sum_{g=0}^{k}\binom{k}{g}(-1/e)^{k-g}\sum_{c=0}^{g}\binom{g}{c}(1-rq_r)^{g-c}\,c!$$

$$\cdot\left[e^{-[\rho+(i+g)(1-rq_r)]}\sum_{d=0}^{c}\frac{1}{d!\,(\rho+i+g)^{c-d+1}}\right.$$

$$\left.\left.- e^{-[(1+rq_r)\rho+i+g]}\sum_{d=0}^{c}\frac{(1+rq_r)^d}{d!\,(\rho+i+g)^{c-d+1}}\right] + \frac{e^{-[(1+rq_r)\rho+i+k]}}{\rho+i+k}\right\}$$

$$7)\quad W_O = \frac{i\binom{k}{j-i+1}\rho^{l-k+j-i+1}}{rp_r\,(1-1/e)^k(1-rp_r)^{j-i+1}} \sum_{g=0}^{j-i+1} \binom{j-i+1}{g} (e^{rq_r}-1+rp_r)^g$$

$$\cdot (-rp_r)^{j-i+1-g} \sum_{n=0}^{k-j+i-1} \binom{k-j+i-1}{n} (-1/e)^{k-j+i-1-n} \Bigg\{ e^{-rq_r(\rho+n+g)}$$

$$\cdot \sum_{m=0}^{l-k+j-i+1} \frac{(rq_r)^m}{m!\,[\rho+n+g+(j+1-g)/(rp_r)]^{l-k+j-i+2-m}}$$

$$- e^{-[\rho+n+g+(j+1-g)(1-rq_r)/(rp_r)]}$$

$$\cdot \sum_{m=0}^{l-k+j-i+1} \frac{1}{m!\,[\rho+n+g+(j+1-g)/(rp_r)]^{l-k+j-i+2-m}} \Bigg\}$$

$$8)\quad W_O = \frac{i\binom{k}{j-i+1}\rho^{l-k+j-i+1}}{(l-k+j-i+1)!\,rp_r(1-1/e)^k} \sum_{g=0}^{j-i+1} \binom{j-i+1}{g} e^{grq_r} (-1)^{j-i+1-g}$$

$$\cdot \sum_{c=0}^{g} \binom{g}{c} (1-rq_r)^{g-c} \sum_{n=0}^{k-j+i-1} \binom{k-j+i-1}{n} (-1/e)^{k-j+i-1-n}$$

$$\cdot (l-k+j-i+1+c)!$$

$$\cdot \Bigg\{ e^{-rq_r(\rho+k+c)} \sum_{m=0}^{l-k+j-i+1+c} \frac{(rq_r)^m}{m!\,(\rho+j+1+n-irq_r)^{l-k+j-i+2+c-m}}$$

$$- e^{-[(1+rq_r)\rho+j+1+n-irq_r]}$$

$$\cdot \sum_{m=0}^{l-k+j-i+1+c} \frac{1}{m!\,(\rho+j+1+n-irq_r)^{l-k+j-i+2+c-m}} \Bigg\}$$

$$9)\quad W_O = \frac{(rq_r\rho)^l}{l!} \left(\frac{1-e^{-rq_r}}{1-1/e}\right)^k e^{-rq_r\rho}$$

$$10)\quad W_O = \frac{\binom{k}{j-i+1}(rq_r\rho)^{l-k+j-i+1}}{(l-k+j-i+1)!\,(1-1/e)^k} (e^{rq_r}-1)^{j-i+1}(e^{-rq_r}-1/e)^{k-j+i-1}$$

$$\cdot e^{-rq_r(\rho+j-i+1)}$$

$$11)\quad W_O = 0 \qquad (3.99)$$

Domains for W_I in Equation (3.100)								
1	$0 \le q_r \le 1$	$r=0$	$i=0$	$k \ge j-i+1$	$j=i$	$l \le k-j+i-1$		
2	$0 \le q_r < 1$	$r>0$	$i \ge 0$	$k=j-i+1$	$j \ge i$		$1 < rq_r$	$1 \ne rp_r$
3								$1=rp_r$
4	$q_r=1$		$i=0$					
5	$0 \le q_r < 1$		$i \ge 0$				$1 \ge rq_r$	$1 \ne rp_r$
6								$1=rp_r$
7				$k>j-i+1$				$1 \ne rp_r$
8								$1=rp_r$
9	$q_r=1$		$i=0$	$k=j-i+1$				
10				$k>j-i+1$				
11	otherwise							

<u>Domain</u>

1) $$W_I = \frac{k\,\rho^{l-k+1}}{(1-1/e)^k} \sum_{g=0}^{k-1} \frac{\binom{k-1}{g}\,(-1/e)^{k-1-g}}{(\rho+g+1)^{l-k+2}} \cdot \left\{1 - e^{-(\rho+g+1)} \sum_{m=0}^{l-k+1} \frac{(\rho+g+1)^m}{m!}\right\}$$

2) $$W_I = \frac{k\,\rho^{l}}{l!\,(1-1/e)^k(1-rp_r)^k} \left\{ \sum_{g=0}^{k-1} \binom{k-1}{g}\,(-rp_r)^{k-1-g} \cdot \sum_{c=0}^{g} \binom{g}{c}\,[(rp_r-1)/e]^{g-c}\, e^{-rq_r\rho} \cdot \left[\frac{1-e^{-[\rho+c+1+(i+k-1-g)/(rp_r)]}}{\rho+c+1+(i+k-1-g)/(rp_r)} - \frac{1-e^{-[\rho+c+(i+k-g)/(rp_r)]}}{\rho+c+(i+k-g)/(rp_r)} \right] + (rp_r)^{k-1}\,[e^{-1+1/(rp_r)}-1]^k\, \frac{e^{-[(1+rq_r)\rho+(i+k)/(rp_r)]}}{\rho+(i+k)/(rp_r)} \right\}$$

$$3)\quad W_I = \frac{k\ \rho^l}{l!\ (1-1/e)^k} \left\{ \sum_{g=0}^{k-1} \binom{k-1}{g} (-1/e)^{k-1-g} \sum_{c=0}^{g} \binom{g}{c} (1-rq_r)^{g-c}\ c! \right.$$

$$\cdot \left[e^{-\rho}\ (c+1) \sum_{d=0}^{c+1} \frac{(rq_r)^d}{d!\ (\rho+i+1+g)^{c+2-d}} \right.$$

$$- e^{-\rho}\ rq_r \sum_{d=0}^{c} \frac{(rq_r)^d}{d!\ (\rho+i+1+g)^{c+1-d}}$$

$$- e^{-[(1+rq_r)\rho+i+1+g]}\ (c+1) \sum_{d=0}^{c+1} \frac{(1+rq_r)^d}{d!\ (\rho+i+1+g)^{c+2-d}}$$

$$\left. + e^{-[(1+rq_r)\rho+i+1+g]}\ rq_r \sum_{d=0}^{c} \frac{(1+rq_r)^d}{d!\ (\rho+i+1+g)^{c+1-d}} \right]$$

$$\left. + \frac{e^{-(\rho+i+k)}}{\rho+i+k} \right\}$$

$$4)\quad W_I = \frac{k\ \rho^l}{l!\ (1-1/e)^k} \sum_{g=0}^{k-1} \binom{k-1}{g} (-1/e)^{k-1-g}\ \frac{e^{-rq_r\rho} - e^{-[(1+rq_r)\rho+g+1]}}{\rho+g+1}$$

5) $$\begin{aligned} W_l = {} & \frac{k\,\rho^l}{(1-1/e)^k(1-rp_r)^k} \left\{ \sum_{g=0}^{k-1} \binom{k-1}{g} \left(e^{rq_r}-1+rp_r\right)^g \left(-rp_r\right)^{k-1-g} \right. \\ & \cdot \left[e^{-rq_r(\rho+g)} \sum_{m=0}^{l} \frac{(rq_r)^m}{m!\,[\rho+g+1+(i+k-1-g)/(rp_r)]^{l-m+1}} \right. \\ & - e^{-rq_r(\rho+g)} \sum_{m=0}^{l} \frac{(rq_r)^m}{m!\,[\rho+g+(i+k-g)/(rp_r)]^{l-m+1}} \\ & - e^{-[\rho+g+1+(i+k-1-g)(1-rq_r)/(rp_r)]} \\ & \cdot \sum_{m=0}^{l} \frac{1}{m!\,[\rho+g+1+(i+k-1-g)/(rp_r)]^{l-m+1}} \\ & + e^{-[\rho+g+(i+k-g)(1-rq_r)/(rp_r)]} \\ & \cdot \left. \sum_{m=0}^{l} \frac{1}{m!\,[\rho+g+(i+k-g)/(rp_r)]^{l-m+1}} \right] \\ & + \frac{1}{l!} \sum_{g=0}^{k-1} \binom{k-1}{g} \left(-rp_r\right)^{k-1-g} \sum_{c=0}^{g} \binom{g}{c} \left[(rp_r-1)/e\right]^{g-c} \\ & \cdot \left[\frac{e^{-\{\rho+(1-rq_r)[c+1+(i+k-1-g)/(rp_r)]\}} - e^{-[(1+rq_r)\rho+c+1+(i+k-1-g)/(rp_r)]}}{\rho+c+1+(i+k-1-g)/(rp_r)} \right. \\ & \cdot \left. \frac{e^{-\{\rho+(1-rq_r)[c+(i+k-g)/(rp_r)]\}} - e^{-[(1+rq_r)\rho+c+(i+k-g)/(rp_r)]}}{\rho+c+(i+k-g)/(rp_r)} \right] \\ & \left. + \frac{1}{l!} \left(rp_r\right)^{k-1} \left[e^{-1+1/(rp_r)}-1\right]^k \frac{e^{-[(1+rq_r)\rho+(i+k)/(rp_r)]}}{\rho+(i+k)/(rp_r)} \right\} \end{aligned}$$

6) $$W_I = \frac{k\,\rho^l}{l!\,(1-1/e)^k} \left\{ \sum_{g=0}^{k-1} \binom{k-1}{g} e^{grq_r} \left[(1-rq_r)e^{rq_r}-1\right]^{k-1-g} (l+g)! \right.$$

$$\cdot \left[(l+g+1)\, e^{-rq_r(\rho+k-1)} \sum_{m=0}^{l+g+1} \frac{(rq_r)^m}{m!\,(\rho+i+k)^{l+g+2-m}} \right.$$

$$- rq_r e^{-rq_r(\rho+k-1)} \sum_{m=0}^{l+g} \frac{(rq_r)^m}{m!\,(\rho+i+k)^{l+g+1-m}}$$

$$- (l+g+1)\, e^{-[\rho+i+k-(i+1)rq_r]} \sum_{m=0}^{l+g+1} \frac{1}{m!\,(\rho+i+k)^{l+g+2-m}}$$

$$\left. + rq_r e^{-[\rho+i+k-(i+1)rq_r]} \sum_{m=0}^{l+g} \frac{1}{m!\,(\rho+i+k)^{l+g+1-m}} \right]$$

$$+ \sum_{g=0}^{k-1} \binom{k-1}{g} (-1/e)^{k-1-g} \sum_{c=0}^{g} \binom{g}{c} (1-rq_r)^{g-c}\, c!$$

$$\cdot \left[e^{-[\rho+(i+1+g)(1-rq_r)]} (c+1) \sum_{d=0}^{c+1} \frac{1}{d!\,(\rho+i+1+g)^{c+2-d}} \right.$$

$$- e^{-[\rho+(i+1+g)(1-rq_r)]}\, rq_r \sum_{d=0}^{c} \frac{1}{d!\,(\rho+i+1+g)^{c+1-d}}$$

$$- e^{-[(1+rq_r)\rho+i+1+g]} (c+1) \sum_{d=0}^{c+1} \frac{(1+rq_r)^d}{d!\,(\rho+i+1+g)^{c+2-d}}$$

$$\left. \left. + e^{-[(1+rq_r)\rho+i+1+g]}\, rq_r \sum_{d=0}^{c} \frac{(1+rq_r)^d}{d!\,(\rho+i+1+g)^{c+1-d}} \right] + \frac{e^{-(\rho+i+k)}}{\rho+i+k} \right\}$$

$$7) \quad W_O = \frac{(j-i+1)\binom{k}{j-i+1}\rho^{l-k+j-i+1}}{(1-1/e)^k(1-rp_r)^{j-i+1}} \sum_{g=0}^{j-i} \binom{j-i}{g} (e^{rq_r}-1+rp_r)^g$$

$$\cdot (-rp_r)^{j-i-g} \sum_{n=0}^{k-j+i-1} \binom{k-j+i-1}{n} (-1/e)^{k-j+i-1-n} \Bigg\{ e^{-rq_r(\rho+n+g)}$$

$$\cdot \sum_{m=0}^{l-k+j-i+1} \frac{(rq_r)^m}{m!\,[\rho+n+g+1+(j-g)/(rp_r)]^{l-k+j-i+2-m}}$$

$$- e^{-rq_r(\rho+n+g)}$$

$$\cdot \sum_{m=0}^{l-k+j-i+1} \frac{(rq_r)^m}{m!\,[\rho+n+g+(j-g+1)/(rp_r)]^{l-k+j-i+2-m}}$$

$$- e^{-[\rho+n+g+1-rq_r+(j-g)(1-rq_r)/(rp_r)]}$$

$$\cdot \sum_{m=0}^{l-k+j-i+1} \frac{1}{m!\,[\rho+n+g+1+(j-g)/(rp_r)]^{l-k+j-i+2-m}}$$

$$+ e^{-[\rho+n+g+(j-g+1)(1-rq_r)/(rp_r)]}$$

$$\cdot \sum_{m=0}^{l-k+j-i+1} \frac{1}{m!\,[\rho+n+g+(j-g+1)/(rp_r)]^{l-k+j-i+2-m}} \Bigg\}$$

$$8)\quad W_I = \frac{(j-i+1)\binom{k}{j-i+1}\rho^{l-k+j-i+1}}{(l-k+j-i+1)!\,(1-1/e)^k} \sum_{g=0}^{j-i} \binom{j-i}{g} \left[(1-rq_r)e^{rq_r}-1\right]^{j-i-g}$$

$$\cdot\, e^{grq_r}\,(l-k+j-i+1+g)! \sum_{n=0}^{k-j+i-1} \binom{k-j+i-1}{n} (-1/e)^{k-j+i-1-n}$$

$$\cdot \Bigg\{ e^{-rq_r(\rho+j-i+n)}\,(l-k+j-i+2+g)$$

$$\cdot \sum_{m=0}^{l-k+j-i+2+g} \frac{(rq_r)^m}{m!\,(\rho+j+n+1)^{l-k+j-i+3+g-m}}$$

$$-\, e^{-(\rho+j-i+n+1)}\, rq_r \sum_{m=0}^{l-k+j-i+1+g} \frac{1}{m!\,(\rho+j+n+1)^{l-k+j-i+2+g-m}}$$

$$-\, e^{-[\rho+j+n+1-(i+1)rq_r]}\,(l-k+j-i+2+g)$$

$$\cdot \sum_{m=0}^{l-k+j-i+2+g} \frac{1}{m!\,(\rho+j+n+1)^{l-k+j-i+3+g-m}}$$

$$+\, e^{-[\rho+j+n+1-(i+1)rq_r]}\, rq_r$$

$$\cdot \sum_{m=0}^{l-k+j-i+1+g} \frac{1}{m!\,(\rho+j+n+1)^{l-k+j-i+2+g-m}} \Bigg\}$$

$$9)\quad W_I = \frac{k\,\rho^l}{(1-1/e)^k} \Bigg\{ (e^{rq_r}-1)^{k-1} \Bigg[e^{-rq_r(\rho+k-1)} \sum_{m=0}^{l} \frac{(rq_r)^m}{m!\,(\rho+k)^{l-m+1}}$$

$$-\, e^{-(\rho+k-rq_r)} \sum_{m=0}^{l} \frac{1}{m!\,(\rho+k)^{l-m+1}} \Bigg] + \frac{1}{l!} \sum_{g=0}^{k-1} \binom{k-1}{g}$$

$$\cdot\,(-1/e)^{k-1-g}\, \frac{e^{-[\rho+(g+1)(1-rq_r)]} - e^{-[(1+rq_r)\rho+g+1]}}{\rho+g+1} \Bigg\}$$

$$10)\quad W_I = \frac{(j-i+1)\binom{k}{j-i+1}\rho^{l-k+j-i+1}}{(1-1/e)^k} (e^{rq_r}-1)^{j-i} \sum_{n=0}^{k-j+i-1} \binom{k-j+i-1}{n}$$

$$\cdot\,(-1/e)^{k-j+i-1-n}$$

$$\cdot \Bigg[e^{-rq_r(\rho+j-i+n)} \sum_{m=0}^{l-k+j-i+1} \frac{(rq_r)^m}{m!\,(\rho+j-i+n+1)^{l-k+j-i+2-m}}$$

$$-\, e^{-(\rho+j-i+n+1-rq_r)} \sum_{m=0}^{l-k+j-i+1} \frac{1}{m!\,(\rho+j-i+n+1)^{l-k+j-i+2-m}} \Bigg]$$

$$11) \quad W_I = 0 \tag{3.100}$$

Domains for W_N in Equation (3.101)								
1	$0 \leq q_r \leq 1$	$r \geq 0$	$i = 0$	$k = j - i$	$j = i$	$l > 0$		
2	$0 \leq q_r < 1$	$r > 0$	$i > 0$				$1 < rq_r$	
3			$i \geq 0$		$j > i$			$1 \neq rp_r$
4								$1 = rp_r$
5	$q_r = 1$		$i = 0$					
6			$i > 0$		$j \geq i$			
7	$0 \leq q_r < 1$				$j = i$		$1 \geq rq_r$	
8			$i \geq 0$		$j > i$			$1 \neq rp_r$
9								$1 = rp_r$
10	$q_r = 1$		$i = 0$					
11	otherwise							

Domain

$$1) \quad W_N = \frac{\rho^l}{l!} e^{-\rho}$$

$$2) \quad W_N = \frac{\rho^l}{l!} \left[e^{-\rho} - \frac{i}{\rho r p_r + i} e^{-r q_r \rho} \right]$$

$$3) \quad W_N = \frac{\rho^l}{l!} \left(e^{-\rho} - e^{-r q_r \rho}\right) + \frac{\rho^{l+1}}{l!\,(1-1/e)^k (1-rp_r)^k} \cdot \left\{ \sum_{g=0}^{k} \binom{k}{g} (-rp_r)^{k-g} \sum_{c=0}^{g} \binom{g}{c} [(rp_r - 1)/e]^{g-c} \right.$$

$$\cdot \frac{e^{-rq_r\rho} - e^{-[(1+rq_r)\rho+c+(i+k-g)/(rp_r)]}}{\rho+c+(i+k-g)/(rp_r)}$$

$$+ \{rp_r[e^{-1+1/(rp_r)}-1]\}^k \frac{e^{-[(1+rq_r)\rho+(i+k)/(rp_r)]}}{\rho+(i+k)/(rp_r)} \Bigg\}$$

4) $$W_N = \frac{\rho^l}{l!}(e^{-\rho}-e^{-rq_r\rho}) + \frac{\rho^{l+1}}{l!\,(1-1/e)^k} \Bigg\{ \sum_{g=0}^{k} \binom{k}{g} (-1/e)^{k-g}$$

$$\cdot \sum_{c=0}^{g} \binom{g}{c} (1-rq_r)^{g-c}\, c! \Bigg[e^{-rq_r\rho} \sum_{d=0}^{c} \frac{(rq_r)^d}{d!\,(\rho+i+g)^{c-d+1}}$$

$$- e^{-[(1+rq_r)\rho+i+g]} \sum_{d=0}^{c} \frac{(1+rq_r)^d}{d!\,(\rho+i+g)^{c-d+1}} \Bigg] + \frac{e^{-[(1+rq_r)\rho+i+k]}}{\rho+i+k} \Bigg\}$$

5) $$W_N = \frac{\rho^l}{l!}(e^{-\rho}-e^{-rq_r\rho}) + \frac{\rho^{l+1}}{l!\,(1-1/e)^k} \sum_{g=0}^{k} \binom{k}{g} (-1/e)^{k-g}$$

$$\cdot \frac{e^{-rq_r\rho} - e^{-[(1+rq_r)\rho+g]}}{\rho+g}$$

6) $$W_N = \frac{\rho^l}{l!}(e^{-\rho}-e^{-rq_r\rho})$$

7) $$W_N = \frac{\rho^{l+1}}{l!\,[\rho+i/(rp_r)]} e^{-[\rho+i(1-rq_r)/(rp_r)]}$$

8) $$W_N = \frac{\rho^{l+1}}{l!\,(1-1/e)^k(1-rp_r)^k} \Bigg\{ \sum_{g=0}^{k} \binom{k}{g} (-rp_r)^{k-g} \sum_{c=0}^{g} [(rq_r-1)/e]^{g-c}$$

$$\cdot \binom{g}{c} \frac{e^{-\{\rho+(1-rq_r)[c+(i+k-g)/(rp_r)]\}} - e^{-[(1+rq_r)\rho+c+(i+k-g)/(rp_r)]}}{\rho+c+(i+k-g)/(rp_r)}$$

$$+ \{rp_r[e^{-1+1/(rp_r)}-1]\}^k \frac{e^{-[(1+rq_r)\rho+(i+k)/(rp_r)]}}{\rho+(i+k)/(rp_r)} \Bigg\}$$

9) $$W_N = \frac{\rho^{l+1}}{l!\,(1-1/e)^k} \Bigg\{ \sum_{g=0}^{k} \binom{k}{g} (-1/e)^{k-g} \sum_{c=0}^{g} \binom{g}{c} (1-rq_r)^{g-c}\, c!$$

$$\cdot \Bigg[e^{-[\rho+(i+c)(1-rq_r)]} \sum_{d=0}^{c} \frac{1}{d!\,(\rho+i+c)^{c-d+1}}$$

$$- e^{-[(1+rq_r)\rho+i+c]} \sum_{d=0}^{c} \frac{(1+rq_r)^d}{d!\,(\rho+i+c)^{c-d+1}} \Bigg] + \frac{e^{-[(1+rq_r)\rho+i+k]}}{\rho+i+k} \Bigg\}$$

10) $$W_N = \frac{\rho^{l+1}}{l!\,(1-1/e)^k} \sum_{g=0}^{k} \binom{k}{g} (-1/e)^{k-g} \frac{e^{-[\rho+g(1-rq_r)]} - e^{-[(1+rq_r)\rho+g]}}{\rho+g}$$

11) $$W_N = 0 \qquad (3.101)$$

Following the process of numerical calculation depicted in Figure 3.39, we calculate the normalized power P and normalized response time T_n versus load ρ in Figure 3.54 through 3.59. [9]

The performance of the system gets worse with an increase of the restart-to-initial ratio r. Note that the results match the simulation results well. For $r=0$ the system behaves as an M/D(S.0). Every customer gets served once and leaves regardless of the eventual conflict with other customers. This is an $M/D/\infty$ system, for which $P=\rho$ and $T_n=1$. Thus, results plotted for $r=0$ are trivially analytic (and were not obtained through the numerical computation).

For smaller r, the distribution of the initial service time has more effect on the performance of the system. Since the rearrival of the initial customers are approximate, we have higher errors for smaller r.

3.3.10 System P7: M/D$_q$M-D$_{q_r}$M(BR,r) with Exact $p_{i,k,j,l}$'s

We first substitute Equation (3.53), (3.54), (3.65), (3.66), (3.77), and (3.78) into (3.52). Then, after integration according to Equation (3.52) we get the following expressions for the M/D$_q$M-D$_{q_r}$M(BR,r) system. [10]

[9] Curves for $r = 1$ in M/D-M(sBR,r) and for $r = 2$ in M/D-D$_0$.5M(sBR,r) are not plotted due to instability in the numerical calculations.

[10] For convenience, a field in a domain table is left blank if it has the same contents as the field above it.

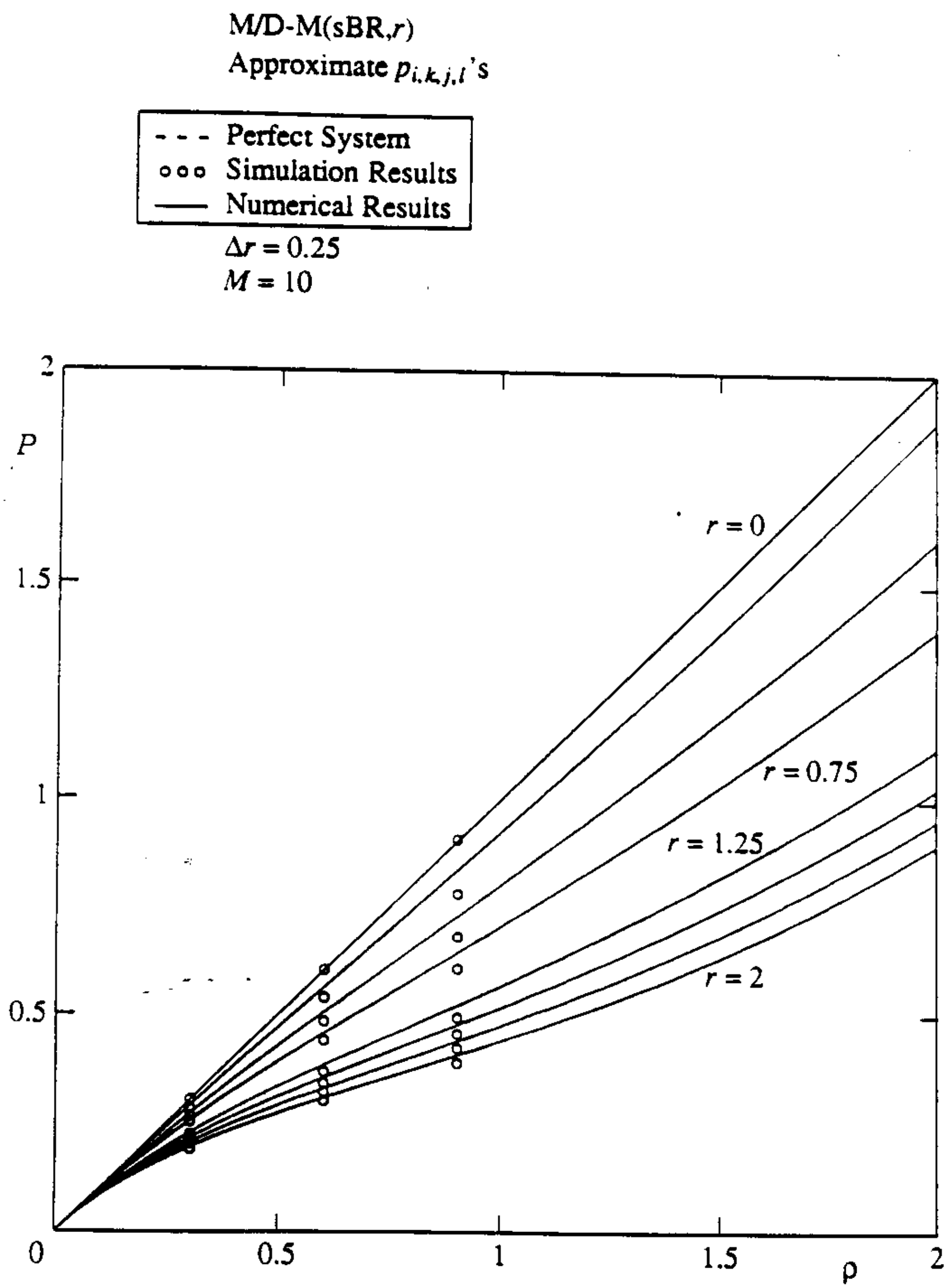


Figure 3.54: Normalized Power for M/D-M(sBR.r)

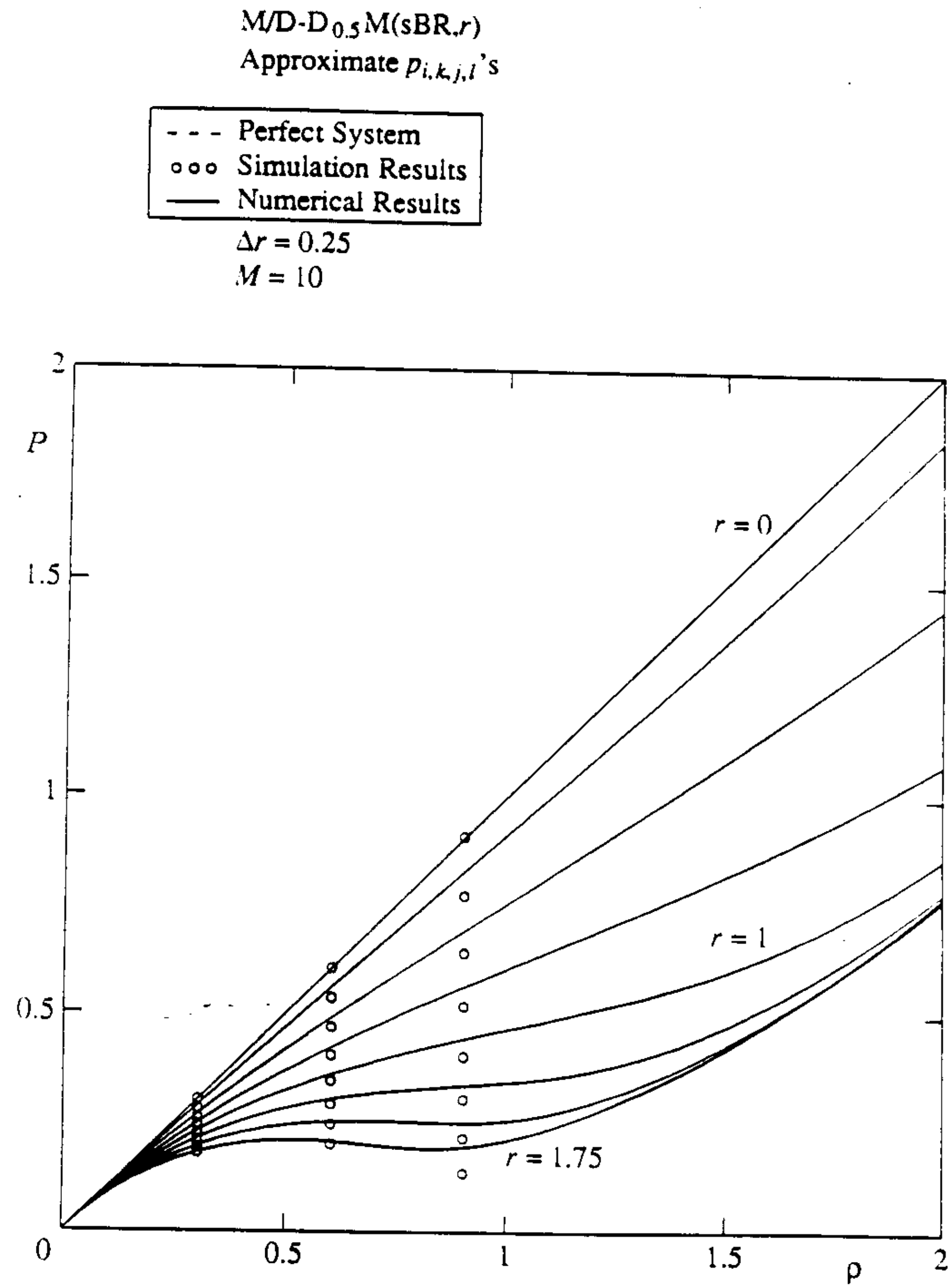


Figure 3.55: Normalized Power for M/D-$D_{0.5}$M(sBR,r)

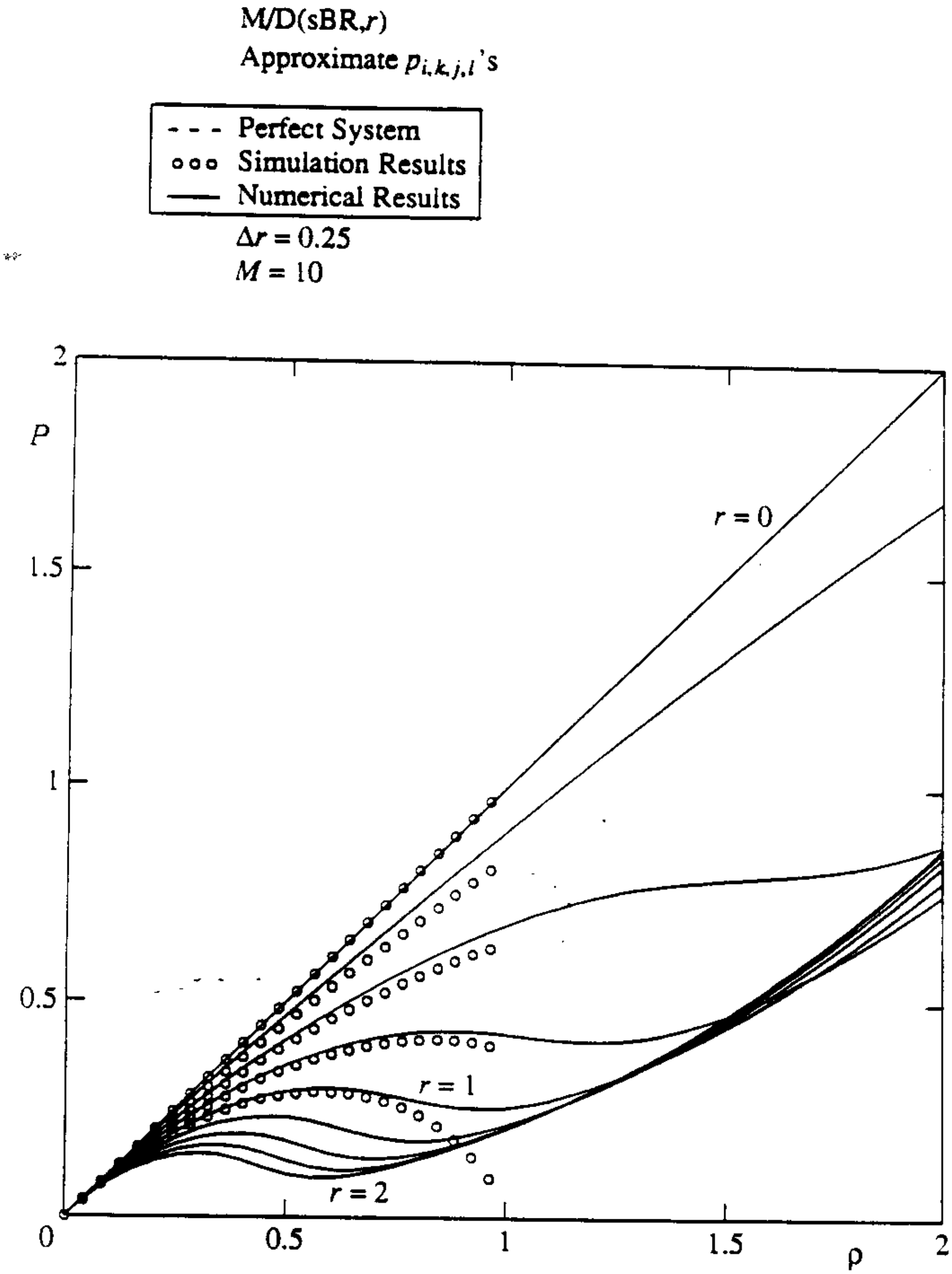


Figure 3.56: Normalized Power for M/D(sBR.r)

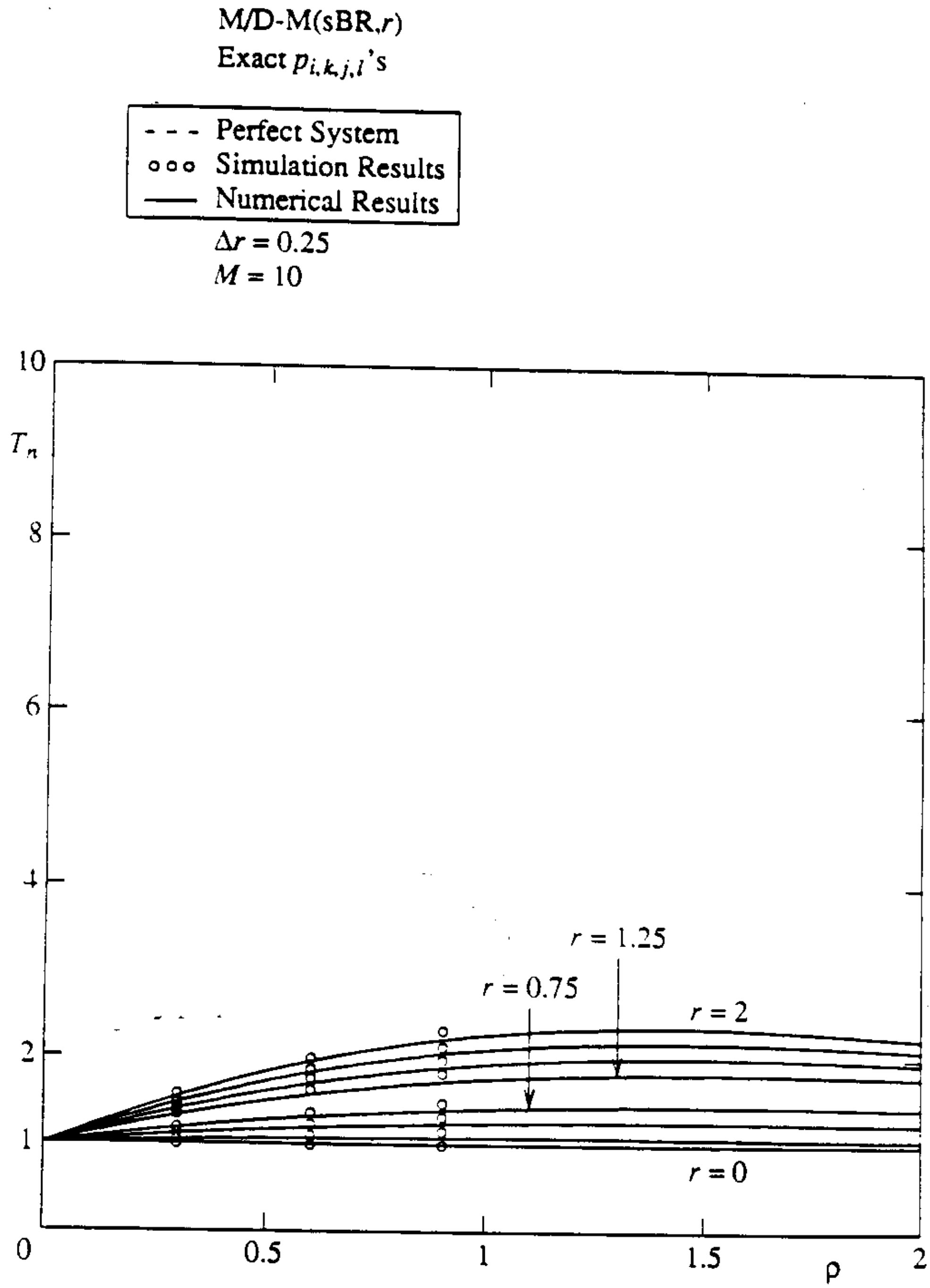


Figure 3.57: Normalized Average Response Time for M/D-M(sBR.r)

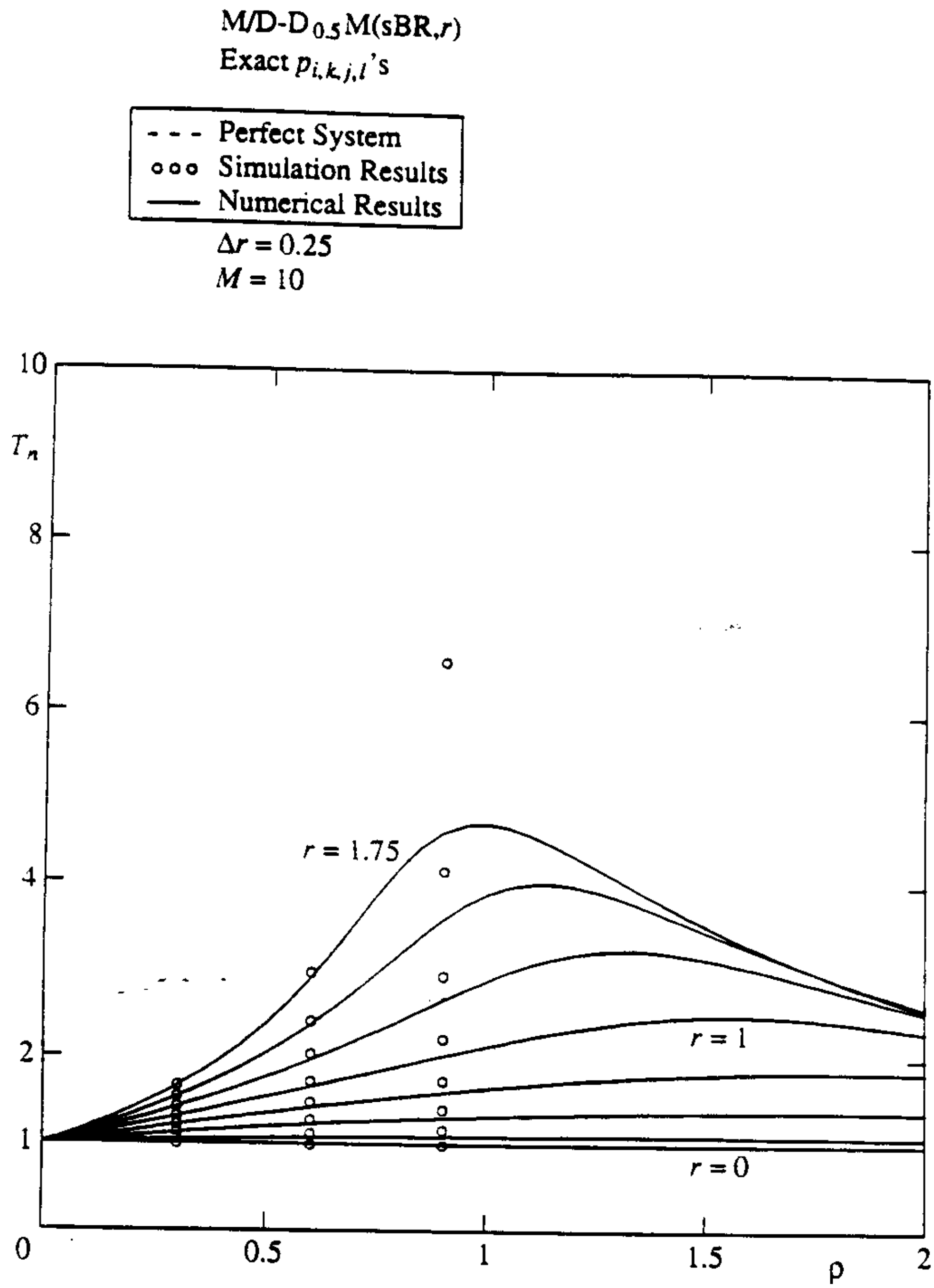


Figure 3.58: Normalized Average Response Time for M/D-$D_{0.5}$M(sBR,r)

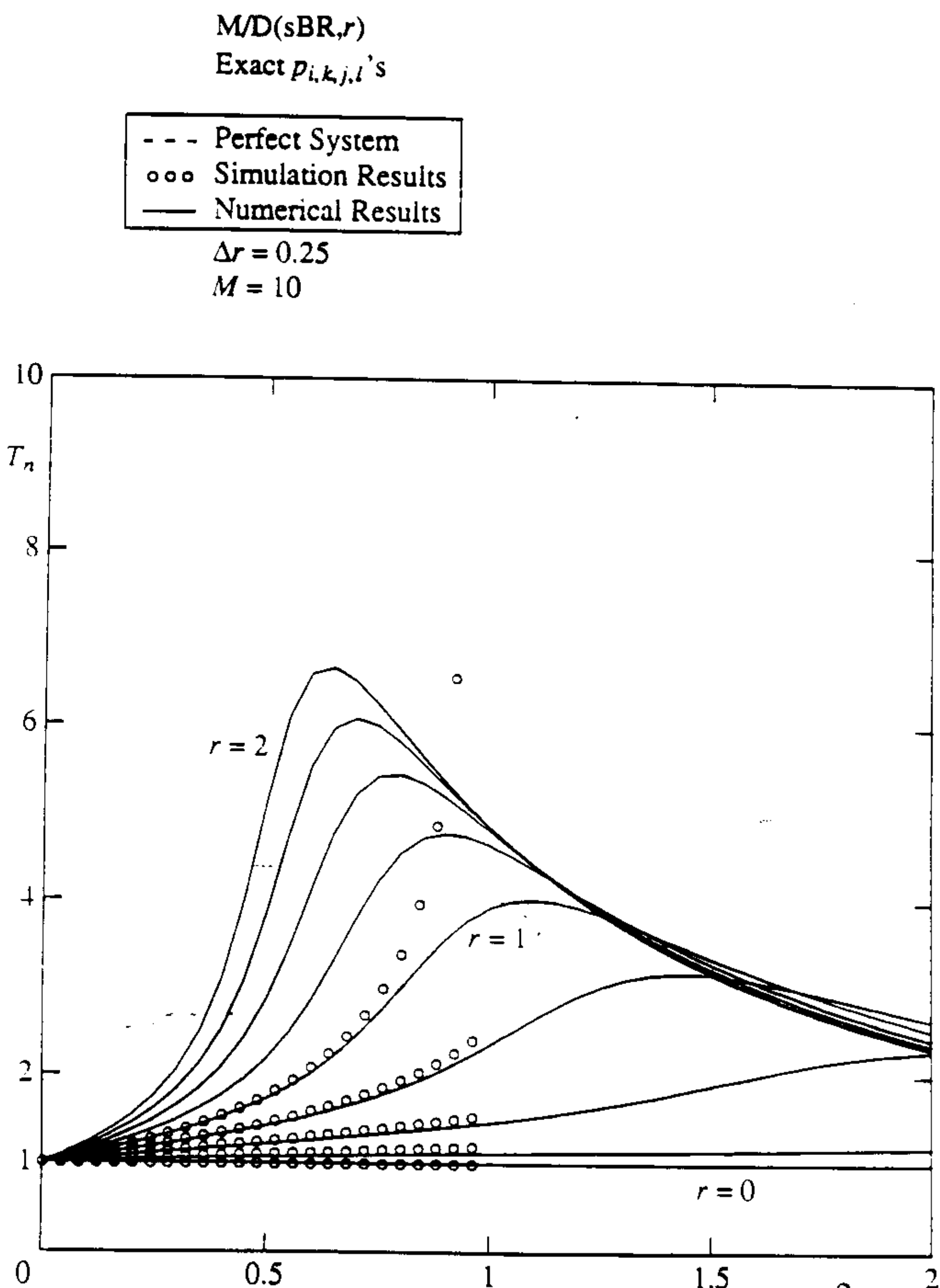


Figure 3.59: Normalized Average Response Time for M/D(sBR,r)

Domains for W_O in Equation (3.102)								
1	$0\leq q\leq 1$	$0\leq q_r\leq 1$	$r=0$	$i\geq 1$	$k\geq 0$	$j=i-1$	$l=0$	
2	$0\leq q<1$	$0\leq q_r<1$	$r>0$		$k=j-i+1$	$j\geq i-1$	$l\geq 0$	$q<rq_r$
3								$q\geq rq_r$
4		$q_r=1$						$q<rq_r$
5	$0\leq q\leq 1$							$q\geq rq_r$
6	$q=1$	$0\leq q_r<1$						$q<rq_r$
7								$q\geq rq_r$
8		$q_r=1$						$q<rq_r$
9	otherwise							

<u>Domain</u>

1) $$W_O = 1$$

2) $$W_O = \frac{i\rho^l}{l!}e^{-rq_r\rho}\sum_{m=0}^{l}\binom{l}{m}(-p)^m\frac{e^{m(q-rq_r)/p}}{i+k+rp_r(m/p+\rho)}$$

3) $$W_O = \frac{i}{rp_r\rho}\left\{e^{-rq_r\rho}\sum_{m=0}^{l}\frac{(rq_r\rho)^m}{m!\;[1+(i+k)/(rp_r\rho)]^{l-m+1}} - e^{-(i+k)(q-rq_r)/(rp_r)-q\rho}\sum_{m=0}^{l}\frac{(q\rho)^m}{m!\;[1+(i+k)/(rp_r\rho)]^{l-m+1}}\right\} + \frac{i\rho^l}{l!}e^{-(i+k)(q-rq_r)/(rp_r)-q\rho}\sum_{m=0}^{l}\frac{\binom{l}{m}(-p)^m}{i+k+rp_r(m/p+\rho)}$$

4) $$W_O = \frac{\rho^l[1-pe^{(q-rq_r)/p}]^l}{l!}e^{-rq_r\rho}$$

5) $$W_O = \frac{(rq_r\rho)^l}{l!}e^{-rq_r\rho}$$

6) $$W_O = \frac{i\rho^l}{l!}\cdot\frac{e^{-rq_r\rho}}{i+k+rp_r\rho}$$

$$7)\quad W_O = \frac{i}{rp_r\rho}\left\{e^{-rq_r\rho}\sum_{m=0}^{l}\frac{(rq_r\rho)^m}{m!\,[1+(i+k)/(rp_r\rho)]^{l-m+1}} - e^{-(i+k)(q-rq_r)/(rp_r)-q\rho}\sum_{m=0}^{l}\frac{(q\rho)^m}{m!\,[1+(i+k)/(rp_r\rho)]^{l-m+1}}\right\} + \frac{i\rho^l}{l!}\cdot\frac{e^{-(i+k)(q-rq_r)/(rp_r)-q\rho}}{i+k+rp_r\rho}$$

$$8)\quad W_O = \frac{\rho^l}{l!}e^{-rq_r\rho}$$

$$9)\quad W_O = 0 \qquad (3.102)$$

Domains for W_I in Equation (3.103)								
1	$0 \le q \le 1$	$0 \le q_r \le 1$	$r=0$	$i=0$	$k \ge 1$	$j=k-1$	$l=0$	
2	$0 \le q < 1$	$0 \le q_r < 1$	$r>0$	$i \ge 0$	$k=j-i+1$	$j \ge i$	$l \ge 0$	$q < rq_r$
3								$q \ge rq_r$
4		$q_r = 1$		$i=0$				$q < rq_r$
5	$0 \le q \le 1$							$q \ge rq_r$
6	$q=1$	$0 \le q_r < 1$		$i \ge 0$				$q < rq_r$
7								$q \ge rq_r$
8		$q_r = 1$		$i=0$				$q < rq_r$
9	otherwise							

Domain

$$1)\quad W_I = 1$$

$$2)\quad W_I = \frac{(j-i+1)\rho^l}{l!}e^{-rq_r\rho}\sum_{m=0}^{l}\binom{l}{m}(-p)^m\frac{e^{m(q-rq_r)/p}}{i+k+rp_r(m/p+\rho)}$$

$$
\begin{aligned}
3) \quad W_I &= \frac{j-i+1}{rp_r\rho}\left\{e^{-rq_r\rho}\sum_{m=0}^{l}\frac{(rq_r\rho)^m}{m!\,[1+(i+k)/(rp_r\rho)]^{l-m+1}}\right. \\
&\quad - \left. e^{-(i+k)(q-rq_r)/(rp_r)-q\rho}\sum_{m=0}^{l}\frac{(q\rho)^m}{m!\,[1+(i+k)/(rp_r\rho)]^{l-m+1}}\right\} \\
&\quad + \frac{(j-i+1)\rho^l}{l!}e^{-(i+k)(q-rq_r)/(rp_r)-q\rho}\sum_{m=0}^{l}\binom{l}{m}(-p)^m \\
&\quad \cdot \frac{1}{i+k+rp_r(m/p+\rho)} \\
4) \quad W_I &= \frac{\rho^l[1-pe^{(q-rq_r)/p}]^l}{l!}e^{-rq_r\rho} \\
5) \quad W_I &= \frac{(rq_r\rho)^l}{l!}e^{-rq_r\rho} \\
6) \quad W_I &= \frac{(j-i+1)\rho^l}{l!}\cdot\frac{e^{-rq_r\rho}}{i+k+rp_r\rho} \\
7) \quad W_I &= \frac{j-i+1}{rp_r\rho}\left\{e^{-rq_r\rho}\sum_{m=0}^{l}\frac{(rq_r\rho)^m}{m!\,[1+(i+k)/(rp_r\rho)]^{l-m+1}}\right. \\
&\quad - \left. e^{-(i+k)(q-rq_r)/(rp_r)-q\rho}\sum_{m=0}^{l}\frac{(q\rho)^m}{m!\,[1+(i+k)/(rp_r\rho)]^{l-m+1}}\right\} \\
&\quad + \frac{(j-i+1)\rho^l}{l!}\cdot\frac{e^{-(i+k)(q-rq_r)/(rp_r)-q\rho}}{i+k+rp_r\rho} \\
8) \quad W_I &= \frac{\rho^l}{l!}e^{-rq_r\rho} \\
9) \quad W_I &= 0
\end{aligned} \tag{3.103}
$$

Domains for W_N in Equation (3.104)									
1	$0 \leq q < 1$	$0 \leq q_r \leq 1$	$r=0$	$i=0$	$k=0$	$j \geq i$	$l=0$		
2	$q=1$								
3	$0 \leq q < 1$	$0 \leq q_r < 1$	$r>0$	$i \geq 0$	$k=j-i$		$l \geq 0$	$q < rq_r$	
4								$q \geq rq_r$	
5		$q_r=1$		$i=0$		$j=0$			
6				$i \geq 0$				$q < rq_r$	$j+i \geq 1$
7	$q=1$	$0 \leq q_r < 1$				$j \geq i$			
8								$q \geq rq_r$	
9		$q_r=1$	$r \geq 0$	$i=0$		$j=0$			
10				$i \geq 0$		$j \geq i$		$q < rq_r$	$j+i \geq 1$
11	otherwise								

Domain

1) $$W_N = \frac{\rho^{j+1}}{j!} e^{-q\rho} \sum_{m=0}^{j} \binom{j}{m} (-p)^m \left[\frac{1}{\rho + m/p} - \frac{1}{\rho + (m+1)/p} \right]$$

2) $$W_N = \frac{\rho^j}{j!} e^{-q\rho}$$

3) $$\begin{aligned} W_N = {} & \frac{\rho^{l+1}}{l!} \sum_{m=0}^{l} \binom{l}{m} (-p)^m \frac{e^{-q\rho} - e^{m(q-rq_r)/p - rq_r\rho}}{\rho + m/p} \\ & - \frac{\rho^{l+1}}{l!} \sum_{m=0}^{l} \binom{l}{m} (-p)^m \frac{e^{-q\rho} - e^{(m+1)(q-rq_r)/p - rq_r\rho}}{\rho + (m+1)/p} \\ & + \frac{\rho^{l+1}}{l!} e^{-rq_r\rho} \sum_{m=0}^{l} \binom{l}{m} (-p)^m e^{m(q-rq_r)/p} \\ & \cdot \left[\frac{1}{\rho + m/p + j/(rp_r)} - \frac{e^{(q-rq_r)/p}}{\rho + (m+1)/p + j/(rp_r)} \right] \end{aligned}$$

$$
\begin{aligned}
4)\quad W_N &= \frac{\rho^{l+1}}{l!} e^{-j(q-rq_r)/(rp_r)-q\rho} \sum_{m=0}^{l} \binom{l}{m} (-p)^m \\
&\cdot \left[\frac{1}{\rho+m/p+j/(rp_r)} - \frac{1}{\rho+(m+1)/p+j/(rp_r)} \right] \\
5)\quad W_N &= \frac{\rho^{l+1}}{l!} e^{-q\rho} \sum_{m=0}^{l} \binom{l}{m} (-p)^m \left[\frac{1}{\rho+m/p} - \frac{1}{\rho+(m+1)/p} \right] \\
6)\quad W_N &= \frac{\rho^{l+1}}{l!} \sum_{m=0}^{l} \binom{l}{m} (-p)^m \frac{e^{-q\rho} - e^{m(q-rq_r)/p-rq_r\rho}}{\rho+m/p} \\
&- \frac{\rho^{l+1}}{l!} \sum_{m=0}^{l} \binom{l}{m} (-p)^m \frac{e^{-q\rho} - e^{(m+1)(q-rq_r)/p-rq_r\rho}}{\rho+(m+1)/p} \\
7)\quad W_N &= \frac{\rho^{l+1}}{l!} (e^{-q\rho} - e^{-rq_r\rho}) + \frac{\rho^{l+1}}{l!} \cdot \frac{e^{-rq_r\rho}}{\rho+j/(rp_r)} \\
8)\quad W_N &= \frac{\rho^{l+1}}{l!} \cdot \frac{e^{-j(q-rq_r)/(rp_r)-q\rho}}{\rho+j/(rp_r)} \\
9)\quad W_N &= \frac{\rho^{l}}{l!} e^{-q\rho} \\
10)\quad W_N &= \frac{\rho^{l}}{l!} (e^{-q\rho} - e^{-rq_r\rho}) \\
11)\quad W_N &= 0 \qquad\qquad (3.104)
\end{aligned}
$$

Following the process of numerical calculation depicted in Figure 3.39, we calculate the normalized power P and normalized response time T_n versus load ρ in Figure 3.60 through 3.77.

The performance of the system gets worse with an increase of the restart-to-initial ratio r. The system performs better as the distribution of either the initial or restart service time becomes more deterministic, i.e., for higher q or $q = r$. For higher ρ the nature of the restart service times has far more effect than does the distribution of the initial service times since restarts become very frequent. The numerical results match the simulation results very well.

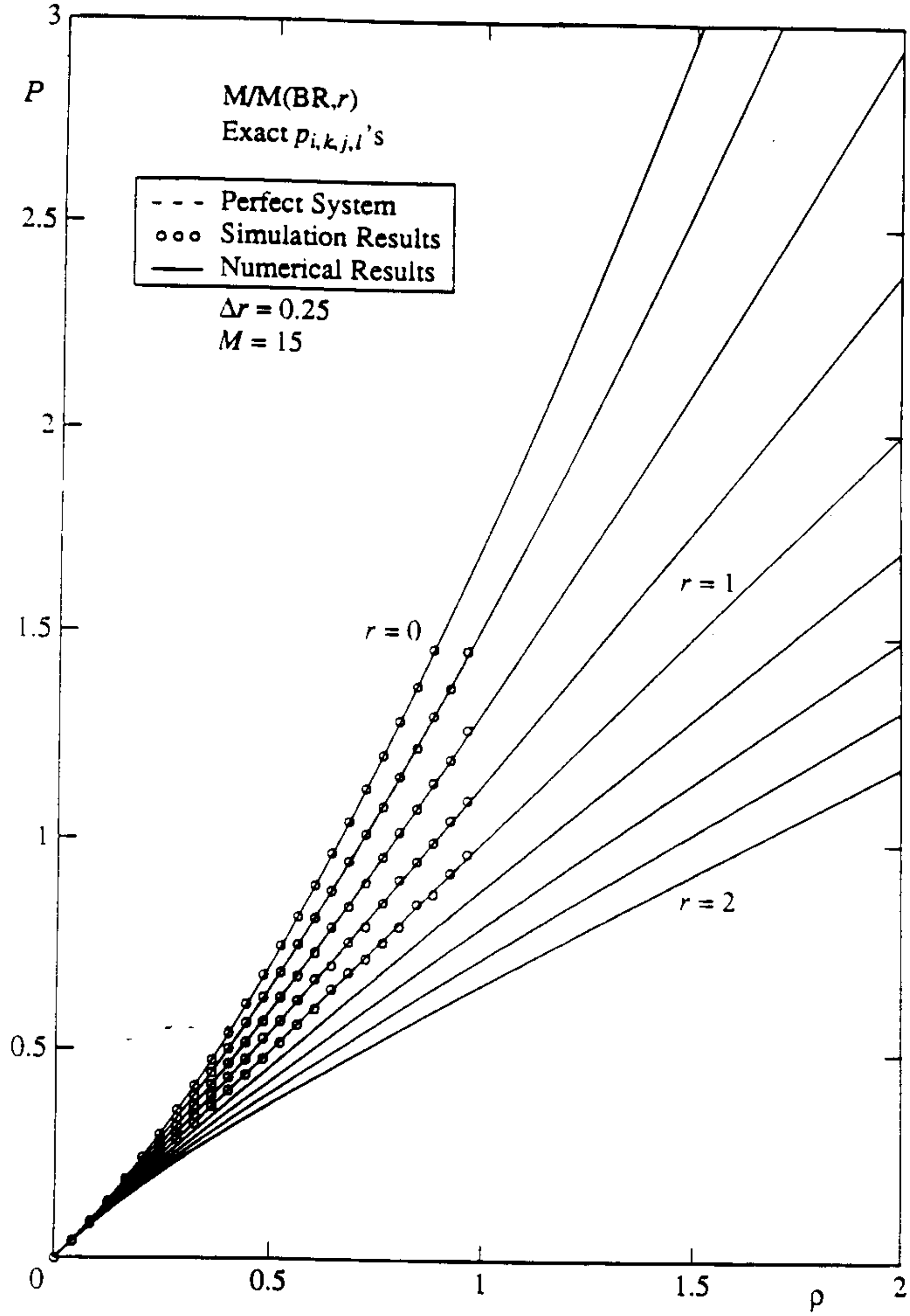


Figure 3.60: Normalized Power for M/M(BR.r)

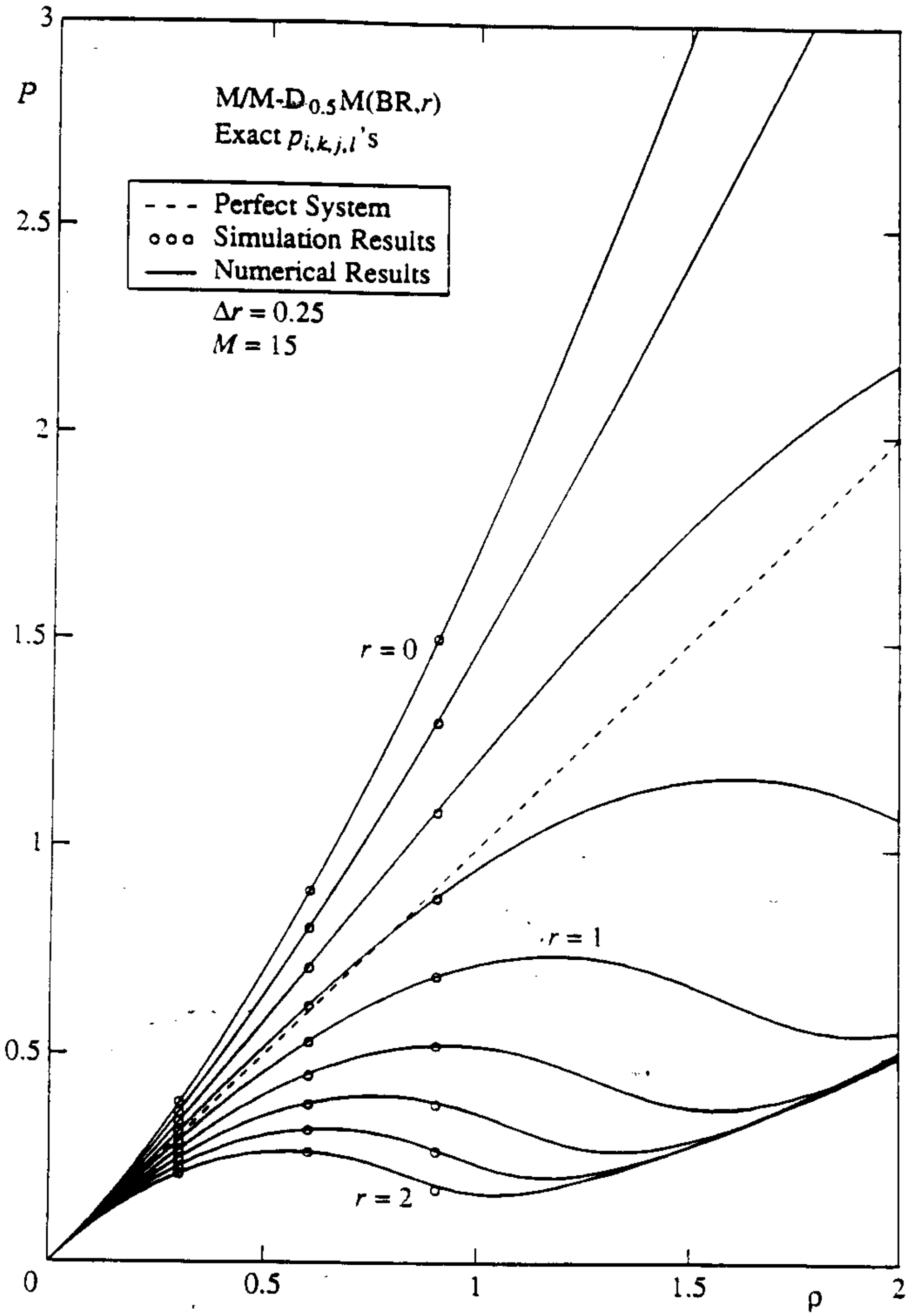


Figure 3.61: Normalized Power for M/M-D$_{0.5}$M(BR.r)

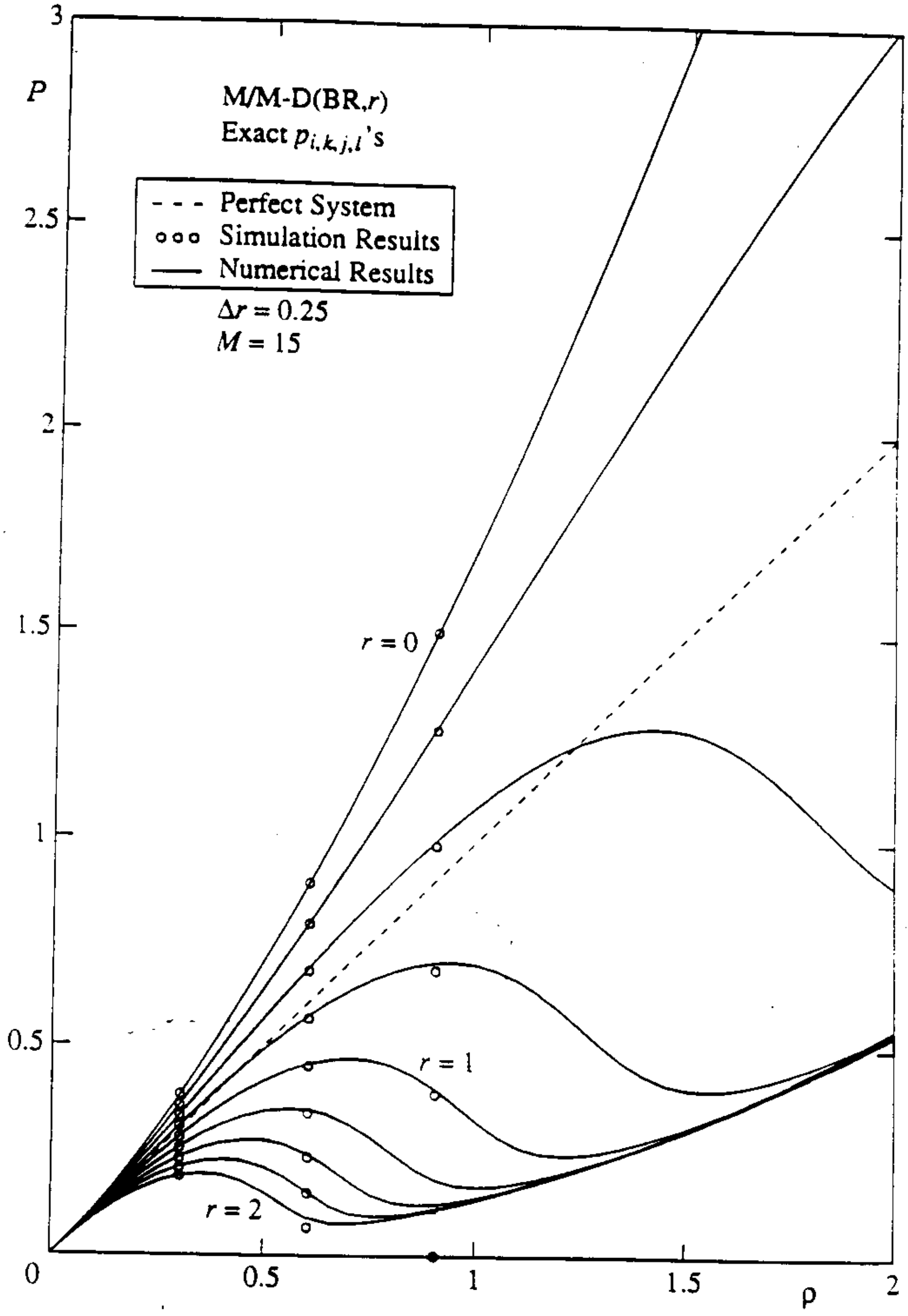


Figure 3.62: Normalized Power for M/M-D(BR,r)

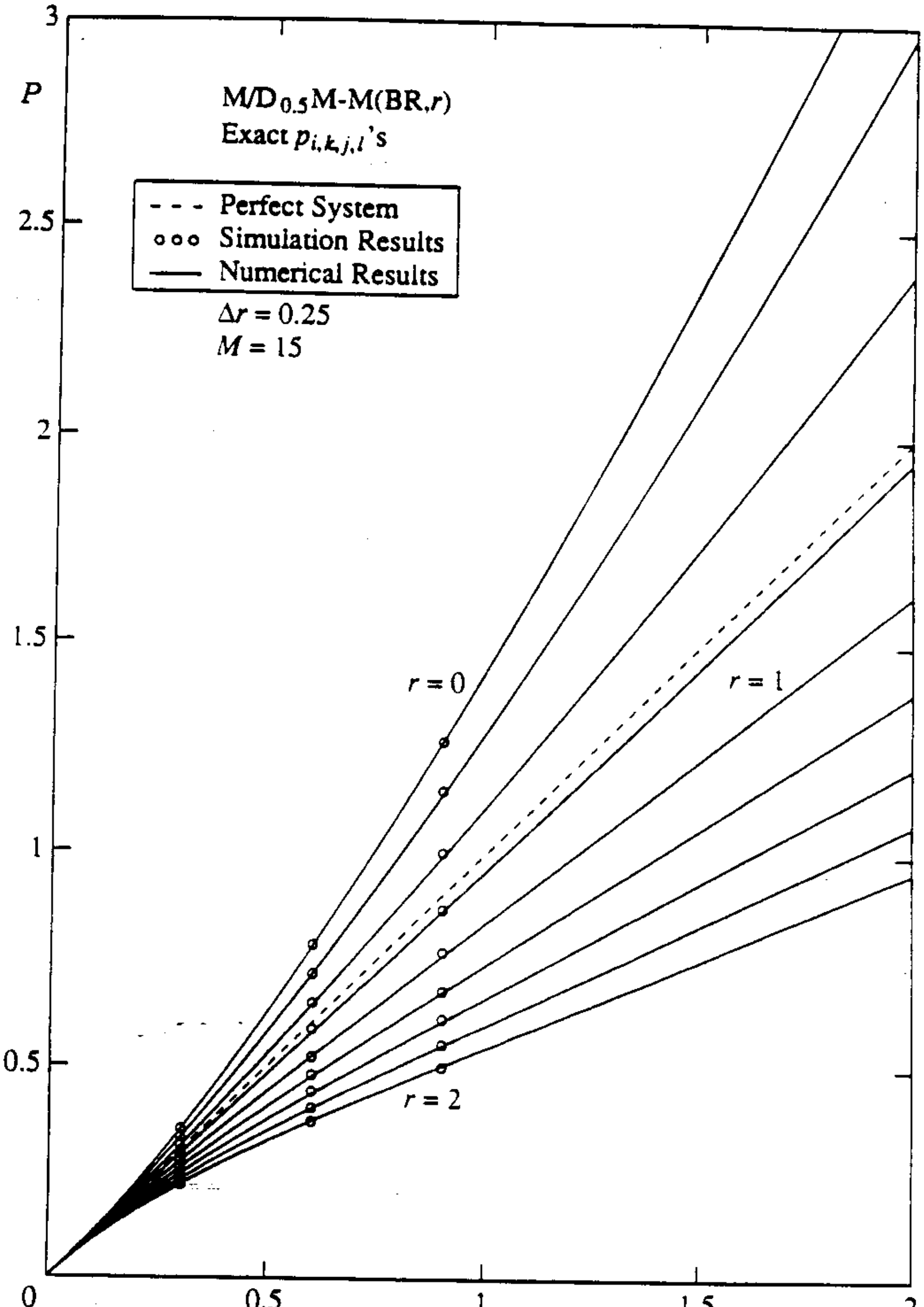


Figure 3.63: Normalized Power for $M/D_{0.5}M$-M(BR,r)

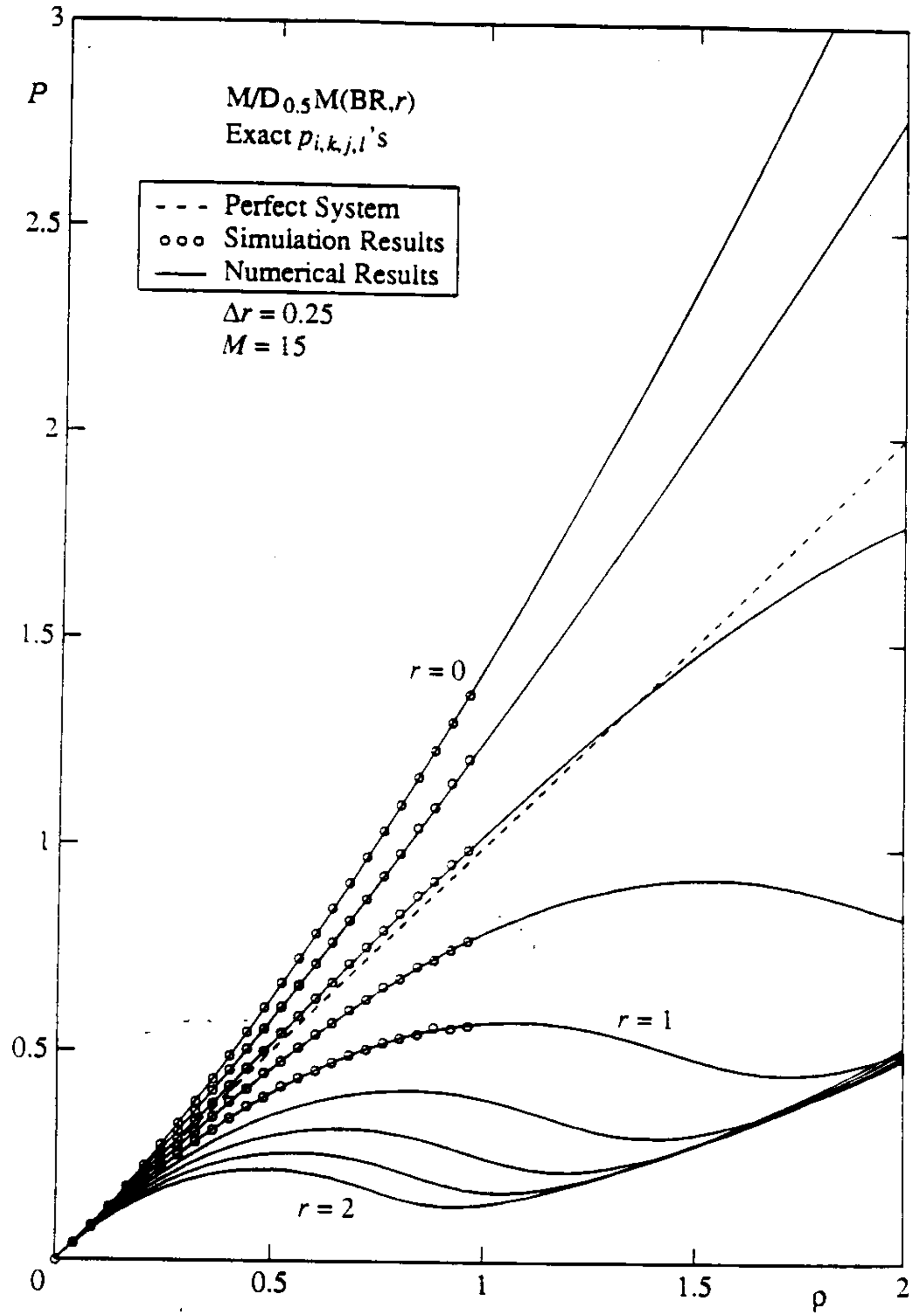


Figure 3.64: Normalized Power for $M/D_{0.5}M(BR,r)$

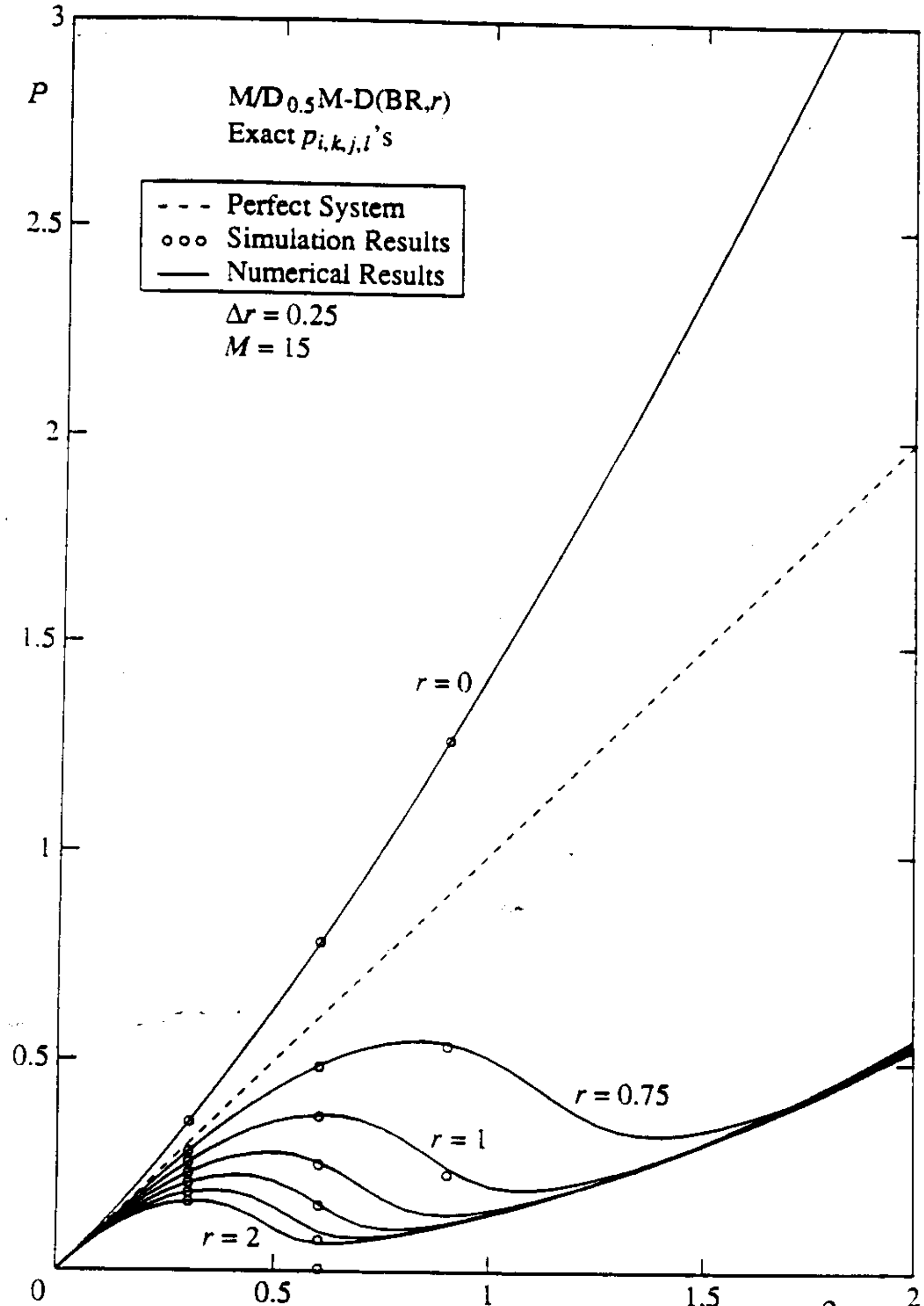


Figure 3.65: Normalized Power for $M/D_{0.5}M$-$D(BR.r)$

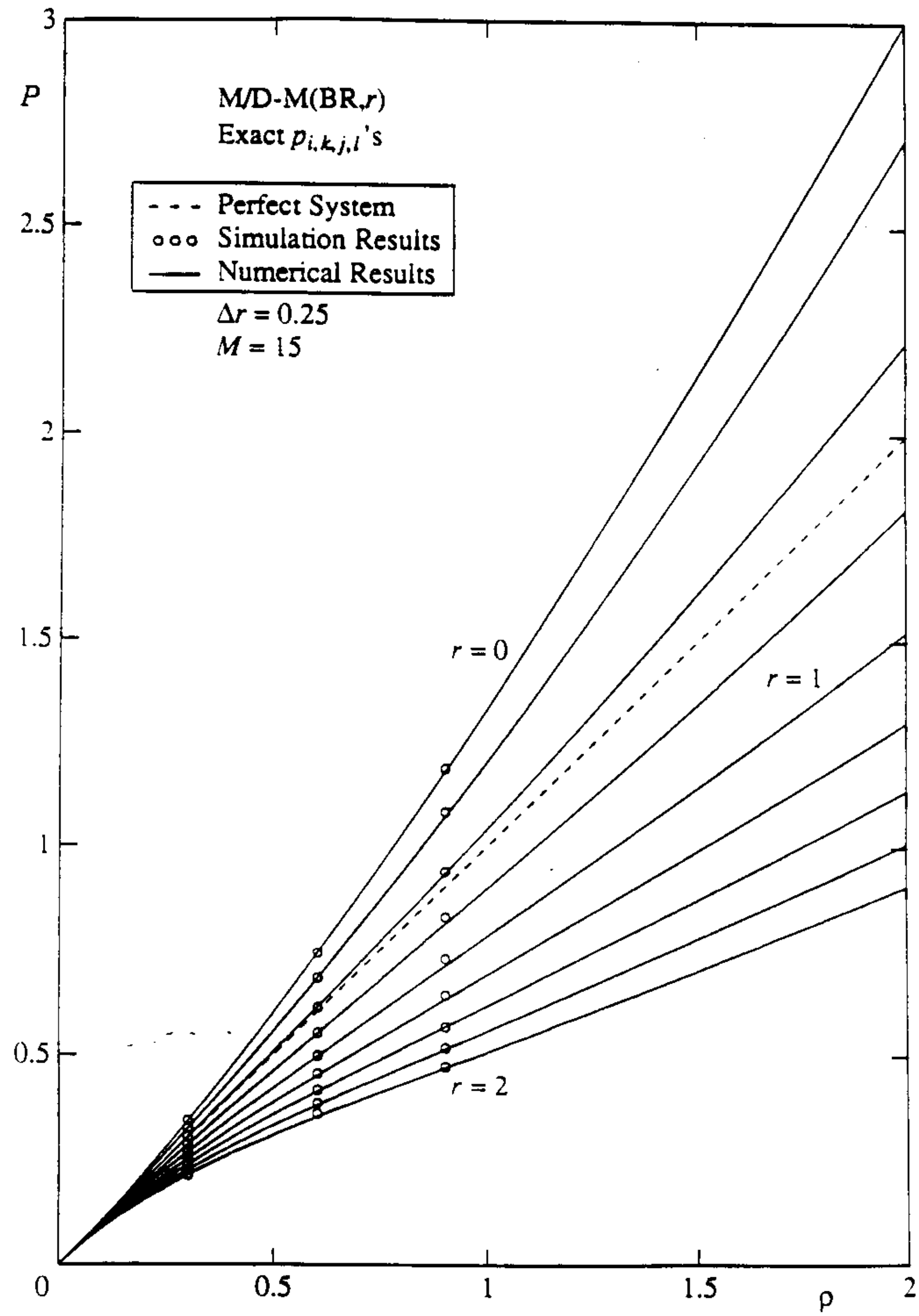


Figure 3.66: Normalized Power for M/D-M(BR,r)

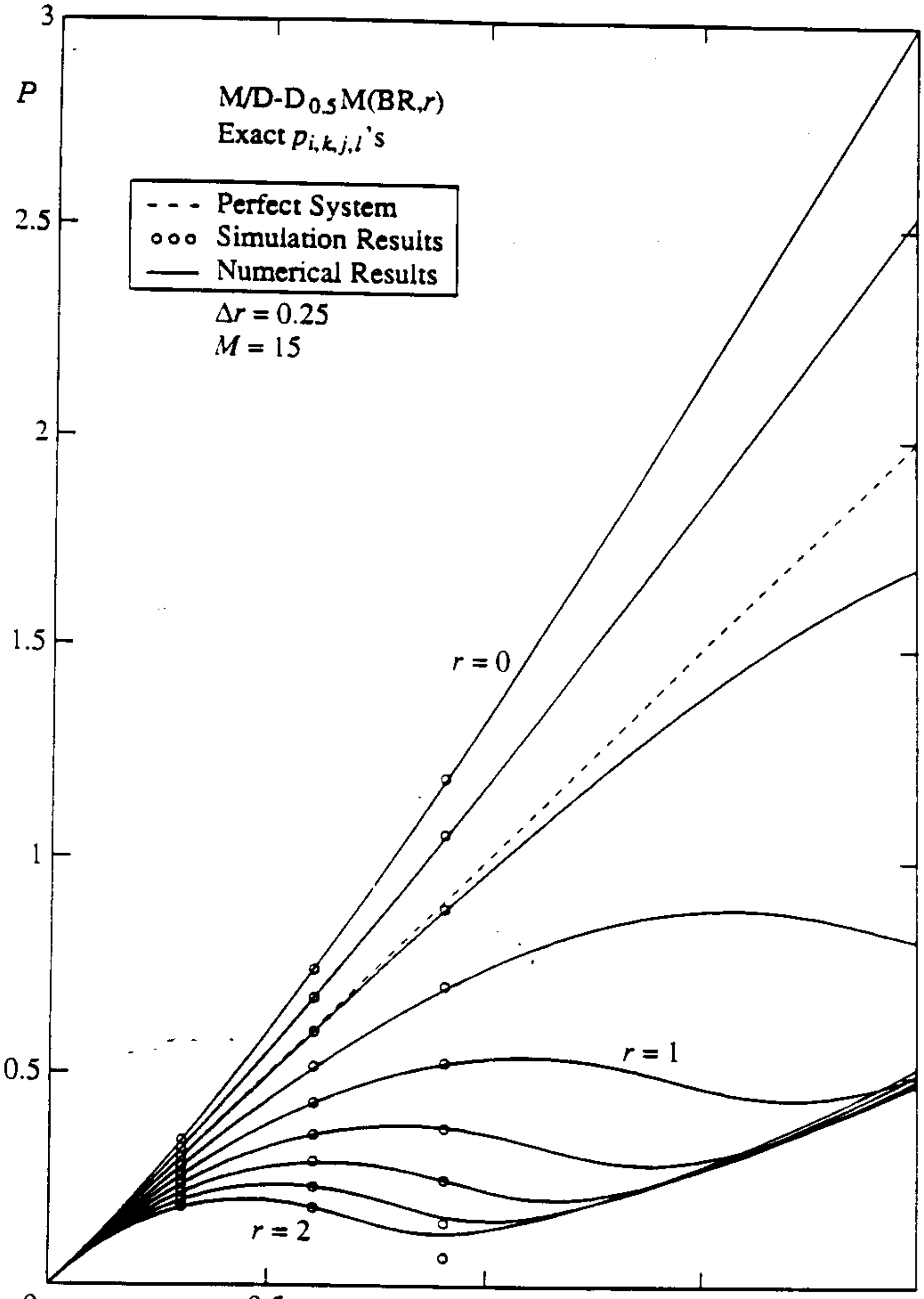


Figure 3.67: Normalized Power for M/D-$D_{0.5}$M(BR,r)

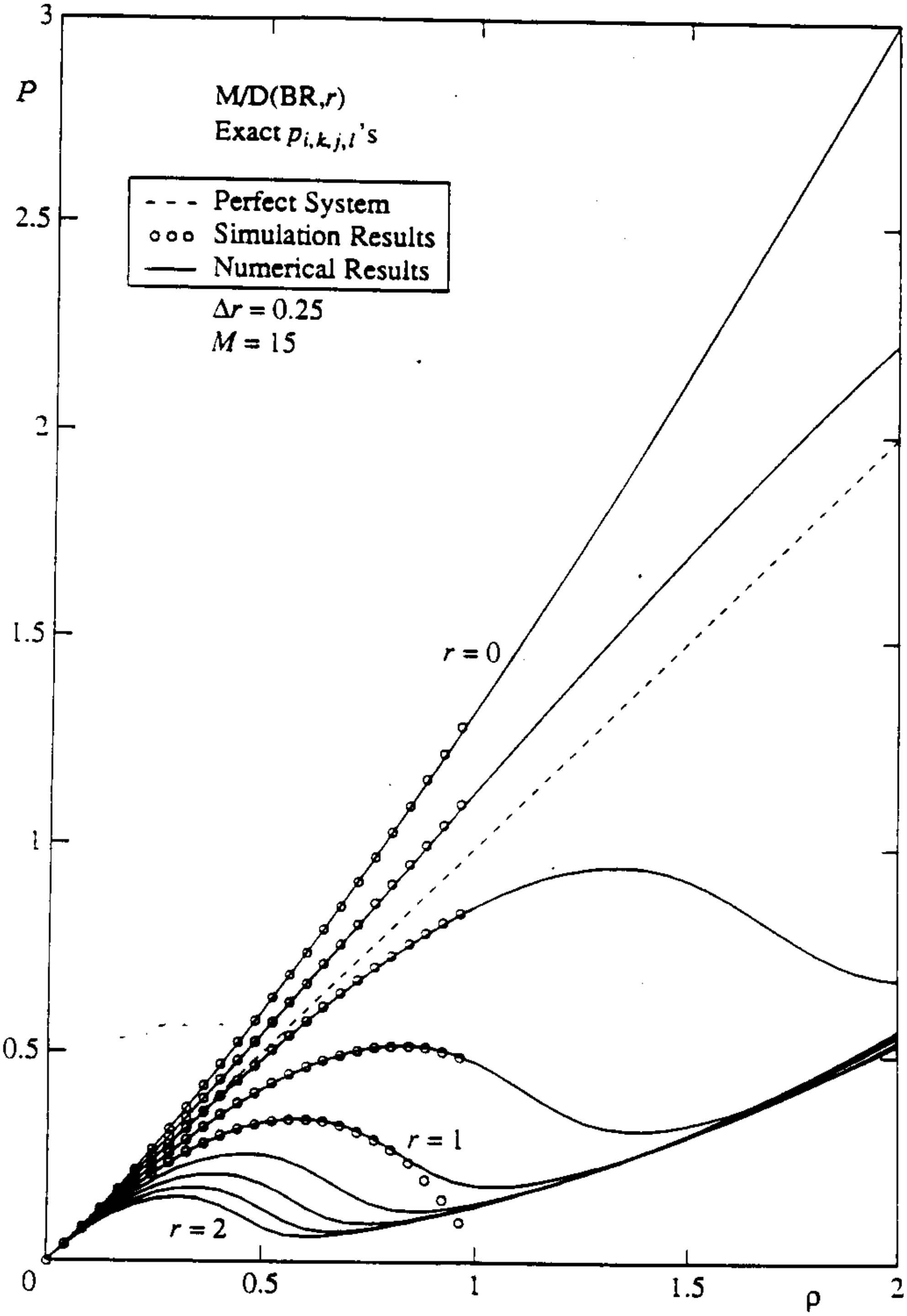


Figure 3.68: Normalized Power for M/D(BR,r)

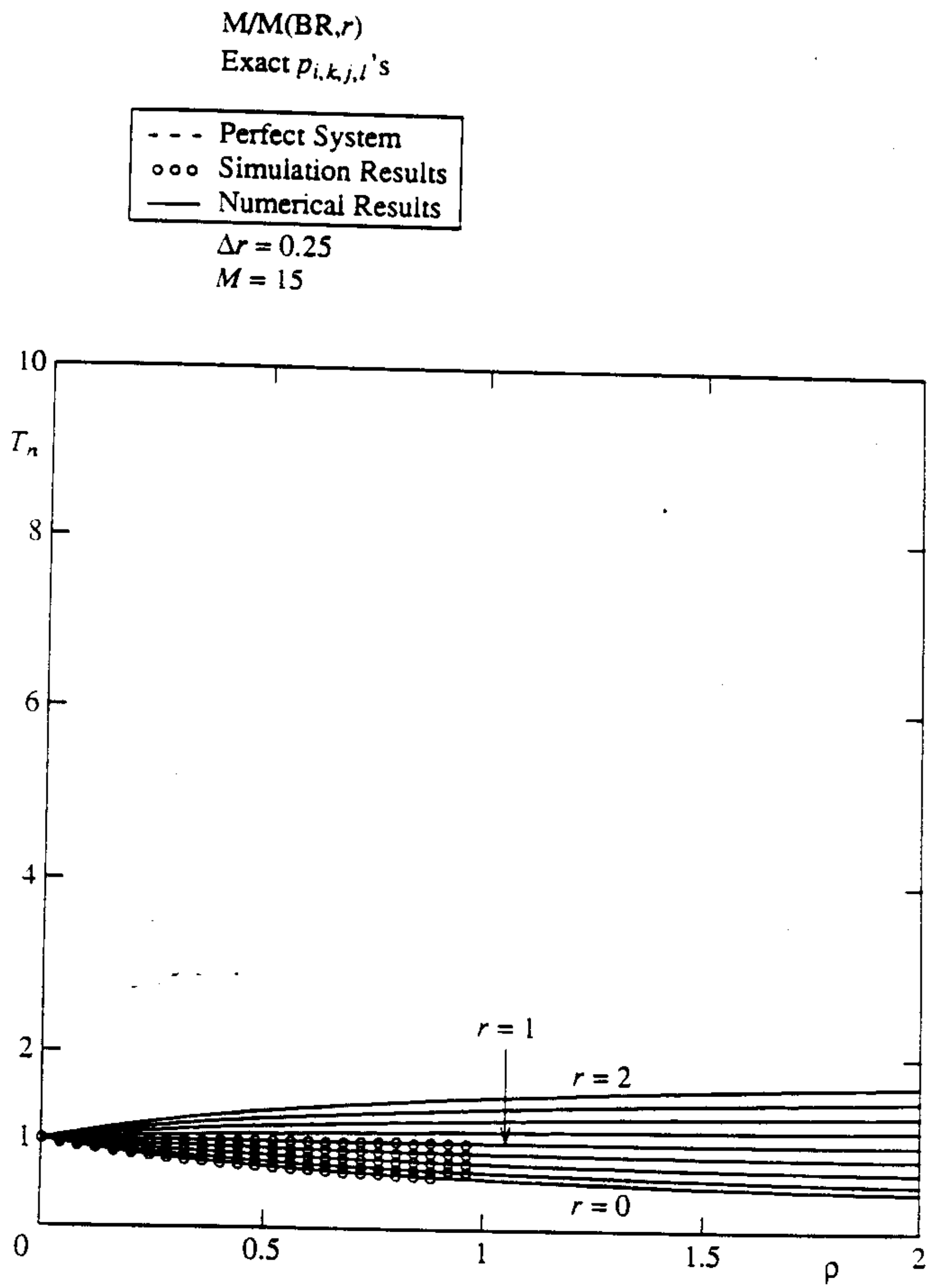


Figure 3.69: Normalized Average Response Time for M/M(BR.r)

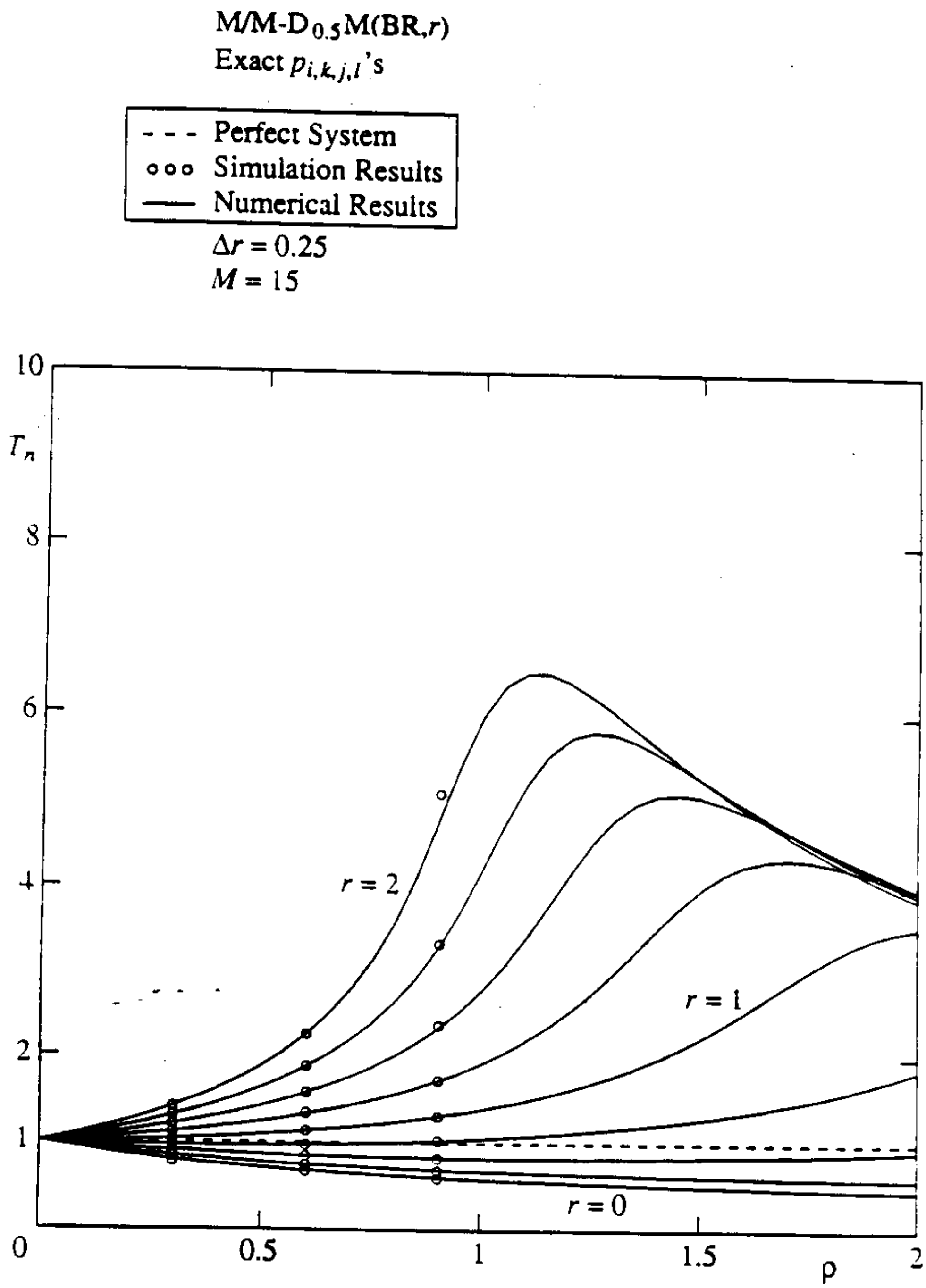


Figure 3.70: Normalized Average Response Time for M/M-$D_{0.5}$M(BR.r)

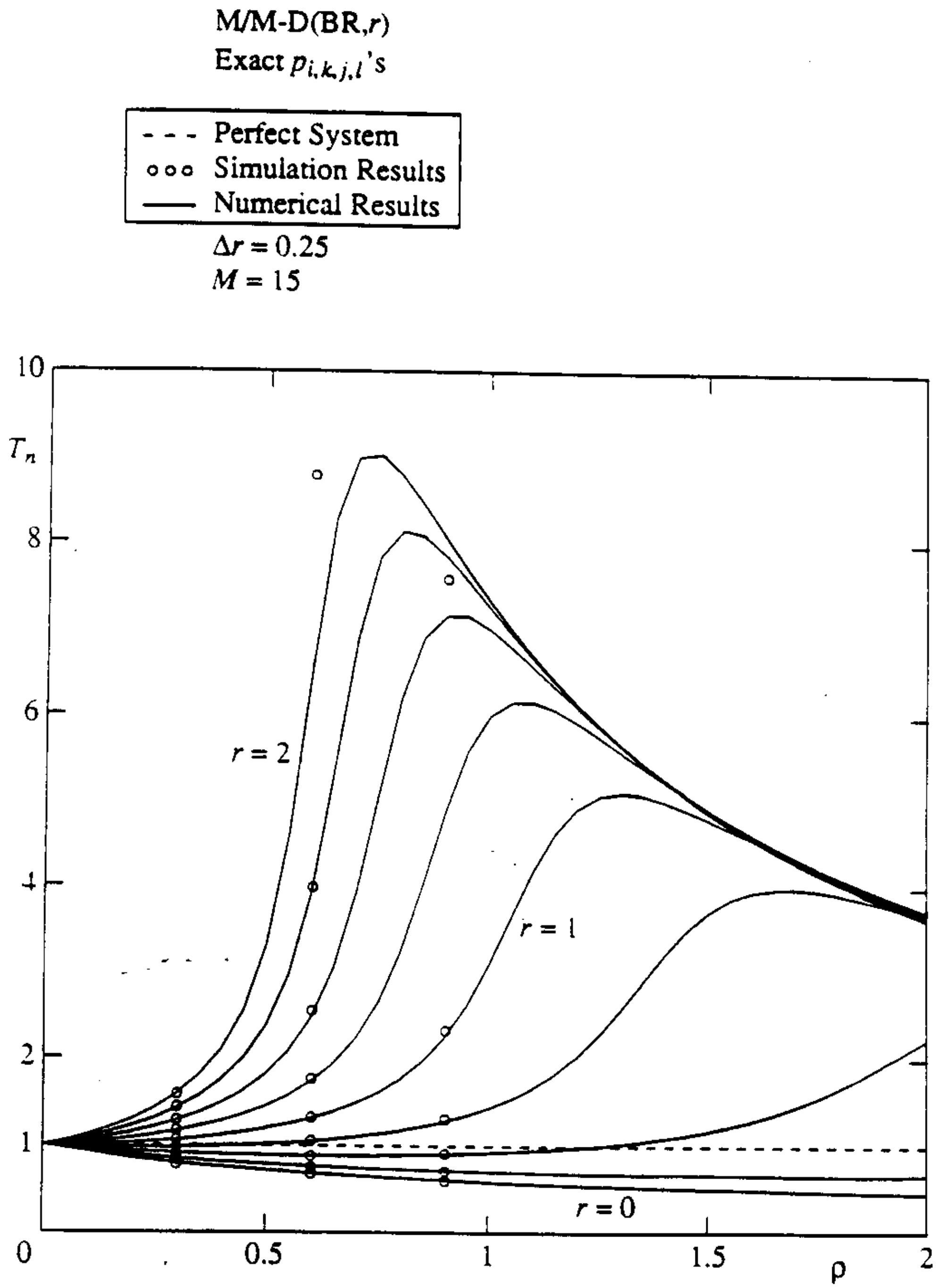


Figure 3.71: Normalized Average Response Time for M/M-D(BR.r)

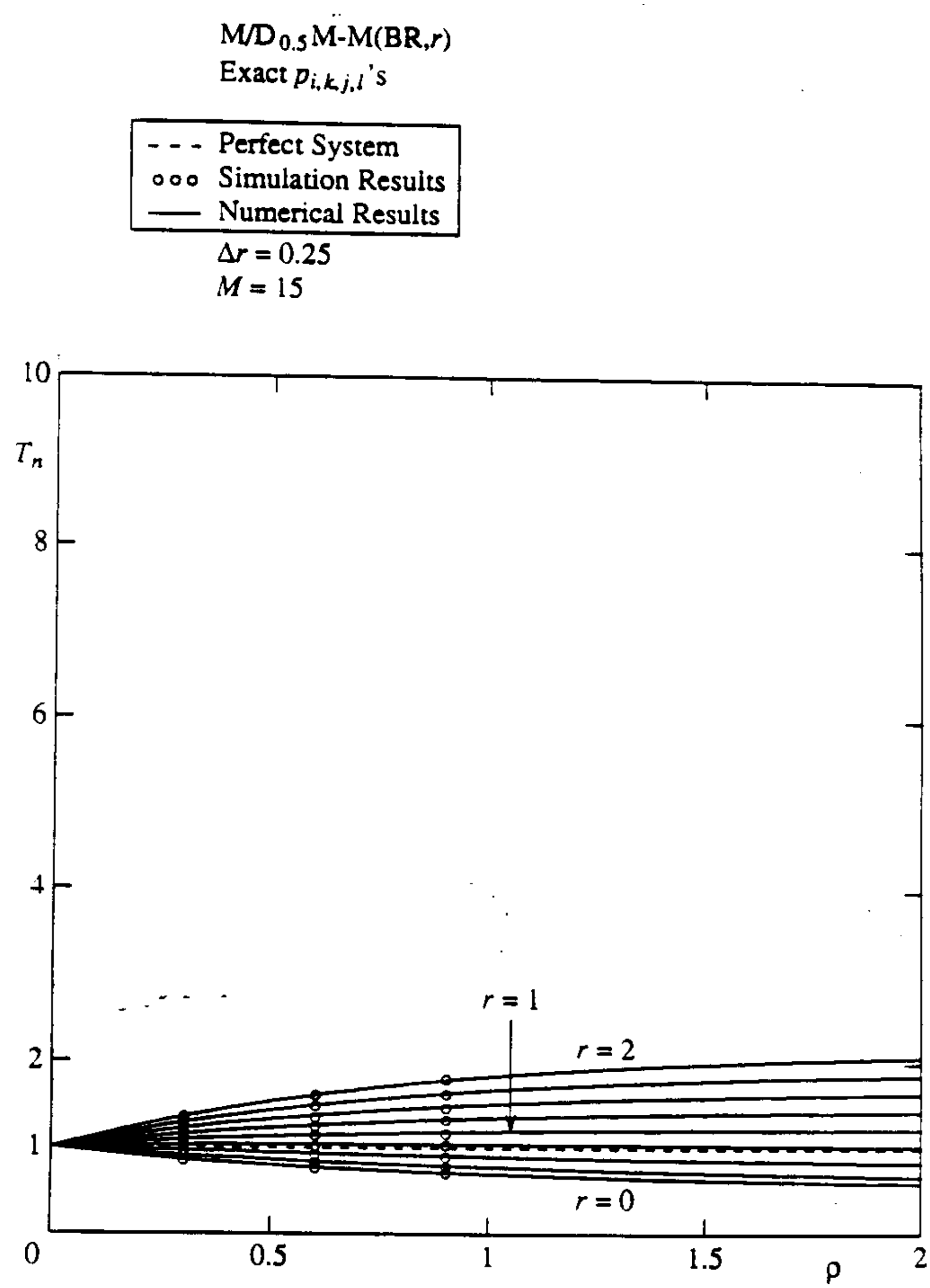


Figure 3.72: Normalized Average Response Time for $M/D_{0.5}M$-M(BR.r)

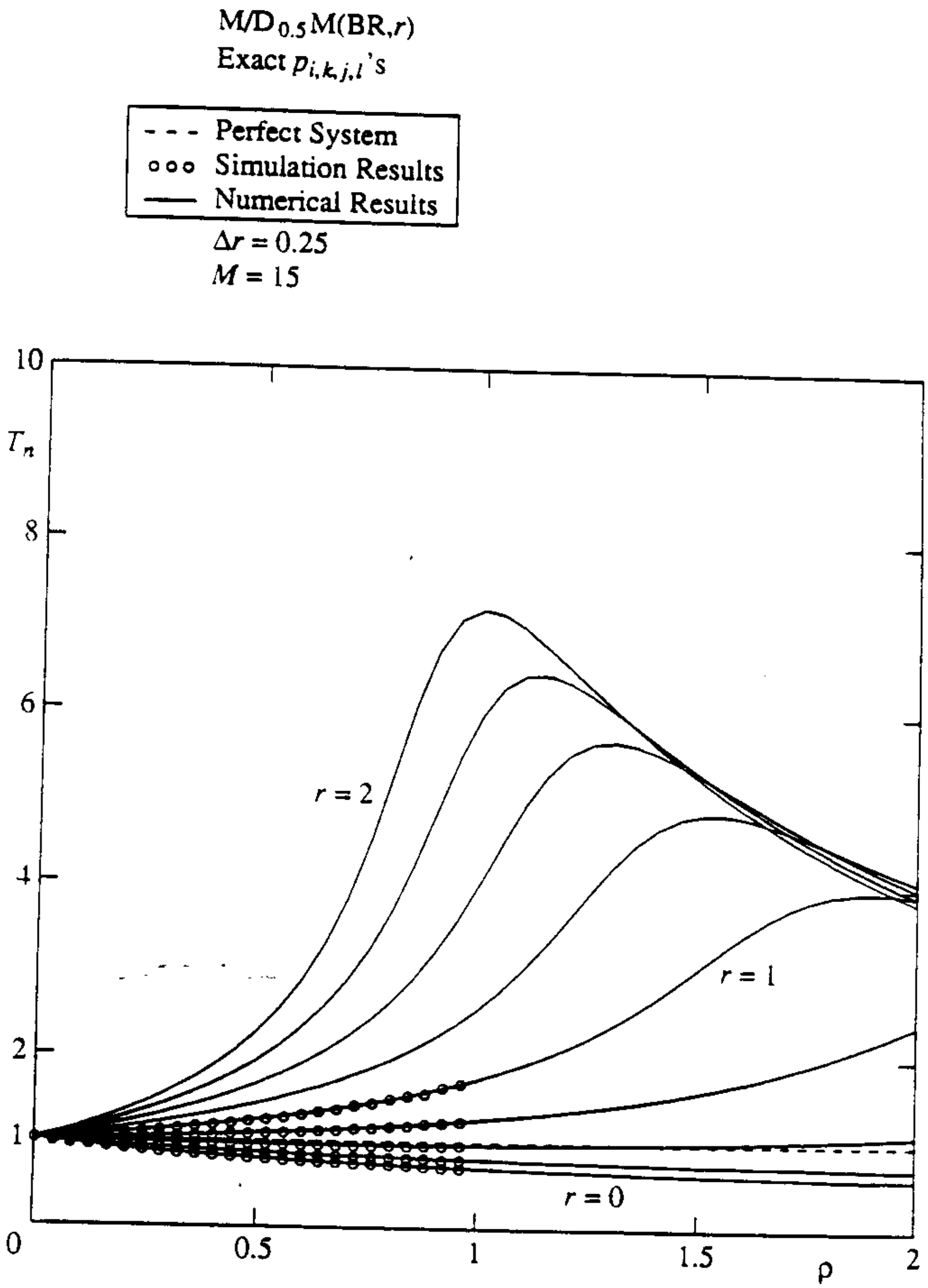


Figure 3.73: Normalized Average Response Time for M/D$_{0.5}$M(BR.r)

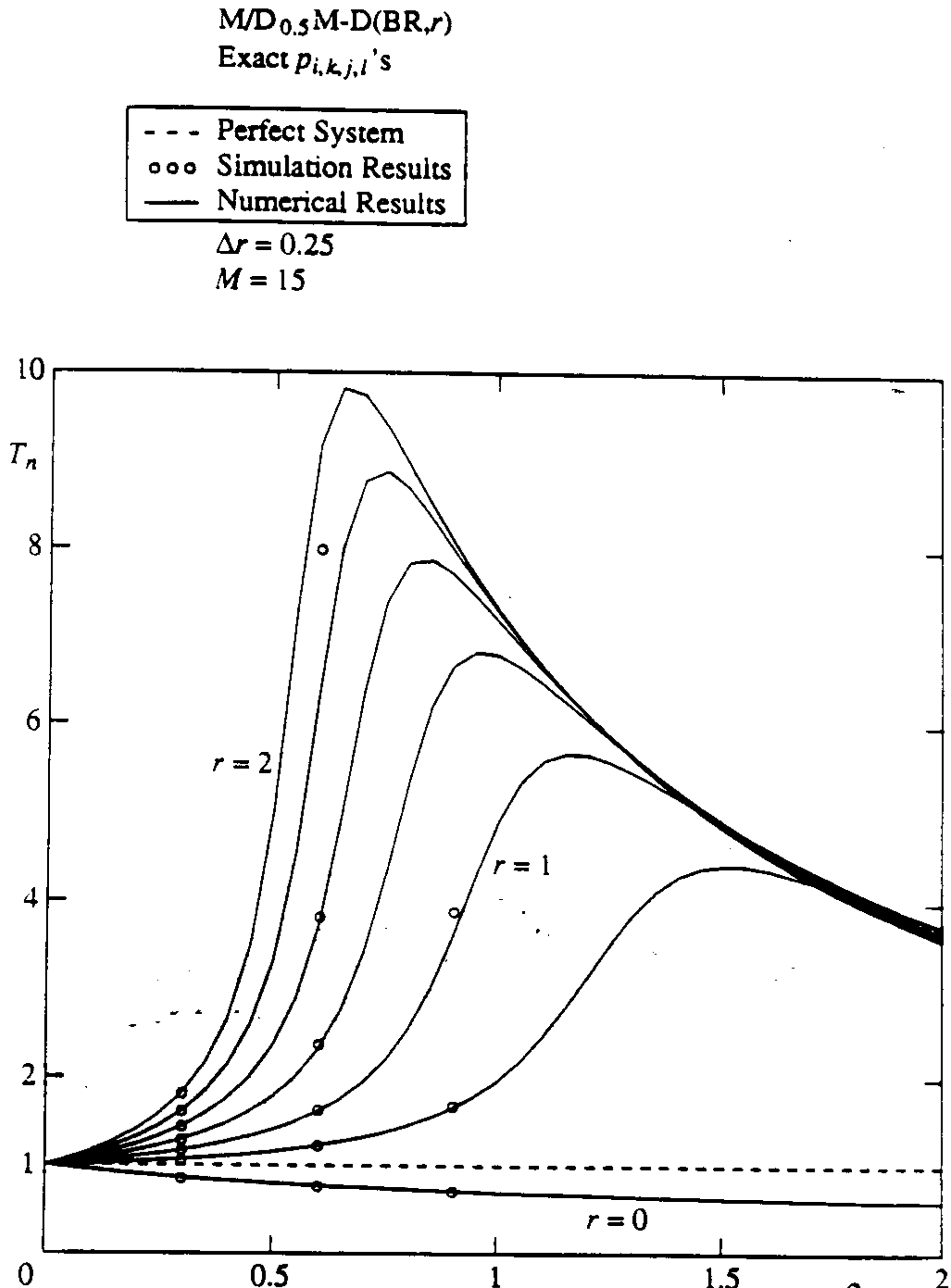


Figure 3.74: Normalized Average Response Time for M/$D_{0.5}$M-D(BR,r)

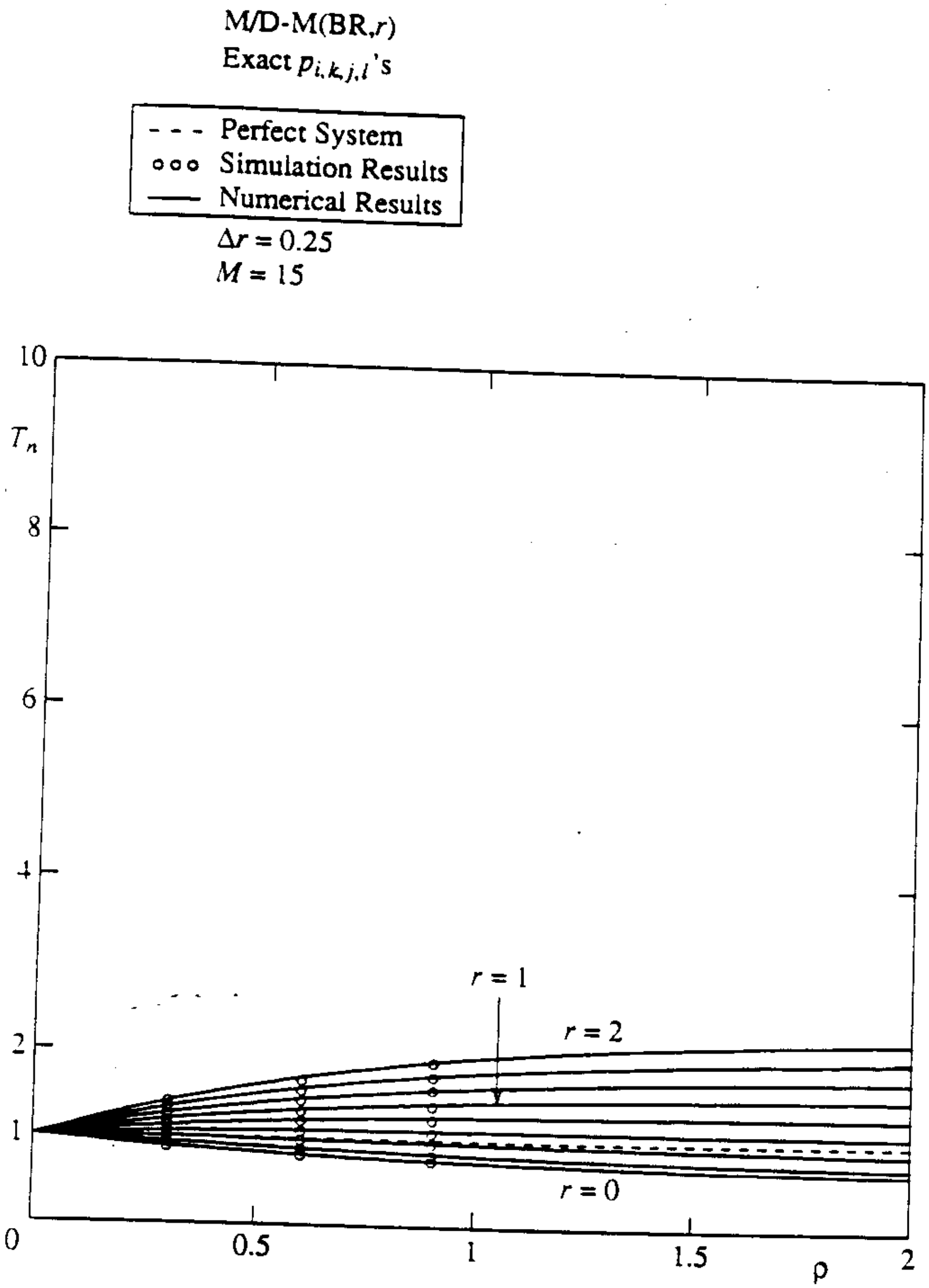


Figure 3.75: Normalized Average Response Time for M/D-M(BR.r)

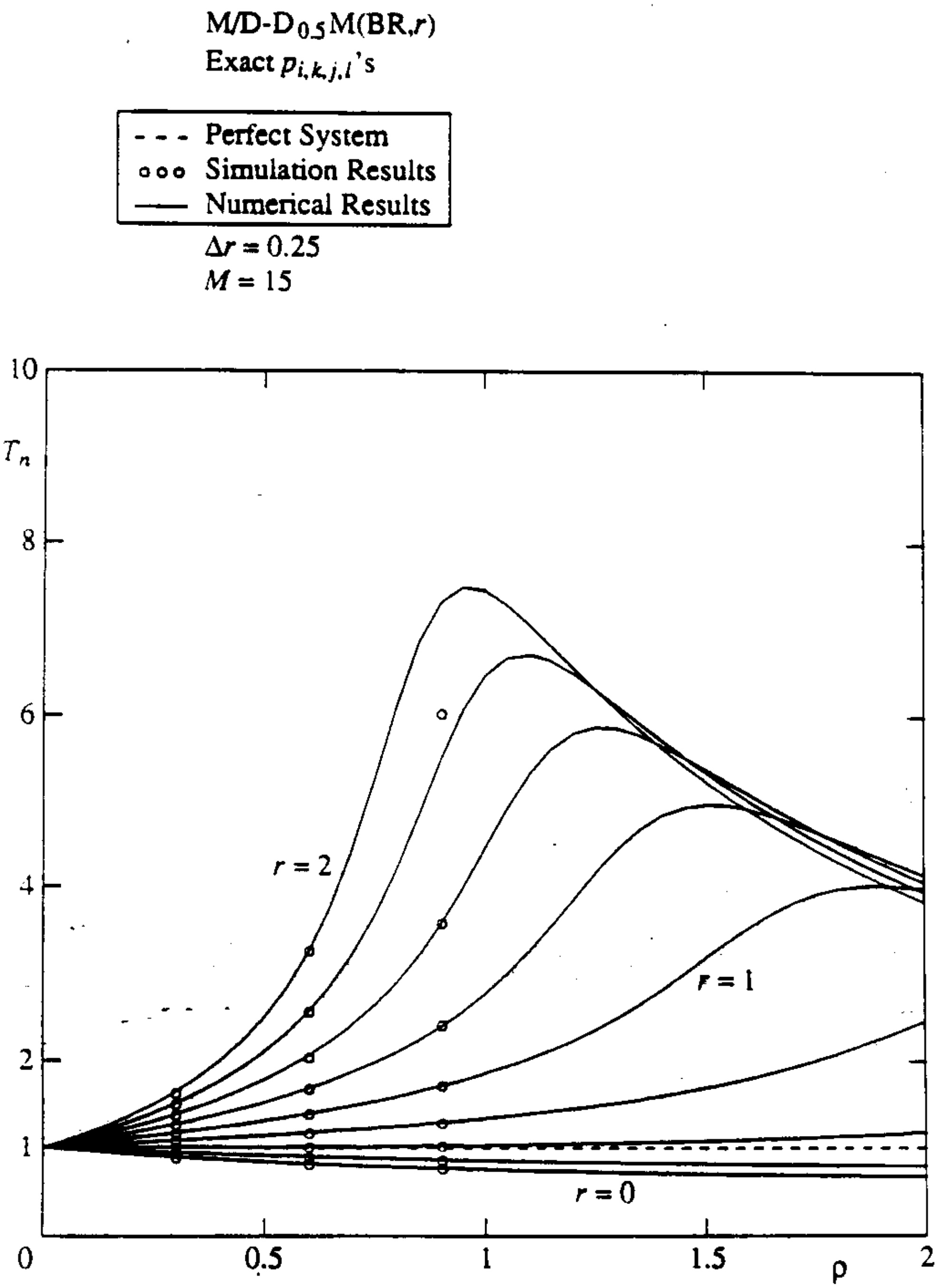


Figure 3.76: Normalized Average Response Time for M/D-D$_{0.5}$M(BR.r)

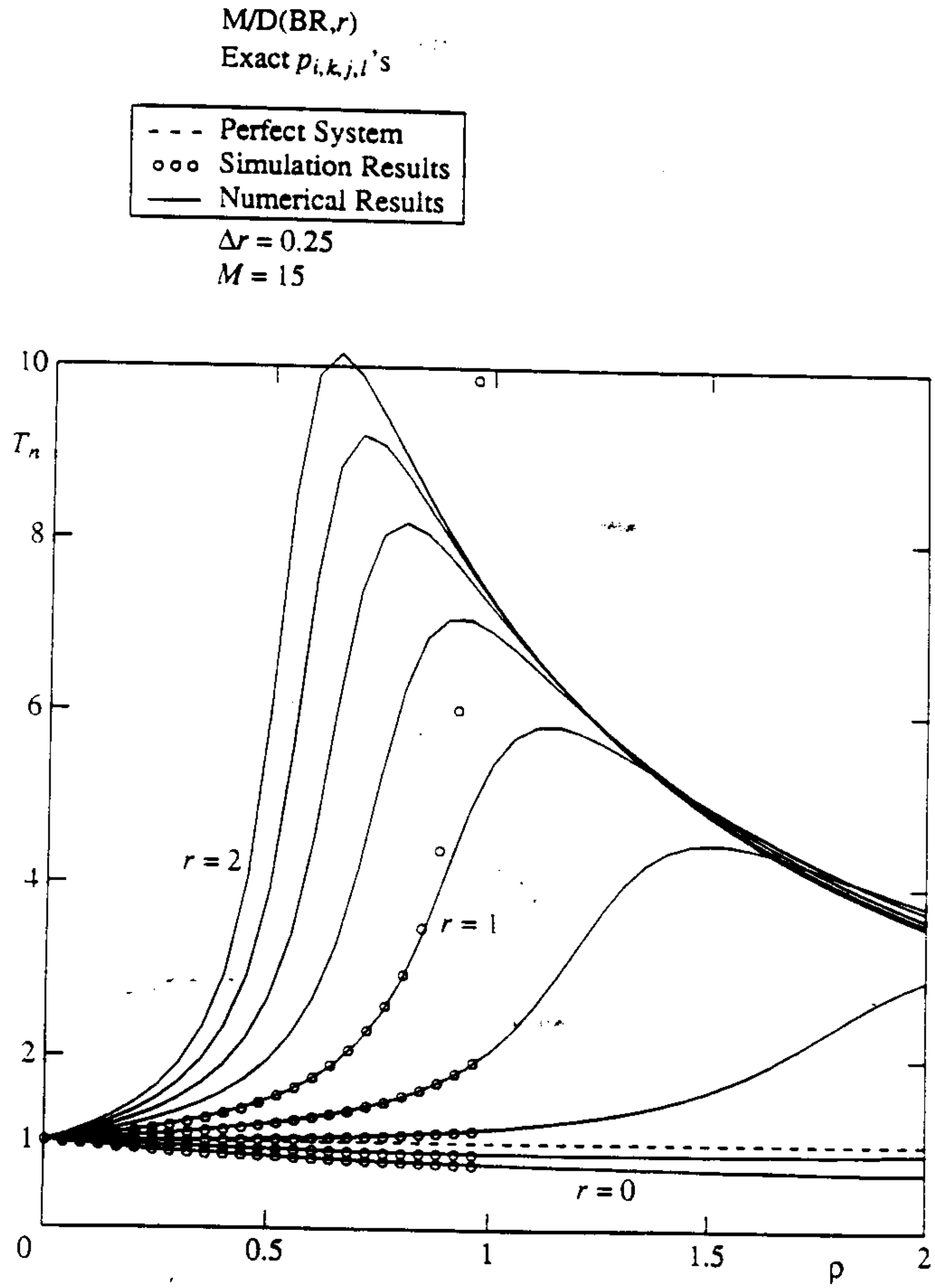


Figure 3.77: Normalized Average Response Time for M/D(BR.r)

3.3.11 Summary

In Sections 3.3.4 through 3.3.10 we have obtained numerical results for the winner queues with partial restarts M/M(SR,r), M/M-D(SR,r), M/D-M(SR,r), M/D(S,r), M/M-$D_{q,r}$M(sBR,r), M/D-$D_{q,r}$M(sBR,r), M/D(sB,r), and M/D_qM--$D_{q,r}$M(BR,r), respectively. Figure 3.78 shows graphically what systems are solved through the analysis of the winner queues with partial restarts.

3.4 Other Approaches and Other Systems

In this section we try to find analytic results for some of the full-conflict systems using different approaches.

3.4.1 System A1: M/M(SR) as a 2-D Markov Chain

Consider the M/M(SR) system at some point in time t. Let all the customers that can still win at time t be called *alive* customers. (Those are the customers that have restarted or have arrived since the last departure.) Let all the customers that will lose for sure be called *dead* customers. (Those are the old customers that haven't restarted yet since the last departure took place.) Every restart of a customer converts it from dead to alive. Every departure kills all the alive customers left in the system. Every arrival increases the number of alive customers by one. Figure 3.79 shows the model of the system. Customers arrive to the system and leave the system at the average rate λ. From the "ALIVE" box, bulks of customers move into the "DEAD" box at the average rate λ. The average rate of customers moving from the "ALIVE" box to "DEAD" box and vice versa is denoted γ (messages/sec).

We now consider a two-dimensional Markov chain with states defined as the number of alive and the number of dead customers. Figure 3.80 shows a portion of the transition rate state diagram of such a system, as used in [13]. [18] solves numerically a truncated version of a similar Markov chain. In [18] states represent

		M	D_qM	D
SR,r	M	E		A
	$D_{q,r}$M			
	D	A		A
SN,r				A
sBR,r	M	E		A
	$D_{q,r}$M	E		A
	D	E		A
sBN,r				A
BR,r	M	E	E	E
	$D_{q,r}$M	E	E	E
	D	E	E	E
BN,r				E
L				

Sys tem	Sec tion	Equa tion	Figure
P1	3.3.4	(3.84)	3.40 thru 3.41
P2	3.3.5	(3.87)	3.42 thru 3.43
P3	3.3.6	(3.90)	3.44 thru 3.45
P4	3.3.7	(3.94)	3.46 thru 3.47
P5	3.3.8	(3.97)	3.48 thru 3.53
P6	3.3.9	(3.100)	3.54 thru 3.59
P7	3.3.10	(3.103)	3.60 thru 3.77

□ numerical results

○ analytic results

E exact transition probabilities

A approximate transition probabilities

Figure 3.78: Overview Table of Results for Winner Queues with Partial Restarts

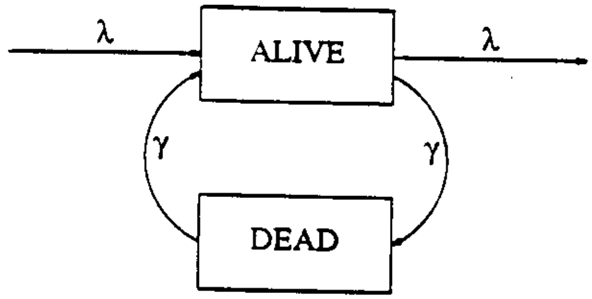


Figure 3.79: Dead/Alive Model

the number of alive customers and total number of customers in the system.

In this section we derive equations for the complete (infinite) Markov chain from Figure 3.81 which accounts for all the special cases of the Markov chain.

From parts a), b), c), and d) of Figure 3.81 we write the following four equations, respectively. The probability of the system being in state (i, j), (i.e., having i alive and j dead customers), we denote as $d_{i,j}, i, j = 0, 1, 2, \ldots$

$$d_{0,0}\lambda = d_{1,0}\mu, \quad i = 0, \ j = 0 \tag{3.105}$$

$$d_{i,0}(\lambda + i\mu) = d_{i-1,0}\lambda + d_{i-1,1}\mu, \quad i > 0, \ j = 0 \tag{3.106}$$

$$d_{i,j}[\lambda + (i+j)\mu] = d_{i-1,j}\lambda + d_{i-1,j+1}(j+1)\mu, \quad i > 0, \ j > 0 \tag{3.107}$$

$$d_{0,j}(\lambda + j\mu) = \sum_{k=0}^{j} d_{k+1,j-k}(k+1)\mu, \quad i = 0, \ j > 0 \tag{3.108}$$

We now try to find the double transform of the state probabilities $d(i, j), i, j = 0, 1, 2, \ldots$. The transform is defined as $P(y, z) = \sum_i \sum_j d_{i,j} y^i z^j$. To find the transform, we multiply Equation (3.105) through (3.108) by appropriate powers

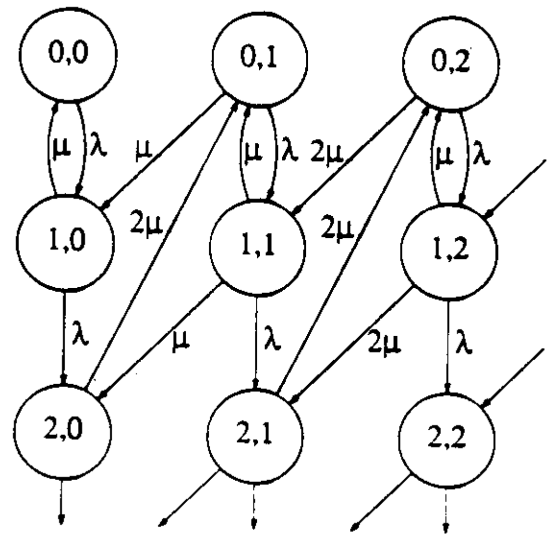


Figure 3.80: Dead/Alive Transition Rate State Diagram

of y and z, to obtain the following.

$$d_{0,0}\lambda = d_{1,0}\mu \tag{3.109}$$

$$\sum_{i=1}^{\infty} d_{i,0}(\lambda + i\mu)y^i = \sum_{i=1}^{\infty} d_{i-1,0}\lambda y^i + \sum_{i=1}^{\infty} d_{i-1,1}\mu y^i \tag{3.110}$$

$$\sum_{i=1}^{\infty}\sum_{j=1}^{\infty} d_{i,j}[\lambda + (i+j)\mu]y^i z^j = \sum_{i=1}^{\infty}\sum_{j=1}^{\infty} d_{i-1,j}\lambda y^i z^j + \sum_{i=1}^{\infty}\sum_{j=1}^{\infty} d_{i-1,j+1}(j+1)\mu y^i z^j \tag{3.111}$$

$$\sum_{j=1}^{\infty} d_{0,j}(\lambda + j\mu)z^j = \sum_{j=1}^{\infty}\sum_{k=1}^{j} d_{k+1,j-k}(k+1)\mu z^j \tag{3.112}$$

The sum of all the terms with λ on the left hand side of Equation (3.109) through (3.112) gives

$$d_{0,0}\lambda + \sum_{i=1}^{\infty} d_{i,0}\lambda y^i + \sum_{i=1}^{\infty}\sum_{j=1}^{\infty} d_{i,j}\lambda y^i z^j + \sum_{j=1}^{\infty} d_{0,j}\lambda z^j = \lambda P(y,z)$$

The sum of all the terms with $i\mu$ on the left hand side of Equation (3.109)

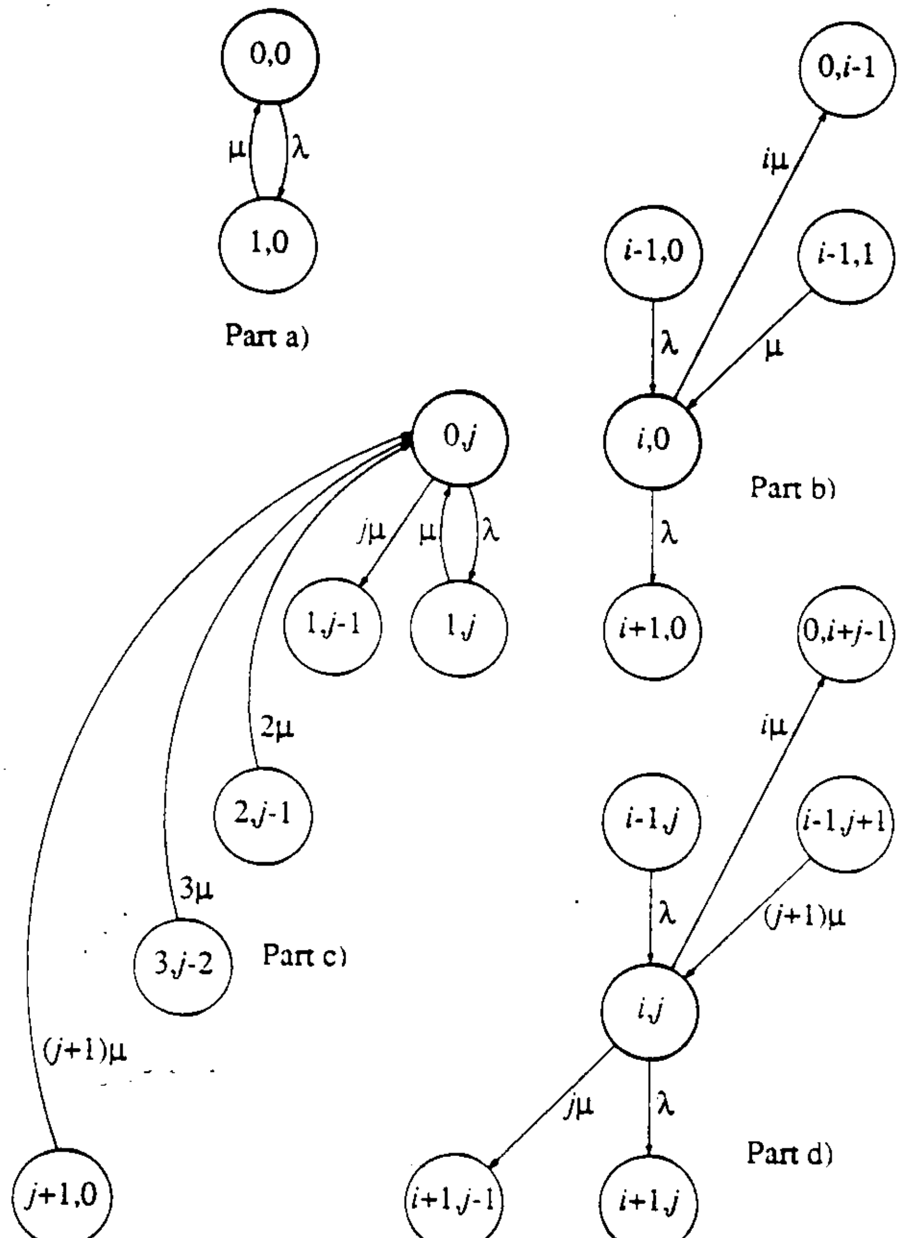


Figure 3.81: The Four Cases in Dead/Alive Transition Rate State Diagram

through (3.112) gives

$$\begin{aligned}
\sum_{i=1}^{\infty} d_{i,0} i\mu y^i + \sum_{i=1}^{\infty}\sum_{j=1}^{\infty} d_{i,j} i\mu y^i z^j &= \sum_{i=0}^{\infty} d_{i,0} i\mu y^i + \sum_{i=0}^{\infty}\sum_{j=1}^{\infty} d_{i,j} i\mu y^i z^j \\
&= \mu y \frac{\partial}{\partial y}\sum_{i=0}^{\infty} d_{i,0} y^i + \mu y \frac{\partial}{\partial y}\sum_{i=0}^{\infty}\sum_{j=1}^{\infty} d_{i,j} y^i z^j \\
&= \mu y \frac{\partial}{\partial y} P(y,z)
\end{aligned}$$

The sum of all the terms with $j\mu$ on the left hand side of Equation (3.109) through (3.112) gives

$$\begin{aligned}
\sum_{i=1}^{\infty}\sum_{j=1}^{\infty} d_{i,j} j\mu y^i z^j + \sum_{j=1}^{\infty} d_{0,j} i\mu z^j &= \sum_{i=1}^{\infty}\sum_{j=0}^{\infty} d_{i,j} j\mu y^i z^j + \sum_{j=0}^{\infty} d_{0,j} j\mu z^j \\
&= \mu z \frac{\partial}{\partial z}\sum_{i=1}^{\infty}\sum_{j=0}^{\infty} d_{i,j} y^i z^j + \mu z \frac{\partial}{\partial z}\sum_{j=0}^{\infty} d_{0,j} z^j \\
&= \mu z \frac{\partial}{\partial z} P(y,z)
\end{aligned}$$

The sum of all the terms with λ on the right hand side of Equation (3.109) through (3.112) gives

$$\sum_{i=1}^{\infty} d_{i-1,0}\lambda y^i + \sum_{i=1}^{\infty}\sum_{j=1}^{\infty} p_{i-1,j}\lambda y^i z^j = \lambda y P(y,z)$$

The sum of all the terms with μ on the right hand side of Equation (3.109) through (3.112) gives

$$\begin{aligned}
\sum_{i=1}^{\infty} d_{i-1,0}\mu y^i &+ \sum_{i=1}^{\infty}\sum_{j=1}^{\infty} p_{i-1,j+1}(j+1)\mu y^i z^j \\
&= y\sum_{i=0}^{\infty} d_{i,1}\mu y^i + y\sum_{i=0}^{\infty}\sum_{j=2}^{\infty} p_{i,j} j\mu y^i z^{j-1} \\
&= \mu y \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} p_{i,j} j y^i z^{j-1} \\
&= \mu y \frac{\partial}{\partial z} P(y,z)
\end{aligned}$$

The sum of all the terms with μ on the right hand side of Equation (3.109) through (3.112) gives

$$\begin{aligned}
d_{1,0}\mu + \sum_{j=1}^{\infty}\sum_{k=0}^{j} p_{k+1,j-k}(k+1)\mu z^j &= \sum_{j=0}^{\infty}\sum_{k=0}^{j} d_{k+1,j-k}(k+1)\mu z^j \\
&= \sum_{k=0}^{\infty}\sum_{j=k}^{\infty} d_{k+1,j-k}(k+1)\mu z^{j-k} z^k \\
&= \mu\sum_{k=0}^{\infty}(k+1)z^k \sum_{j=0}^{\infty} d_{k+1,j} z^j \\
&= \mu\sum_{k=0}^{\infty}\sum_{j=0}^{\infty} d_{k,j} k z^{k-1} z^j \\
&= \mu\frac{\partial}{\partial y}P(y,z)\,|_{y=z}
\end{aligned}$$

We now sum Equation (3.109) through (3.112) and write

$$\begin{aligned}
\lambda P(y,z) &+ \mu y\frac{\partial}{\partial y}P(y,z) + \mu z\frac{\partial}{\partial z}P(y,z) \\
&= \lambda y P(y,z) + \mu y\frac{\partial}{\partial z}P(y,z) + \mu\frac{\partial}{\partial y}P(y,z)\,|_{y=z}
\end{aligned}$$

which is equivalent to

$$\mu y\frac{\partial}{\partial y}P(y,z) - \mu\frac{\partial}{\partial y}P(y,z)\,|_{y=z} + \mu(z-y)\frac{\partial}{\partial z}P(y,z) = \lambda(y-1)P(y,z) \quad (3.113)$$

From Equation (3.109) and (3.113) we derive the following.

$$\begin{aligned}
\sum_{j=0}^{\infty} d_{0,j}(\lambda + j\mu)z^j &= \lambda\sum_{j=0}^{\infty} d_{0,j}z^j + \sum_{j=0}^{\infty} d_{0,j} j\mu z^j \\
\lambda P(0,z) + \mu z\frac{\partial}{\partial z}P(0,z) &= \mu\frac{\partial}{\partial y}P(y,z)\,|_{y=z} \qquad (3.114)
\end{aligned}$$

After plugging $y = z$ into Equation (3.114) we have

$$\mu z\frac{\partial}{\partial y}P(y,z)\,|_{y=z} - \mu\frac{\partial}{\partial y}P(y,z)\,|_{y=z} = \lambda(z-1)P(z,z)$$

$$
\begin{aligned}
\mu(z-1)\frac{\partial}{\partial y}P(y,z)\,|_{y=z} &= \lambda(z-1)P(z,z) \\
\mu\frac{\partial}{\partial y}P(y,z)\,|_{y=z} &= \lambda P(z,z) \\
\mu\frac{\partial}{\partial y}P(y,z)\,|_{y=z} &= \lambda P(y,z)\,|_{y=z}
\end{aligned} \tag{3.115}
$$

From Equation (3.115) the following holds only for $y = z$.

$$
\begin{aligned}
\mu\frac{\partial}{\partial y}P(y,z) &= \lambda P(y,z) \\
\frac{dP(y,z)}{P(y,z)} &= \frac{\lambda}{\mu}dy \\
lnP(y,z) &= \frac{\lambda}{\mu}y + C \\
P(y,z) &= Ke^{\frac{\lambda}{\mu}y}
\end{aligned} \tag{3.116}
$$

For $y = z$, we have

$$P(z,z) = Ke^{\frac{\lambda}{\mu}z}$$

Because $P(1,1) = 1$, we have that $K = e^{-\frac{\lambda}{\mu}}$, and so

$$P(z,z) = e^{\frac{\lambda}{\mu}(z-1)} \tag{3.117}$$

If we use Equation (3.117) in Equation (3.116), we get

$$\mu\frac{\partial}{\partial y}P(y,z)\,|_{y=z} = \lambda e^{\rho(z-1)} \tag{3.118}$$

where $\rho = \lambda/\mu$. We can now use (3.118) in (3.115) as follows.

$$\mu y\frac{\partial}{\partial y}P(y,z) + \mu(z-y)\frac{\partial}{\partial z}P(y,z) = \lambda(y-1)P(y,z) + \lambda e^{\rho(z-1)}$$

which gives us the main equation:

$$y\frac{\partial}{\partial y}P(y,z) + (z-y)\frac{\partial}{\partial z}P(y,z) = \rho(y-1)P(y,z) + \rho e^{\rho(z-1)} \tag{3.119}$$

Solving partial differential equation (3.119) requires the solution of an integral of the form $\int a^x x^{bx+c} dx$, for which no closed form solution has been found. From Equation (3.119) we obtain the following two results. After substituting $y = z = 1$ we get

$$\frac{\partial}{\partial y} P(y, z) \mid_{y=z=1} = \overline{N_A} = \rho \tag{3.120}$$

where $\overline{N_A}$ is the average number of alive customers in the system. We can find $\overline{N_A}$ using simpler arguments as follows. Refer to Figure 3.79. Let p_k represent the probability of k alive customers in the system, i.e., the probability of k customers being in the "ALIVE" box. When there are k customers in the "ALIVE" box, then the rate of leaving that box is $k\mu$. Upon every departure exactly one customer leaves the system. Arrival rate of customers into the system is λ. We can now write the following relation

$$\lambda = \sum_{k=1}^{\infty} 1 k \mu p_k = \mu \overline{N_A} \tag{3.121}$$

which is equivalent to Equation (3.120).

For $y = z = 0$, Equation (3.119) gives

$$P(0,0) = d_{0,0} = e^{-\rho} \tag{3.122}$$

3.4.2 System A2: M/M(BR) is a Perfect System

Figure 3.16 shows that the M/M(BR) system gives "perfect" performance. Consider that an arrival finds k customers in the system. Because the system is a broadcast system, the system will always do useful work on exactly one customer while there are customers in the system. Thus the length of a busy period equals that of a regular M/M/1 system where service times are distributed as successful service times X_s in M/M(BR) system. Since the average system time in an M/M/1 system does not depend on the scheduling discipline, the average system time is always equal to that of a FCFS discipline. Thus we can find the average system time in M/M(BR) as

$$T = P[\tilde{N} = 0]\overline{X} + \sum_{k=1}^{\infty} P[\tilde{N} = k]k\overline{X_{s|k}}$$

where $\tilde{N}$ is the number of customers in the system, and $X_{s|k}$ is the length of a successful service given k customers in the system. If we define $X_{min|k}$ as minimum of k service times, then we can write

$$\begin{aligned}
\overline{X_{s|k}} &= -\int_0^{\infty} x dP[X_{min|k} > x] \\
&= -\int_0^{\infty} x d\{P[X > x]^k\} \\
&= -\int_0^{\infty} x d(e^{-k\mu x}) \\
&= -\int_0^{\infty} x dP[X > kx] \\
&= \frac{1/\mu}{k} = \frac{\overline{x}}{k} \\
T &= P[\tilde{N} = 0]\overline{x} + \sum_{k=1}^{\infty} P[\tilde{N} = k]k\frac{\overline{x}}{k} \\
&= \overline{x}
\end{aligned}$$

Thus, for M/M(BR) system we have the normalized power as

$$P = \rho \tag{3.123}$$

3.4.3 System A3: M/D(B) is an M/D/1 Queue

For the M/D(B) system we get the same normalized power as for an ordinary M/D/1 queue. In an M/D(BR), service time is never wasted (i.e., the service for exactly one customer is always useful, given at least one customer in the system). As soon as a customer leaves, the customers left in the system will restart. It is obvious that the system behaves as an M/D/1 queue. Thus we should get

$$P = 2\rho(1-\rho)/(2-\rho) \tag{3.124}$$

for $q = 1$, and this is plotted as dashed curve in Figure 3.16.

3.4.4 System A4: M/D$_q$M(L) is an M/D$_q$M/1 Queue

A full-conflict locking ISR system G/G(L) is equivalent to a corresponding regular queue G/G/1. The only difference is that in the G/G(L) systems, customers wait for the common resource, while in the G/G/1 they wait for the common server to be assigned to them. Once the resource in the G/G(L) system is assigned to a customer, the rate of service is the same as the rate of service in the corresponding G/G/1 queue. While we find the power for a wide range of arrival/service time distributions, we only use it here for Poisson arrivals and D$_q$M distributed service times.

The known expression for an M/G/1 queue gives us [12]

$$N = \rho + \frac{\lambda^2 \overline{x^2}}{2(1-\rho)} \tag{3.125}$$

The second moment for the D_qM distribution is

$$\overline{x^2} = (\overline{x})^2(2p^2+2pq+q^2)$$

and thus

$$\begin{aligned} N &= \rho+\rho^2\frac{2-2q+q^2}{2(1-\rho)} \\ T_n &= 1+\rho\,\frac{2-2q+q^2}{2(1-\rho)} \qquad (3.126) \\ P &= \frac{2\rho(1-\rho)}{2-2q\rho+q^2\rho} \qquad (3.127) \end{aligned}$$

Figures 3.82 and 3.83 show the normalized power and the normalized response time, respectively, for the $M/D_qM(L)$ system.

3.4.5 Summary

In Sections 3.4.1 through 3.4.4 we have analyzed the full-conflict systems M/M(SR), M/M(BR), M/D(B), and $M/D_qM(L)$. Figure 3.84 shows graphicaly the systems analyzed in Section 3.4.

3.5 Departure Processes in Winner Queues

Consider again the joint probabilities of the transitions between the states and the interdeparture time density, $p_{i,j}(v)$ and $p_{i,k,j,l}(v)$, defined for the simple winner queues and the winner queues with partial restarts, respectively. The above probabilities allow us to take a closer look at the departure processes. The distribution of the random variable V (the length of the interdeparture time) we

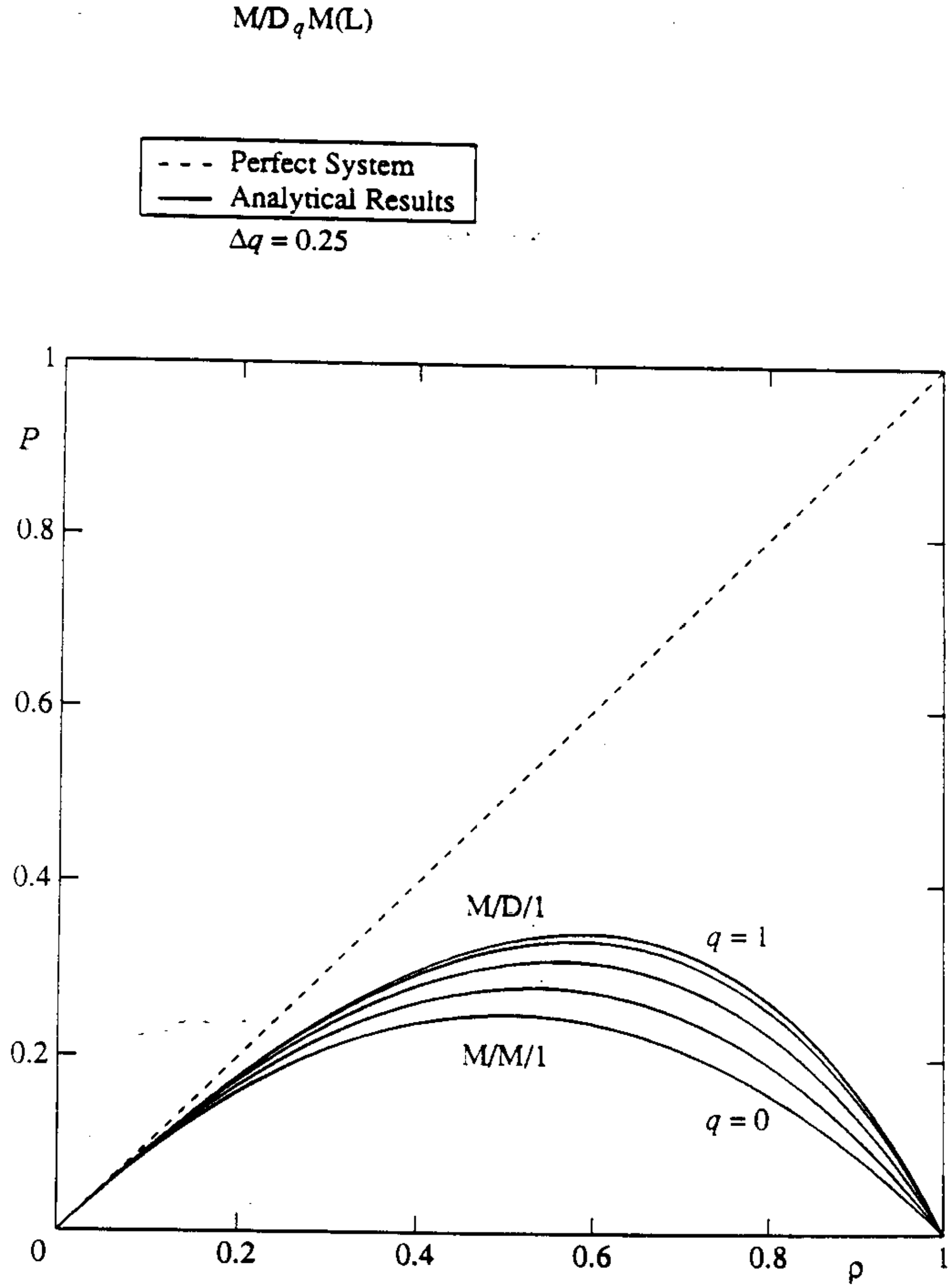


Figure 3.82: Normalized Power for M/D$_q$M(L)

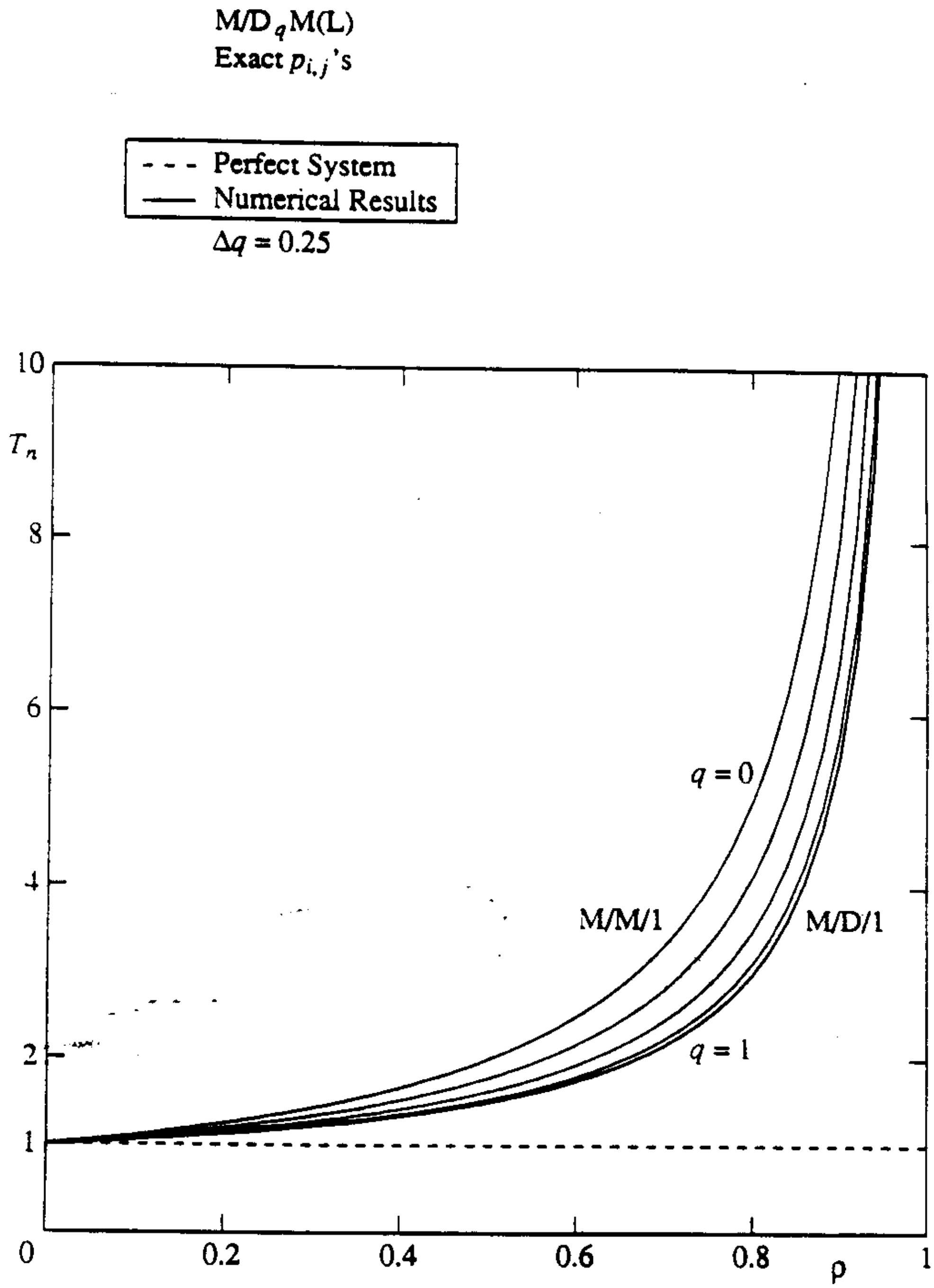


Figure 3.83: Normalized Response Time for $M/D_qM(L)$

		M	D_qM	D
SR,r	M	(†)		
	$D_{q,r}M$			
	D			
SN,r				
sBR,r	M			
	$D_{q,r}M$			
	D			
sBN,r				
BR,r	M	(E†)		
	$D_{q,r}M$			
	D			(E†)
BN,r				(E†)
L		(E)	(E)	(E)

System	Section	Equation	Figure
A1	3.4.1	(3.119)	
A2	3.4.2	(3.123)	3.16 thru 3.19
A3	3.4.3	(3.124)	3.16 thru 3.19
A4	3.4.4	(3.127)	3.82 thru 3.83

□ numerical results

○ analytic results

E exact transition probabilities

A approximate transition probabilities

† only for $r = 1$

Figure 3.84: Overview Table of Results for "Other Approaches"

get as follows.

$$\begin{aligned} \nu(v) &\stackrel{\text{def}}{=} \frac{1}{dv} P[v < V < v+dv] \\ &= \sum_{i=0}^{\infty} d_i \sum_{j=0}^{\infty} p_{i,j}(v) dv \end{aligned} \tag{3.128}$$

Let us look at the M/M(SR) system once again. The distribution of the random variable V we get as follows. After including the value for $p_{i,j}(v)$ from Equation 3.38 in Equation 3.128, we get the following.

$$\begin{aligned} \nu(v) &= d_0 \lambda e^{-\lambda v}(1-e^{-\mu v}) e^{\rho(1-e^{\mu v})} \\ & \sum_{i=1}^{\infty} d_i \left[\lambda(1+\mu v)(1-e^{-\mu v}) + \mu^2 i v\right] \\ & (1+\mu v)^{i-1} e^{-(\lambda+i\mu)v} e^{\rho(1-e^{-\mu v})} \end{aligned} \tag{3.129}$$

The probability density function 3.129 differs quite a lot from an exponential. This shows that Morris and Wong in [19,20] and Ryu and Thomasian in [25,34] were making a big approximation by assuming the transaction commit process to be Poisson, even though the systems they analyzed were closed systems with the number of transactions in the system kept fixed.

We can write yet another relation among the transition and state probabilities as follows.

$$\frac{1}{\lambda} = \overline{V} = \sum_{i=0}^{\infty} d_i \sum_{j=0}^{\infty} v p_{i,j}(v) dv \tag{3.130}$$

As shall be pointed out later, complicated expressions for transition probabilities in some systems result in simple expressions for the average number of customers in the system. Equation 3.130 is another equation with a simple value on one side, and an expression including complex transition probabilities on the other.

3.6 Conclusion

In order to be able to study the performance of optimistic concurrency control schemes in databases, we have studied special types of queues, winner queues and winner queues with partial restarts.

In these queues we have investigated the average response time of customers. The results obtained by simulation and analysis were shown in terms of the normalized average system time and the normalized power defined as utilization factor divided by normalized average response time of customers.

We first showed simulation results for four different classes of winner queues: silent-redraw (SR), silent-noredraw (SN), broadcast-redraw (BR), and broadcast-noredraw (BN). The results showed the power curves with varying q from 0 to 1. Later, we show simulation results for six different classes of winner queues. silent-redraw (SR), silent-noredraw (SN), silent/broadcast-redraw (sBR), silent/broadcast-noredraw (sBN), broadcast-redraw (BR), and broadcast-noredraw (BN).

We were able to calculate the normalized power for the simple winner queues M/M(SR) and M/D_qM(BR) and for the winner queues with partial restarts M/M(SR,r), M/M-D_{qr}M(sBR,r), and M/D_qM-D_{qr}M(sBR,r). We also did an approximation for the simple winner queue M/D(S) and for the winner queues with partial restarts M/M-D(SR,r), M/D-M(SR,r), M/D(S,r), and M/D-D_{qr}M-(sBR,r).

	Further Studies of Winner Queues
1	Research on noredraw winner queues.
2	Closed formulas.

Table 3.13: Further Research in Winner Queues

Using other approaches we also analyzed simple winner queues M/M(SR), M/M(BR), and M/D(B), and the full-conflict system with locking $M/D_qM(L)$. Figure 3.85 shows graphically the systems analyzed in Chapter 3.

Further studies of winner queues, as shown in Table 3.13, would include research on noredraw winner queues and closed formulas for all the queues, starting from the formula for normalized power for M/D(BR) queue $P = 2\rho(1-\rho)/(2-\rho)$.

		M	D_qM	D
SR,r	M	E		A
	$D_{q,r}$M			
	D	A		A
SN,r				A
sBR,r	M	E		A
	$D_{q,r}$M	E		A
	D	E		A
sBN,r				A
BR,r	M	(E†)	E	E
	$D_{q,r}$M	E	E	E
	D	E	E	(E†)
BN,r				(E†)
L		(E)	(E)	(E)

System	Section	Equation	Figure
S1	3.2.4	(3.39)	3.11 thru 3.13
S2	3.2.5	(3.41)	3.14 thru 3.15
S3	3.2.6	(3.43)	3.16 thru 3.19
P1	3.3.4	(3.84)	3.40 thru 3.41
P2	3.3.5	(3.87)	3.42 thru 3.43
P3	3.3.6	(3.90)	3.44 thru 3.45
P4	3.3.7	(3.94)	3.46 thru 3.47
P5	3.3.8	(3.97)	3.48 thru 3.53
P6	3.3.9	(3.100)	3.54 thru 3.59
P7	3.3.10	(3.103)	3.60 thru 3.77
A1	3.4.1	(3.119)	
A2	3.4.2	(3.123)	3.16 thru 3.19
A3	3.4.3	(3.124)	3.16 thru 3.19
A4	3.4.4	(3.127)	3.82 thru 3.83

□ numerical results

○ analytic results

E exact transition probabilities

A approximate transition probabilities

† analytic for $r = 1$, numerical for all r

Figure 3.85: Overview Table of Results for Full Conflict Systems

CHAPTER 4

Partial Conflict Systems

4.1 Introduction

Partial conflict systems are those ISR systems in which any two concurrent customers may or may not conflict. Database systems in which the resources are database granules are, in general, partial conflict ISR systems. In partial conflict systems we are also interested in performance in terms of the normalized response time and power.

Our approach is to map the results already obtained for the full conflict systems into the partial conflict case. The mapping is shown in Figure 4.1. For some partial conflict systems the mapping is exact, while for others the mapping is an approximation.

In Section 4.2 we describe the conflict measure used in the mapping. Section 4.3 gives cases in which the mapping is exact, while Section 4.4 give cases in which the mapping is approximate. The error in mapping is described in Section 4.5. The graphs showing the mapped results for different schemes are given in Chapter 5, where we apply full-to-partial conflict mapping to database systems. It is those data which are used to show the error of mapping in Section 4.5.

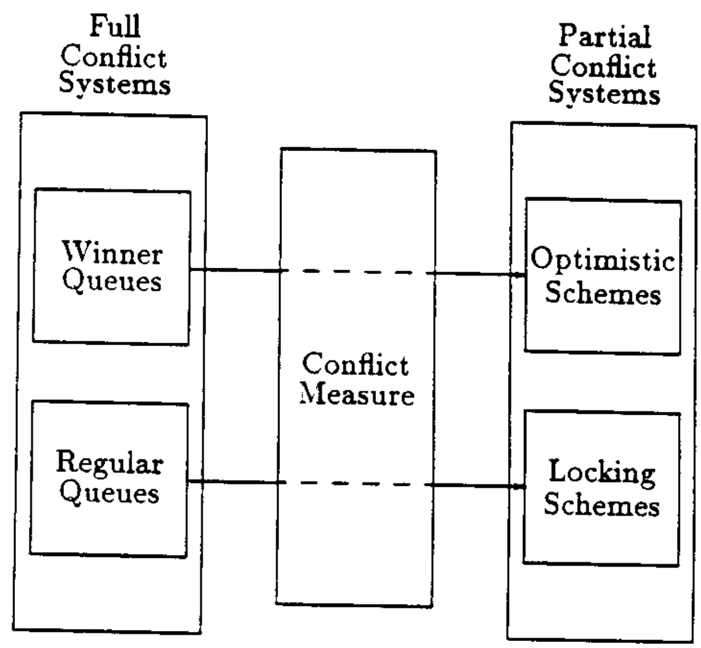


Figure 4.1: Full-to-Partial Conflict Mapping

Finally, the conclusion on partial conflict systems is given in Section 4.6.

4.2 Conflict Measure

We define the *conflict measure* c to be a measure of interference among concurrent customers due to conflicting resource demands. Such a measure has value 0 for no conflict systems, and value 1 for full conflict systems. Figure 4.2 shows all the systems with shared resources and associated conflict measure. Note that single conflict systems can be full conflict systems (when there is one system resource to which access is demanded by every customer).

For partial conflict systems, the conflict measure varies between 0 and 1, $c \in (0, 1)$. In this work we define the conflict measure c to be the probability that the demand sets of two customers overlap. Thus, if S_1 and S_2 are demand sets of two arbitrary customers, then

$$c = P[S_1 \cap S_2 \neq \emptyset]$$

Assuming that demand set size of every customer is fixed at $\bar{s} = s$, the following two sections describe the two data access patterns used herein, *random* and *sequential.*

4.2.1 Random Resource Demands

Every customer demands exactly s resources. If the demand for any subset of size s of the total set of resources T is equally likely, we call this *random resource*

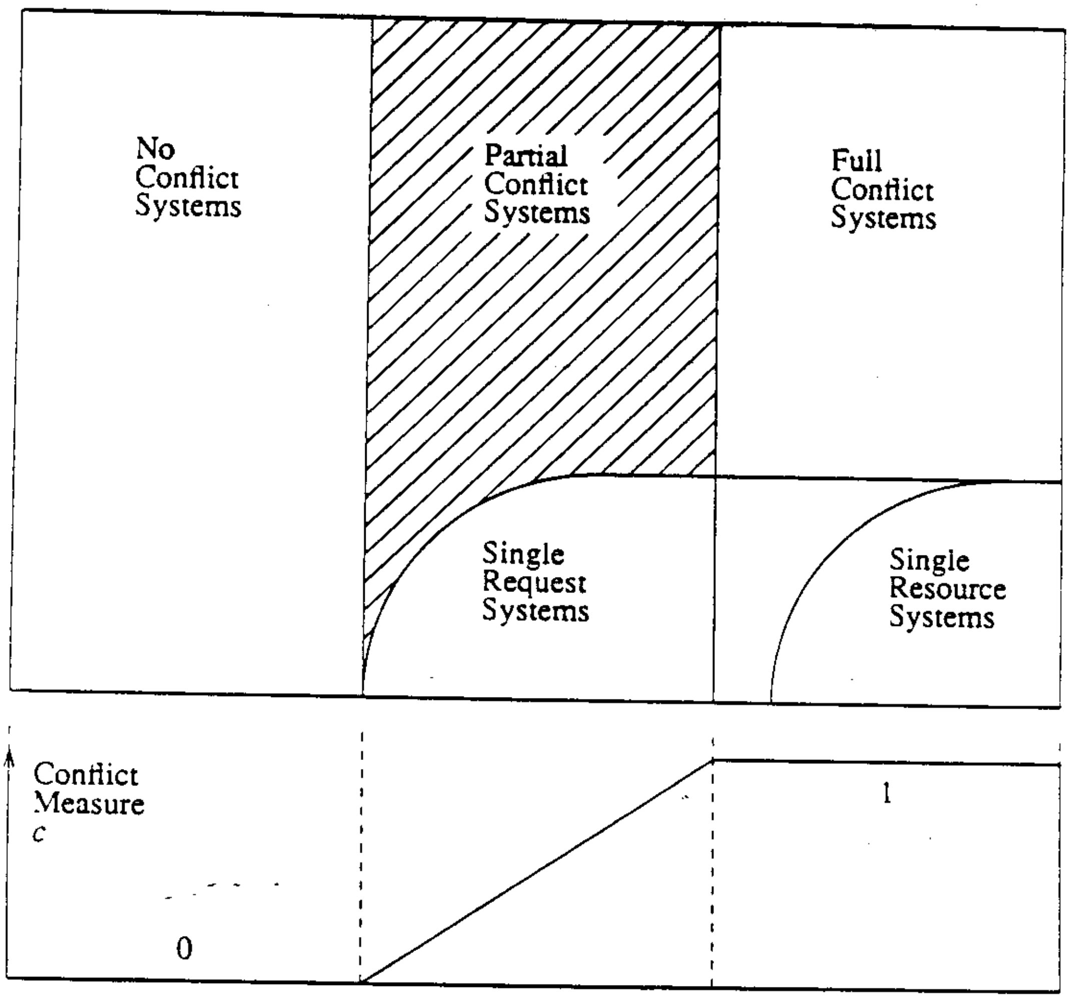


Figure 4.2: Conflict Measure in ISR Systems

demands. The conflict measure, as defined above, is given as

$$c = \begin{cases} 1 - \frac{\binom{t-s}{s}}{\binom{t}{s}}, & s \le t/2 \\ 1, & s > t/2 \end{cases} \tag{4.1}$$

where t is the size of T.

4.2.2 Sequential Resource Demands

Let us assume that the total set of resources is an ordered set in a wrap-around fashion (the last resource is followed by the first one). Then, if every demand set is a sequence of s adjacent resources, we call this *sequential resource demands*. We will also assume in our sequential case that the first resource in a demand set is chosen equally likely from the system pool of resources. The conflict measure, as defined above, is given as

$$c = \begin{cases} \frac{(2s-1)}{t}, & s \le t/2 \\ 1, & s > t/2 \end{cases} \tag{4.2}$$

4.3 Exact Mapping

For *single request systems*, the mapping gives us exact results. We use the mapping as follows. Let $T_{nc}(\rho)$ and $P_c(\rho)$ represent normalized response time and power, respectively, as functions of ρ in a system with conflict measure c. From the definition of the conflict measure, it follows that for the full conflict systems $c = 1$, and so their response time and power we denote as $T_{n1}(\rho)$ and

$P_1(\rho)$, respectively. The mapping we propose is as follows.

$$T_{nc}(\rho) = T_{n1}(c\rho) \tag{4.3}$$

$$P_c(\rho) = \frac{1}{c}P_1(c\rho) \tag{4.4}$$

Except for the special cases of "better-than-perfect" systems (such as M/M-(BR,0)) it is clear for silent and silent/broadcast systems that $P_1(c\rho) \leq c\rho$ and so $P_c(\rho) = P_1(c\rho)/c \leq c\rho/c = \rho$; thus the power $P_c(\rho)$ is properly less than ρ (the ideal case).

To show that Equations 4.3 and 4.4 give exact results for the single request systems, refer to Figure 4.3. Consider just one resource from the pool of t resources. Let us imagine an ISR system for that resource only, and we denote the symbols referring to that system by the subscript s. The average response time of the "small" ISR system equals the average response time of the original system. Since the arrival rate to all the resources are the same, and equal $\rho_s = \bar{x}\lambda_s = \bar{x}\lambda/t = \rho/t$, we have that

$$T_{nc}(\rho) = T_{nc,s}(\rho/t) = T_{n1}(\rho/t)$$

On the other hand, expressions for the conflict measure for both random and sequential resource demand patterns give us for the single request systems

$$c = 1/t$$

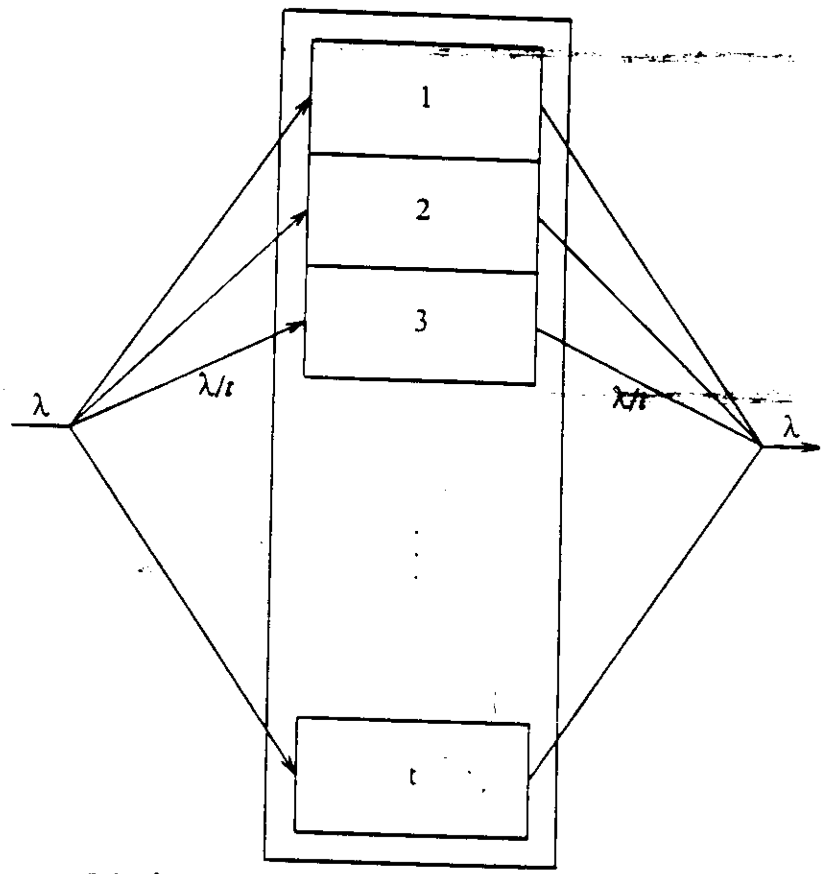


Figure 4.3: Mapping to Single Request Systems

and thus

$$T_{nc}(\rho) = T_{n1}(c\rho)$$

and also

$$\begin{aligned} P_c(\rho) &= \frac{\rho}{T_{nc}(\rho)} \\ &= \frac{1}{c} \cdot \frac{c\rho}{T_{n1}(c\rho)} \\ &= \frac{1}{c} P_1(c\rho) \end{aligned}$$

In the case of the single request ISR systems we have the exact mapping for more general resource demand patterns as well. Consider the IRS system where demands to the resources are non-uniformly distributed. We define t classes of customers as follows. All the customers that access resource $R_k \in T$, $k = 1, 2, \ldots, t$ belong to class k. The normalized response time and power for the customers of class k we will denote as $T_n^{(k)}(\rho)$ and $P^{(k)}$, respectively. Let r_k be the probability that a customer belongs to class k. The following holds.

$$T_{nc}^{(k)}(\rho) = T_{n1}(r_k\rho) \tag{4.5}$$

$$P_c^{(k)}(\rho) = \frac{1}{c} P_1(r_k\rho) \tag{4.6}$$

$$T_{nc}(\rho) = \sum_{k=1}^{t} r_k T_{nc}(r_k\rho) \tag{4.7}$$

$$P_c(\rho) = \frac{\rho}{T_{nc}(\rho)} \tag{4.8}$$

4.4 Approximate Mapping

We explain the mapping given by Equation 4.3 and 4.4 for the multiple (non-single) request systems as follows. Consider the system given in Figure 4.4. Suppose that a demand set $\tilde{S}$ is requested by customer C. Without loss of generality we assume in Figure 4.4 that the s resources are adjacent to each other. The fraction of the arriving customers whose demand sets overlap set $\tilde{S}$, assuming uniform resource demand distribution, is equal the conflict measure c, which follows from our definition in Section 4.2. Since the average response time of the customer C is affected by all transactions that conflict with C, we could define a full conflict ISR system around the set of resources $\tilde{S}$. The average response time of that "small" system equals the average response time of the original system. The arrival rate to the "small" system equals $c\rho$, and thus, we may write the mapping given in Equation 4.3 and 4.4.

The response time of the customer C also depends, indirectly, on the customers whose demand sets do not overlap $\tilde{S}$. Consider customer B whose demand set $\tilde{S_B}$ is adjacent to $\tilde{S}$. Let another customer, customer A demand a set that overlaps both $\tilde{S}$ and $\tilde{S_B}$. Let all three customers be concurrent. Then, the customer C is dependent on the customer A, and A is dependent on the customer B. Thus, C depends on B even though their demand sets do not overlap. This explains why the mapping above is only an approximation for the multiple request systems. Note that in the single request systems no two customers may be

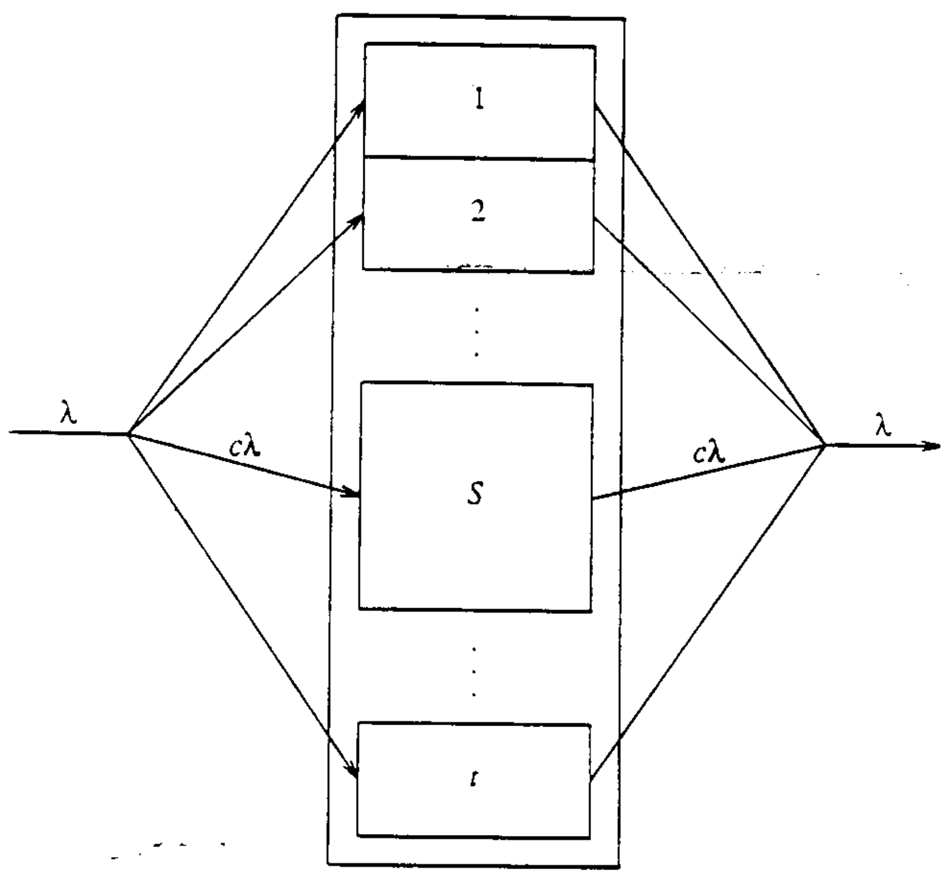


Figure 4.4: Mapping to Multiple Request Systems

indirectly dependent, and, thus, the mapping is exact for those systems. The following section calculates the quantitative error made in the mapping, specifying the domains in the system load, the service time distribution, and the conflict resolution scheme for which the mapping gives a close approximation.

4.5 Error in Mapping

In Figures 4.5 through 4.7 we show the error in mapping for silent, broadcast, and locking schemes, with random resource access for varying $0 < \rho \leq 1$. Figures 4.8 through 4.10 display the error in mapping for silent, broadcast, and locking, in the case of sequential access. In Figures 4.11 through 4.16 we show the same error plotted against the conflict measure c, $0 < c < 1$.

For both normalized power and response time, the errors represent the normalized difference between the results obtained from mapping and the results obtained through simulation. Let P_S and P_C represent the power obtained through simulation and numerical calculation, respectively. Analogously, let T_{nS} and T_{nC} represent the normalized service time obtained by simulation and numerical calculations, respectively. The error plotted in Figures 4.5 through 4.16 is defined as

$$Error \stackrel{\text{def}}{=} \frac{P_C - P_S}{P_S} = \frac{\frac{\rho}{T_{nC}} - \frac{\rho}{T_{nS}}}{\frac{\rho}{T_{nS}}} = \frac{T_{nS} - T_{nC}}{T_{nC}} \tag{4.9}$$

The results are obtained for both uniform random and uniform sequential resource demand patterns. The total number of resources was $t = 100$. For the

Random			Sequential		
s	t	c	s	t	c
2	100	0.040	5	100	0.090
4	100	0.153	10	100	0.190
6	100	0.317	15	100	0.290
8	100	0.500	20	100	0.390
10	100	0.670	25	100	0.490
12	100	0.804	30	100	0.590
14	100	0.897	35	100	0.690
16	100	0.953	40	100	0.790
18	100	0.981	45	100	0.890
20	100	0.993	50	100	0.990

Table 4.1: Conflict Measure Values

random case, the demand set size was varied from 2 to 20 with step $\Delta s = 2$. For the sequential case, the demand set size was varied from 5 to 50 with step $\Delta s = 5$. Table 4.1 shows the values for the conflict measure for those values of s.

Figures 4.5 through 4.10 show that in both random and sequential access the error is very small for the silent and broadcast schemes when the service time distribution is memoryless. Even half-deterministic service time distribution shows very small error for the broadcast case. Pure deterministic service times

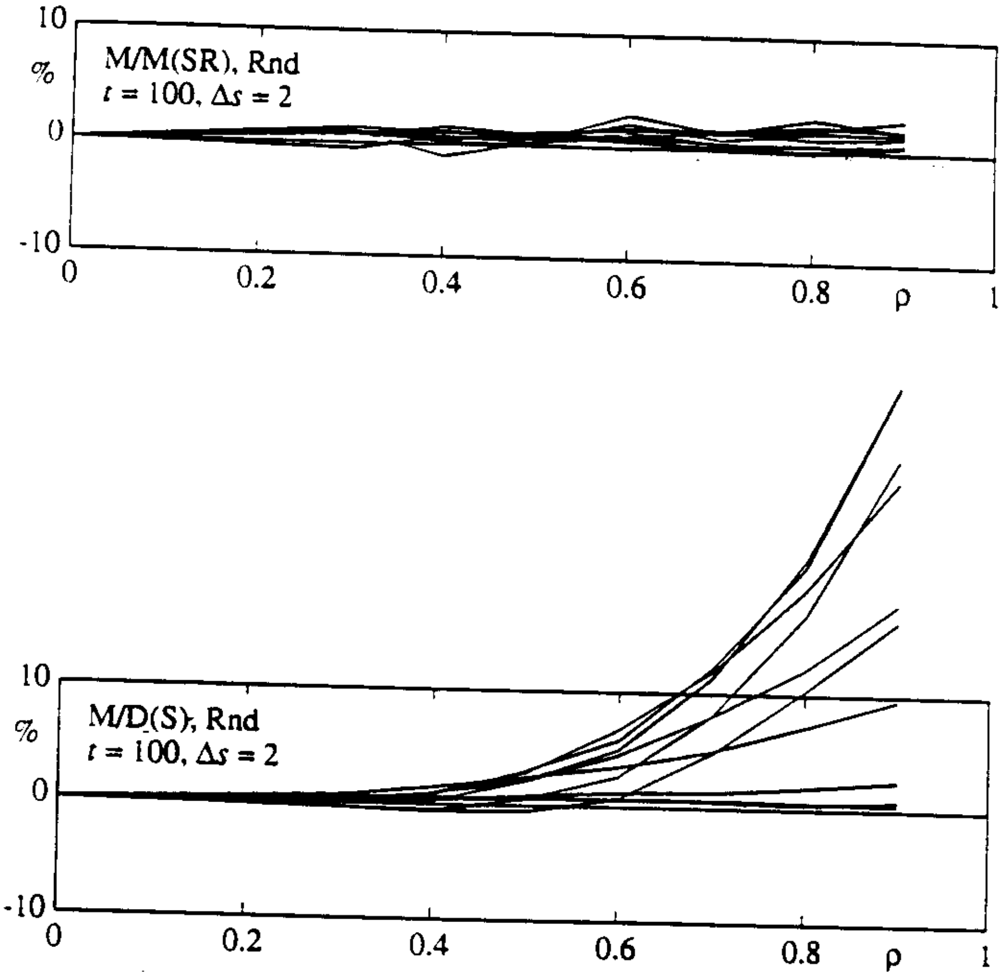


Figure 4.5: Error versus ρ for Silent/Random

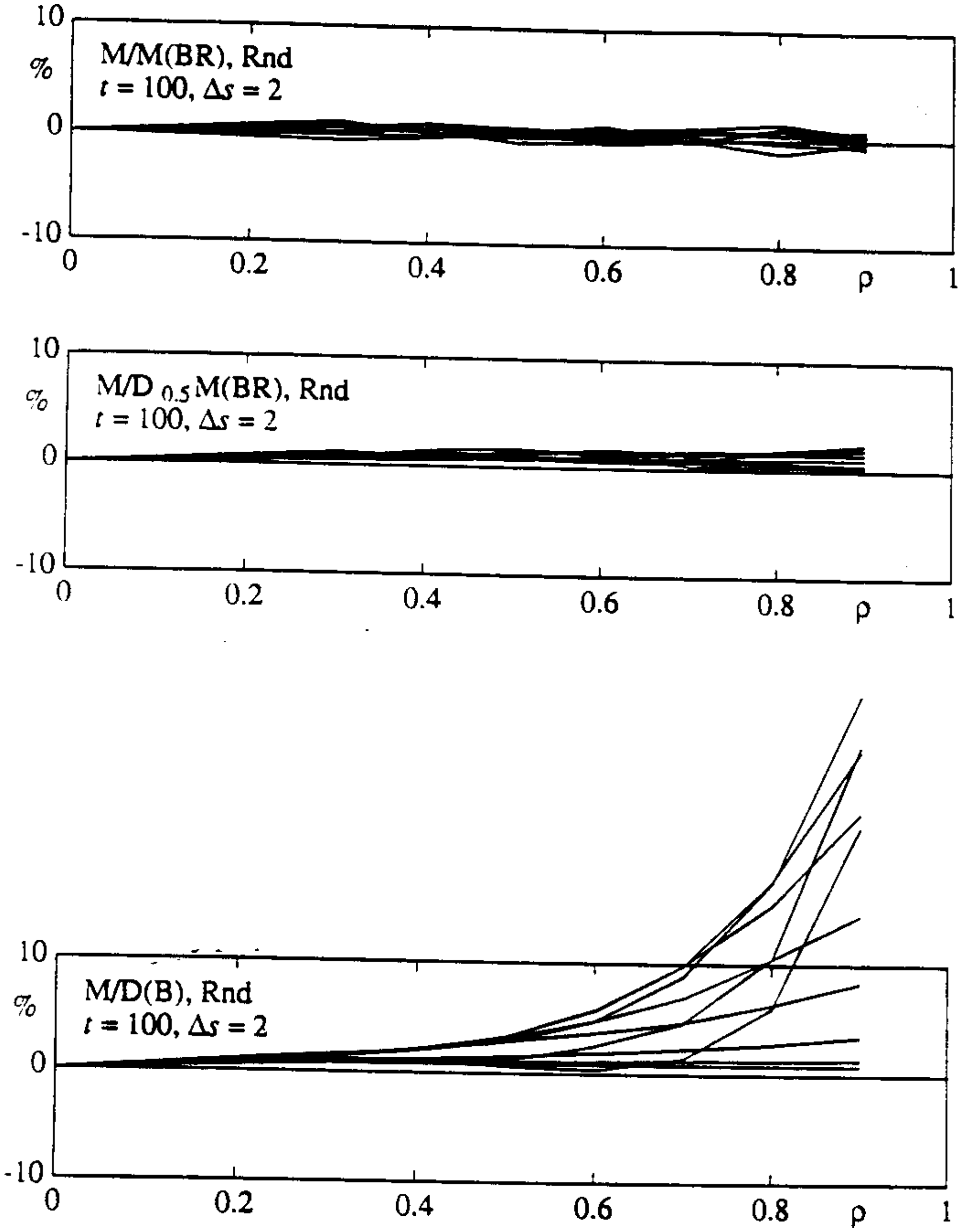


Figure 4.6: Error versus ρ for Broadcast/Random

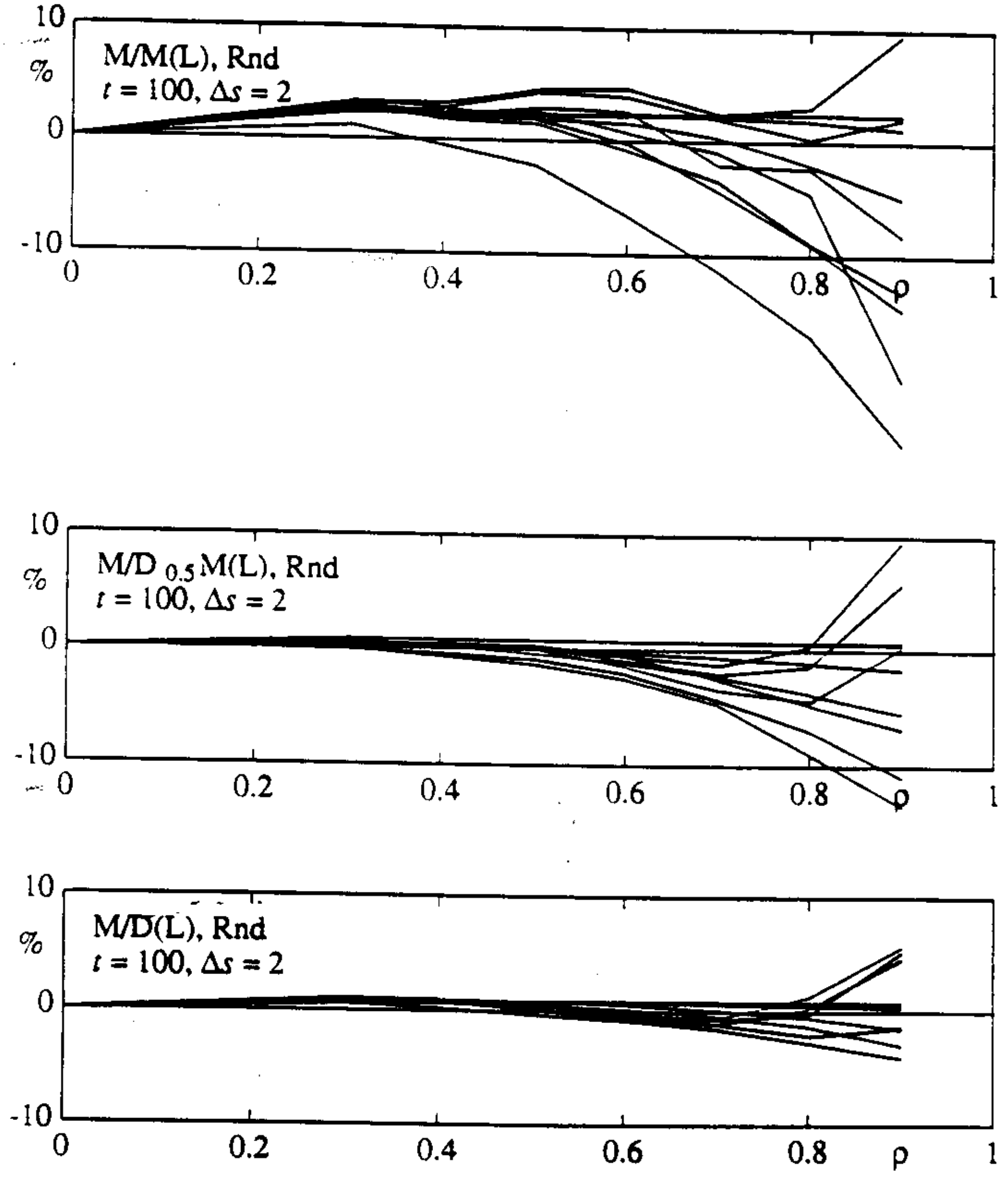


Figure 4.7: Error versus ρ for Locking/Random

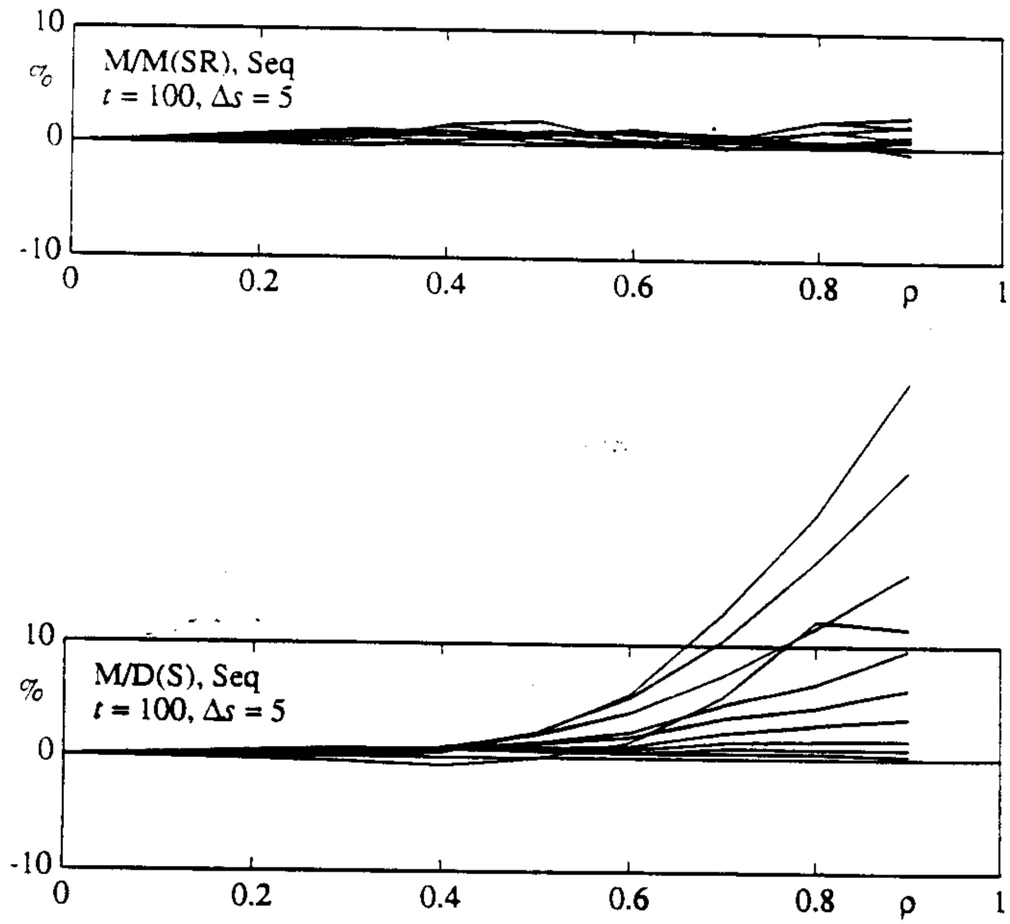


Figure 4.8: Error versus ρ for Silent/Sequential

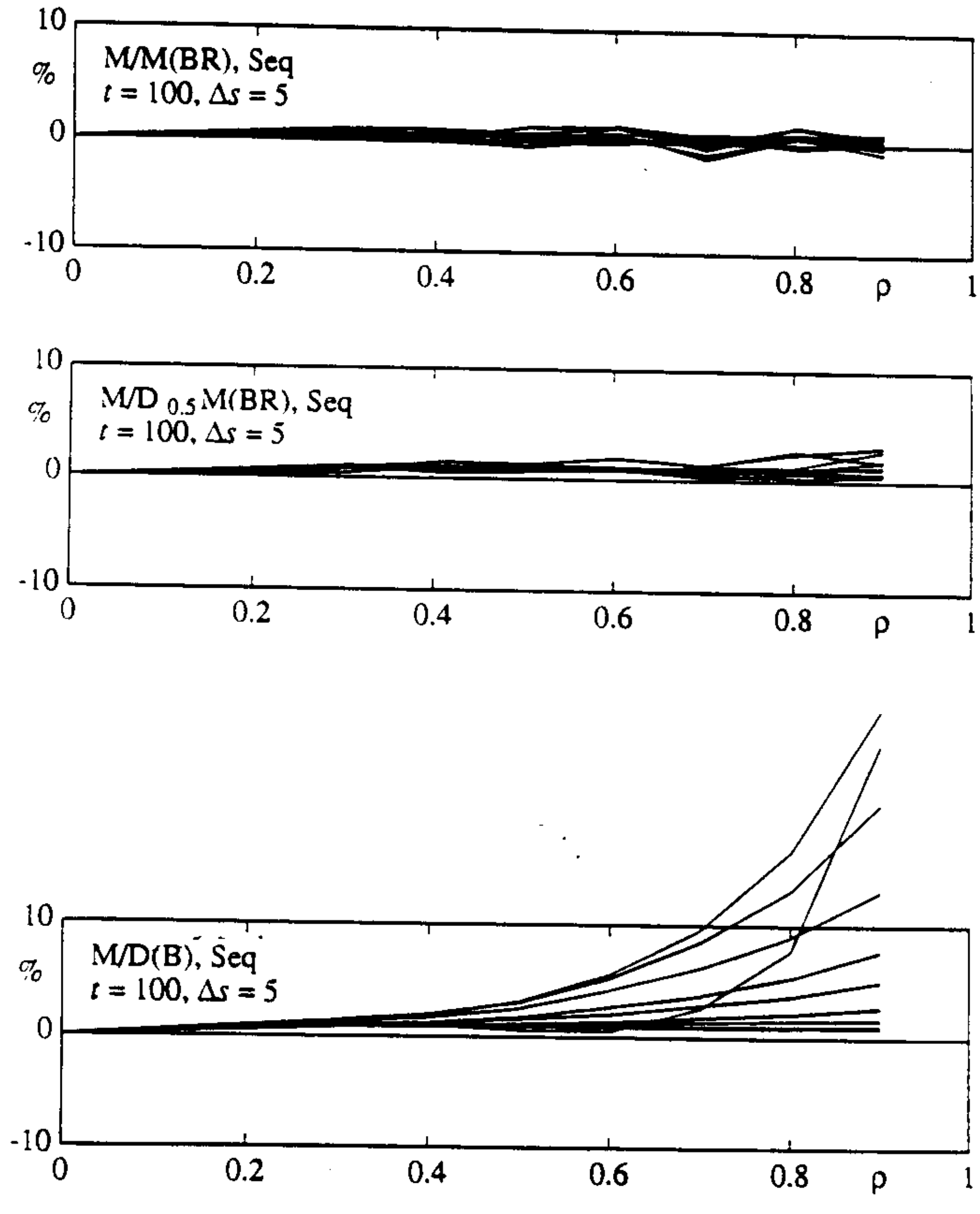


Figure 4.9: Error versus ρ for Broadcast/Sequential

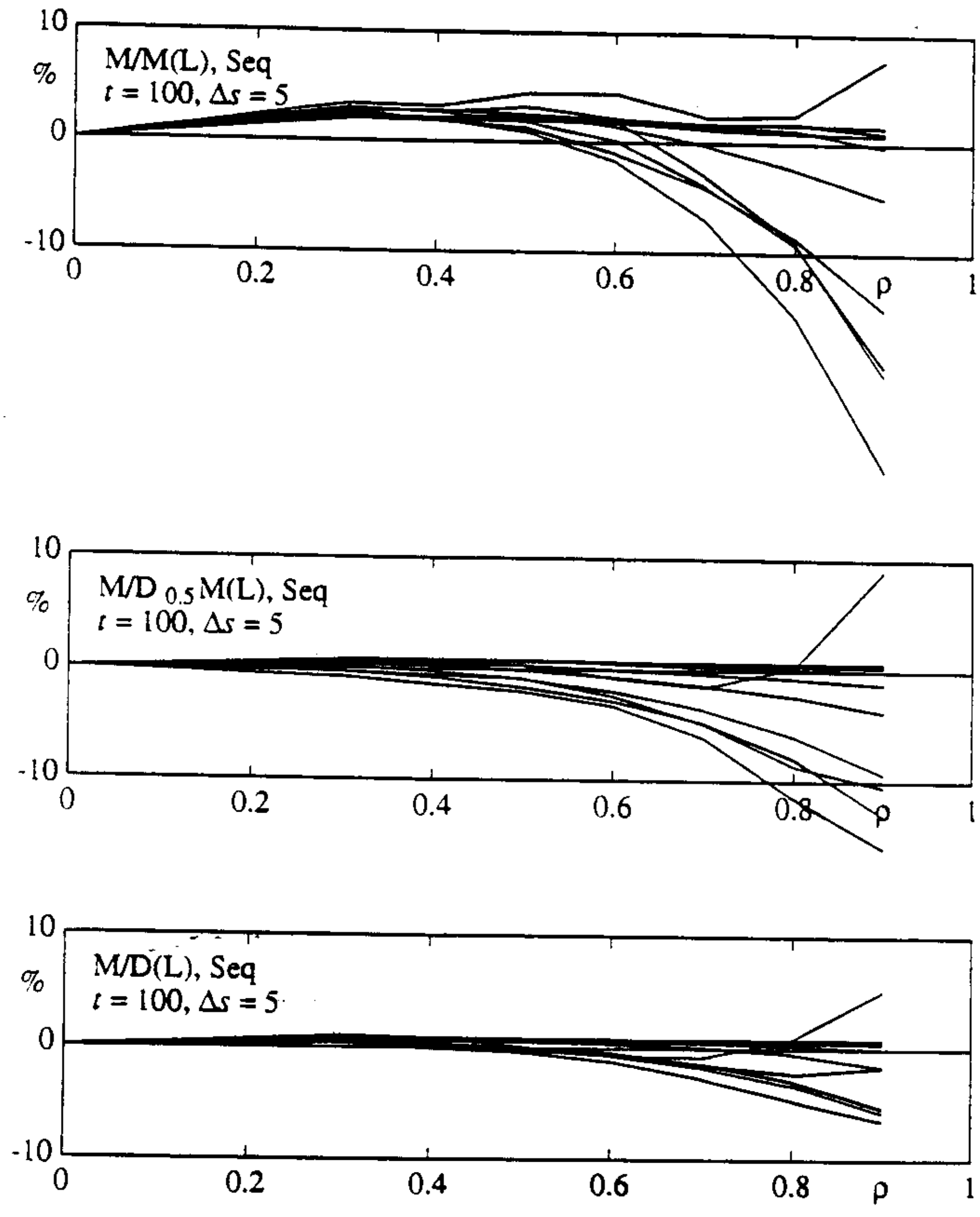


Figure 4.10: Error versus ρ for Locking/Sequential

show larger error (over 10silent and broadcast, but only for $\rho > 0.6$.

For the locking scheme the situation is reversed. Only deterministic service times give small error, while half-deterministic and memoryless give larger error for $\rho > 0.7$.

From Figures 4.11 through 4.16 we see that the error of mapping is small for small and high conflict measure c. The error is highest for $0.6 < c < 0.9$. For silent and broadcast with memoryless and half-deterministic service times the error is very small for the entire range of $0 \leq c \leq 1$. The same goes for locking with deterministic service times.

4.6 Conclusion

Performance results for the partial conflict systems are obtained by mapping from the results for the full conflict systems. The mapping is done using the probability of conflict between any two customers as the conflict measure. For the single request systems the mapping is exact, while for the other partial conflict systems the mapping is approximate.

Finding an exact mapping for all the partial conflict systems is complex. It is this mapping that is the core of the difficulty in modeling the well known and widely researched concurrency control schemes in databases. In this work, however, we explicitly pinpoint the problem of mapping. It may be possible to find the mapping in the simplified case of infinite servers. Since the problem is separated from queueing, we would be satisfied even with some close approximations.

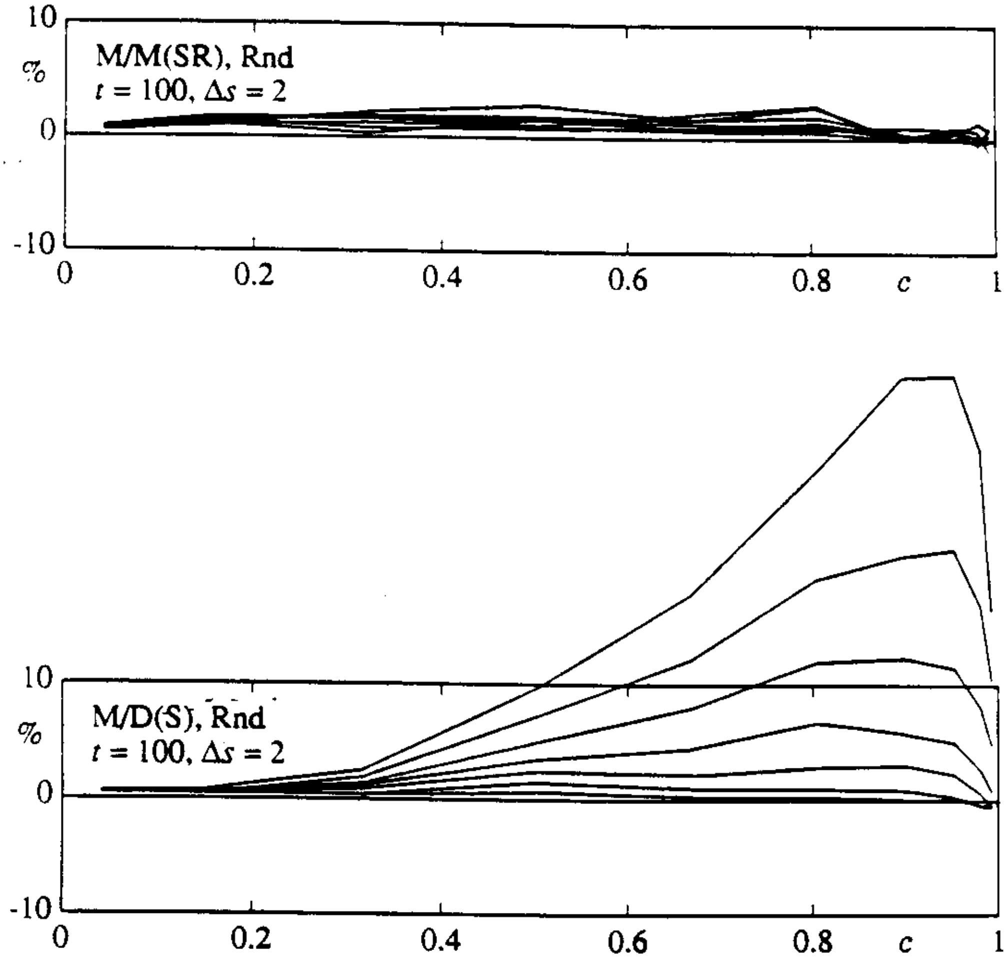


Figure 4.11: Error versus c for Silent/Random

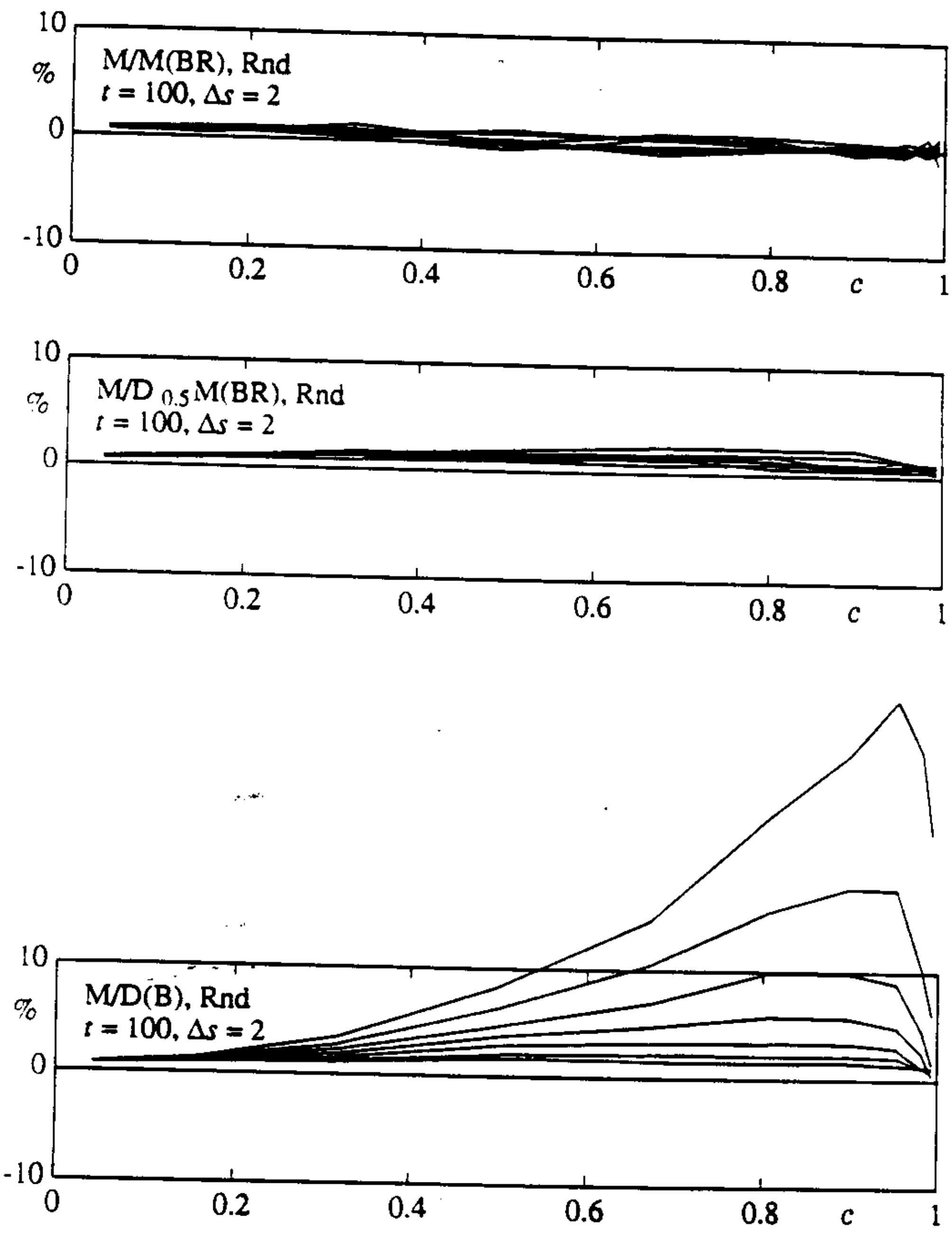


Figure 4.12: Error versus c for Broadcast/Random

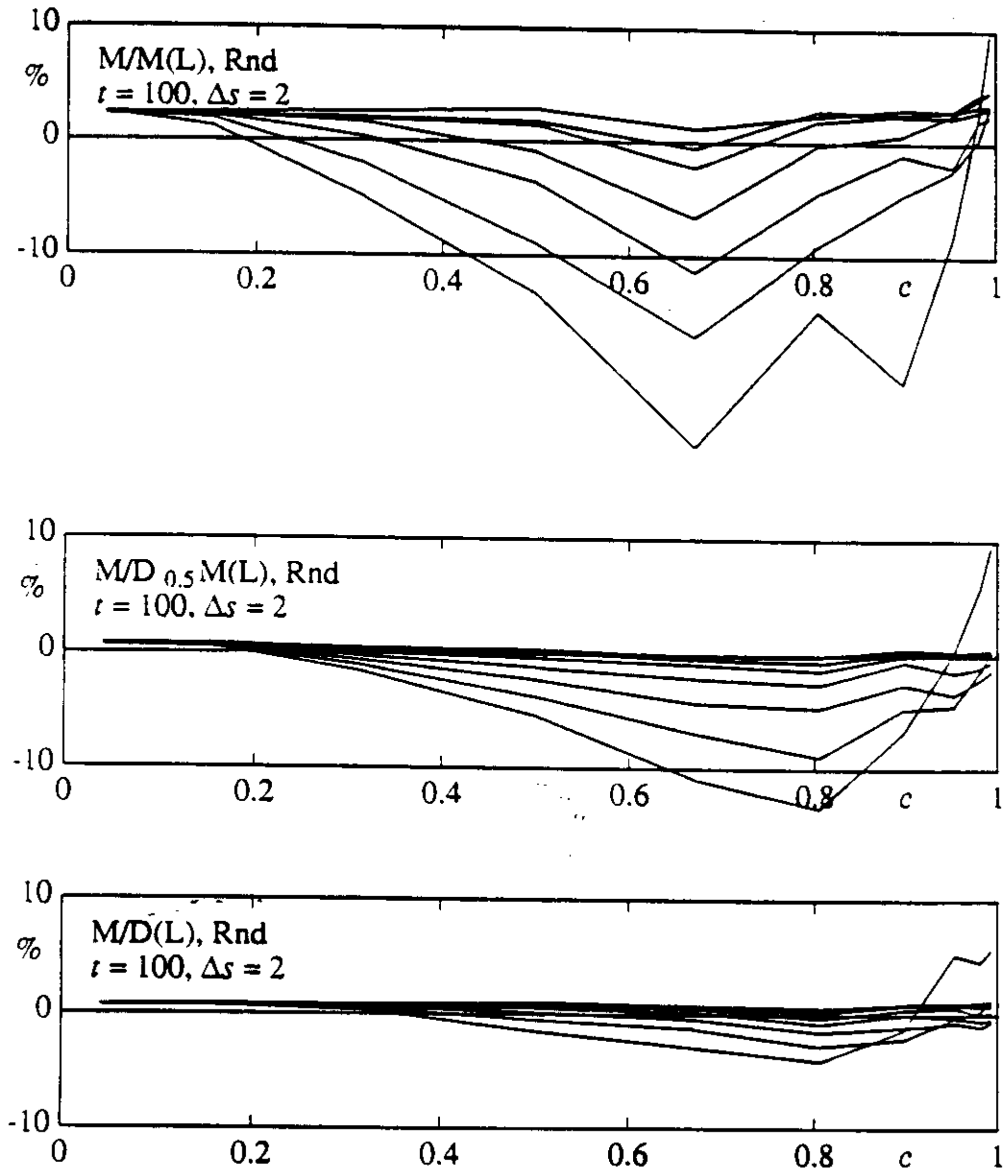


Figure 4.13: Error versus c for Locking/Random

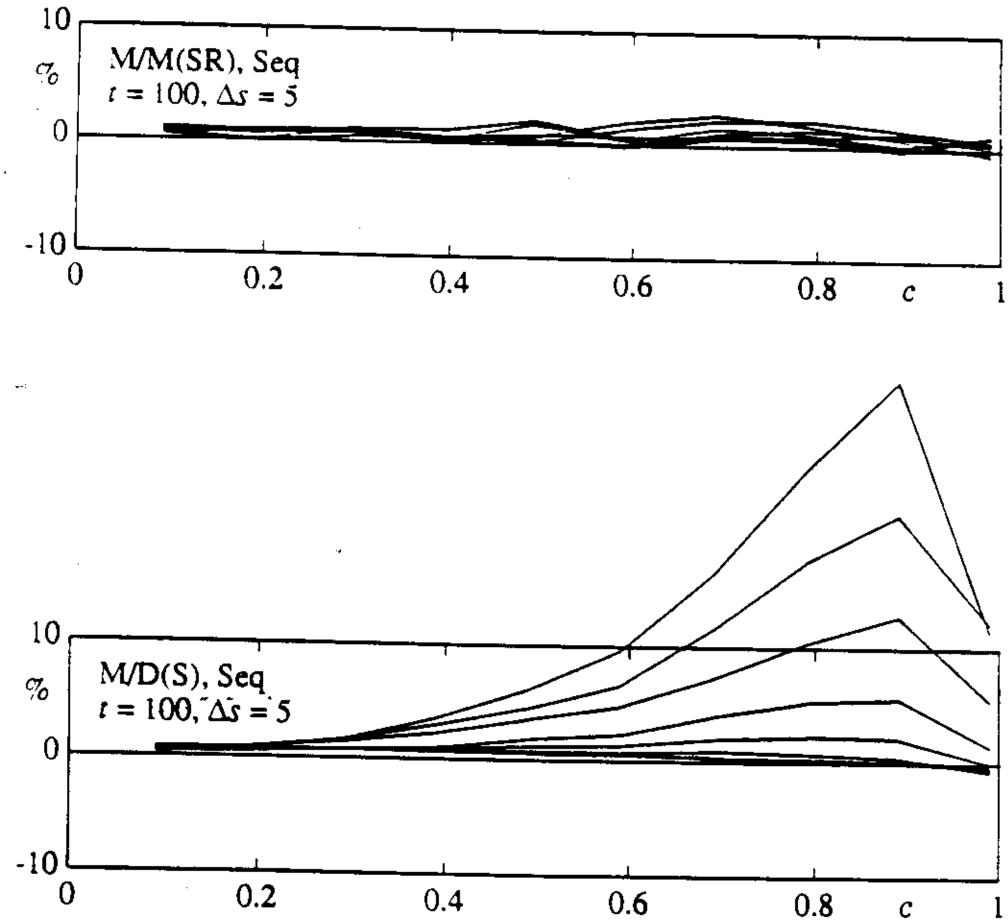


Figure 4.14: Error versus c for Silent/Sequential

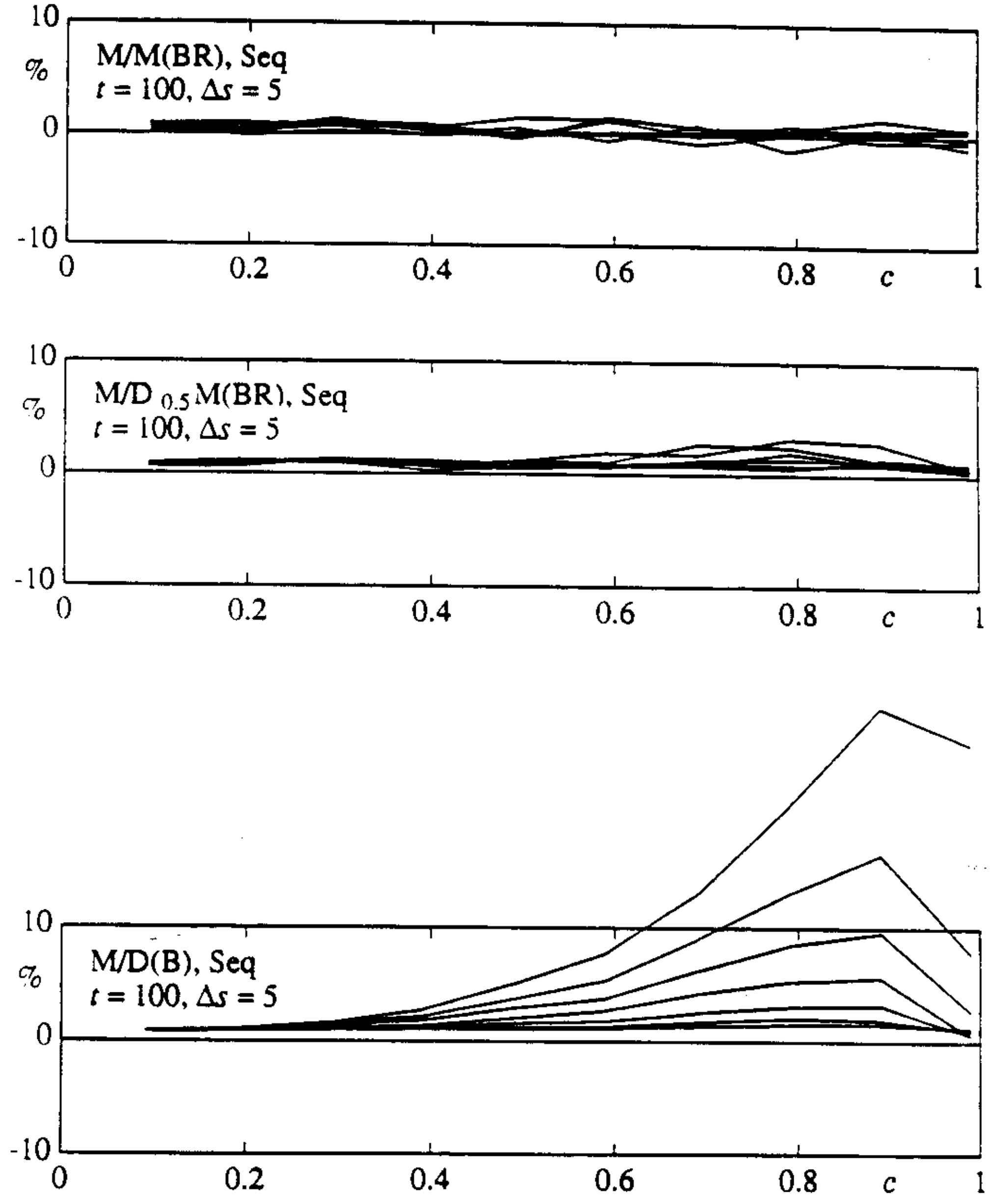


Figure 4.15: Error versus c for Broadcast/Sequential

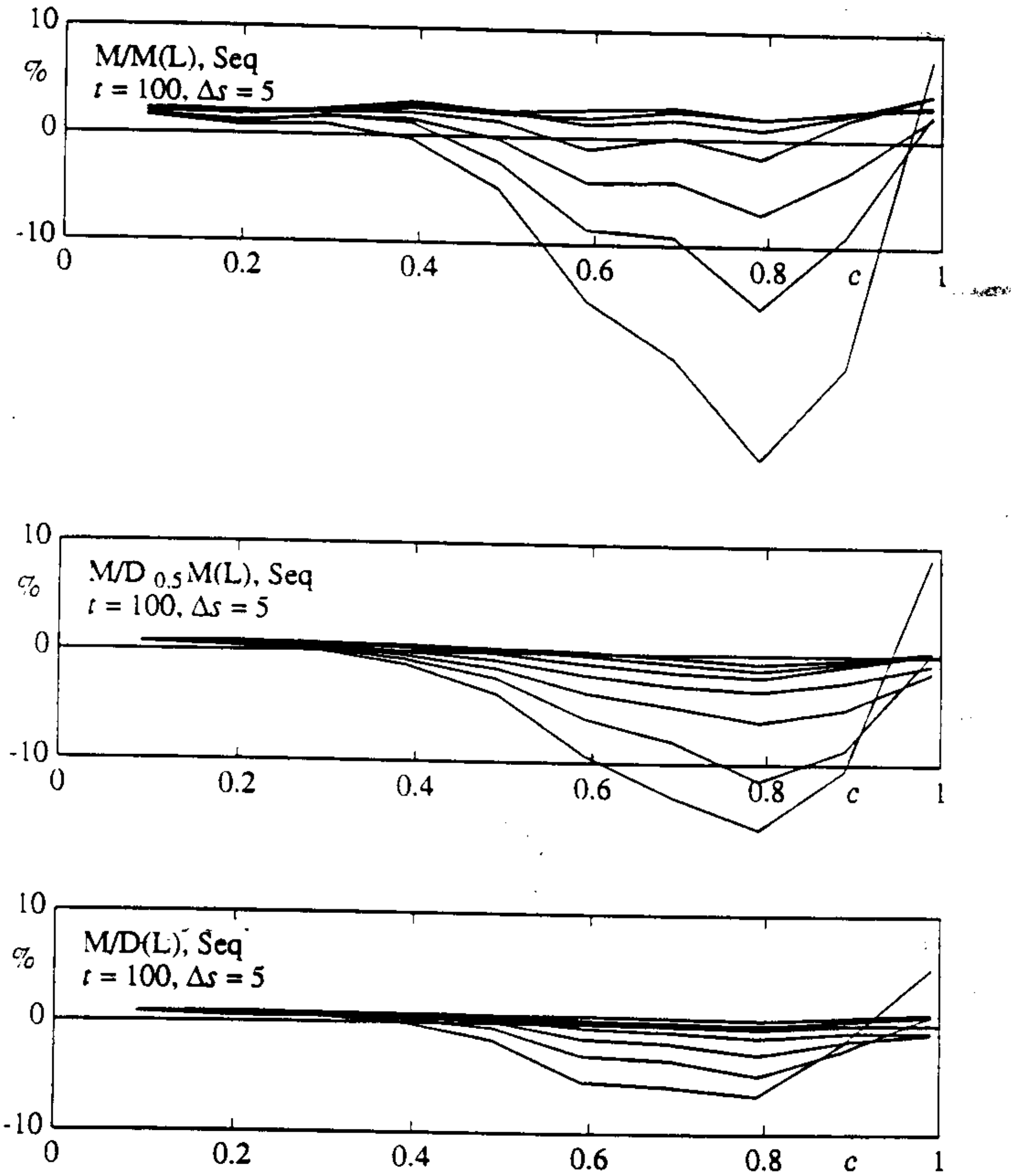


Figure 4.16: Error versus c for Locking/Sequential

More straightforward improvements to the analysis of the partial conflict systems would include arbitrary resource demand patterns, which include variable demand set size and non-uniform resource demand distribution.

CHAPTER 5

Concurrency Control in Databases

5.1 Introduction

Concurrency control must be enforced in order to preserve consistency of a database. Concurrency control algorithms are based on the notion of serializability [11]. The correctness of various proposed concurrency control schemes is well understood [8,10,37]. However, the performance of those protocols is not.

5.1.1 Terminology

A database is a collection of data that may be shared by many users who update the database with transactions. A centralized database resides only on one computer, while a distributed database resides on the network of computers. We are concerned here with centralized databases.

A transaction is an execution of a computer program that accesses, and possibly updates, the database. A transaction, run alone against the database, produces results that are correct, i.e., results that are expected. Thus, a number of transactions run serially produce correct results, i.e., they do not violate data integrity in the database. However, we want transactions to run concurrently in order to obtain results faster. The concurrent execution of transactions may produce incorrect results, and so a database management system must control the timing of the read and write operations against the database.

During the the concurrent execution of the transactions, the read and write operations from different transactions are interleaved. The specific order of the execution of those operations we call a schedule. If the result of a schedule is equivalent to the result obtained by any serial execution of the transactions, then we call that schedule a serializable schedule. Since a serial execution gives correct results, so does a serializable schedule.

The purpose of concurrency control can, therefore, be defined as making sure that the operations of the concurrent transactions are executed in a serializable schedule. The three major types of concurrency control algorithms are locking, timestamp ordering, and optimistic concurrency control.

The use of locking implies that every transaction must lock a resource in shared (exclusive) mode before reading (updating) it. An example of such a protocol, called two-phase locking, appears in [11]. In two-phase locking, for each transaction there is a first phase during which new locks are acquired (growing phase) and a second phase during which locks are only released (shrinking phase). The serializability in timestamp ordering algorithms is achieved using unique sequence numbers, called timestamps, assigned to each transaction. A typical timestamp ordering algorithm is the basic timestamp ordering described in [6]. Optimistic schemes are based on the idea that concurrent updates of the same data elements are infrequent and, therefore, appropriate actions to preserve data integrity should only be taken when they occur. Examples of mechanisms along this line can be found in [16] for centralized databases and in [5] for distributed ones. These different algorithms can be combined, as described in [6]. In this dissertation we analyze optimistic concurrency control schemes.

5.1.2 Database Model

A computer system is usually modeled as one which is multiprogrammed. consisting of one or more CPU's (processors) and one or more IO devices (discs). A database is a set of data items which transactions read and write. All items read (written) by a transaction form a read set (write set) of that transaction. We assume here that the read set is equivalent to the write set. and we call it a data set. The number of items in the database represents the database size. The number of items in the data set represents the transaction size.

In the analysis below, we are given the arrival process of the transactions. usually taken to be Poisson. and the transaction service time distribution. The transactions in optimistic concurrency control have a read. a validation. and (possibly) a commit phase. During the read phase a transaction reads and updates data items. *The updates done in the read phase are stored locally in our model.* In the validation phase, the data set is checked for a possible conflict (which might result in violating data integrity) with some other transaction(s). For example, the data set might be reread and compared with the earlier read. If there was no conflict, the transaction commits, making all the updates permanent (i.e., the update is written into the centralized database). If there was a conflict, the transaction aborts and restarts.

Transactions may read all the items in their data sets at the beginning of their execution. We call this a static data access scheme. In a dynamic data access scheme, transactions read items from their data sets during the execution, as the need arises.

Upon the commit of a transaction, the system may tell all the other transactions about the commit. In that way, those transactions that are in conflict with the committing transaction can abort and restart immediately. We call this a broadcast scheme. In the non-broadcast scheme, the system remains silent upon commits, in which case all other transactions always proceed with the execution, even if they will eventually conflict with the committing transaction. This scheme is called a silent scheme.

5.1.3 Previous Work

There has been considerable research devoted to evaluating the performance of optimistic concurrency control [18,23,25,34,19,20,16,17,7,30]. Manasce and Nakanishi presented the analysis of the Static/Silent system using a two-level decomposition method, which used iteration between the two levels and required the solution of a two-dimensional Markov chain [18]. A decomposition method which obviated the need for iteration and required the solution of a one-dimensional birth-death process was developed by Ryu and Thomasian in [23,25]. Morris and

Wong presented an analytic solution method in [19,20], which is based on a correction applied to a result appearing in [18]. They considered the Static/Silent scheme (non-exclusive accesses are allowed in [20]). Thomasian and Ryu, in [34], generalized that approach to the case of multiple transaction classes and applied it to Static/Silent. Static/Broadcast, Dynamic/Silent, and Dynamic/Broadcast. They also allowed a general distribution of processing time requirements for transactions, while only the exponential distribution was considered in [19.20]. In [34] the assumption was made that the conflict process was Poisson. A nice critical investigation of the assumptions made in the models used in past studies and their implications is given in [1].

5.1.4 Infinite Servers

The model analyzed herein has an infinite number of servers, i.e., there are an infinite number processors. The database access time is considered negligible. The transaction arrival process is Poisson. The processing time required by transactions are distributed according to the D_qM distribution defined in Section 2.

In the remainder of this chapter we describe databases as ISR systems and give results for the seven different concurrency control models obtained through mapping. We also give examples of the use of the winner queues in modeling transactions.

	Condition
1	The data access is static.
2	Every transaction has its own processor.
3	The database server has very high capacity, compared to the capacity of each individual processor.

Table 5.1: Conditions for Databases as ISR Systems

5.1.5 Databases as ISR Systems

A database is a system with independently shared resources with infinite servers, defined in Chapter 1, if the conditions listed in Table 5.1 are met.

Table 5.2 contains the equivalent terminology when considering databases as ISR systems.

5.2 Modeling Database Systems

Consider a database system running under a server with high processing capacity. The database is accessed by workstations. In the following subsections we describe database examples of the concurrency control schemes and their overhead, with q-deterministic service time distributions and partial restarts.

	Database	ISR System
1	DBMS	ISR system
2	record	resource
3	database size	system size
4	transaction	customer
5	data access	resource demand
6	data set	demand set
7	transaction size	demand set size
8	CCS	CRS

Table 5.2: Equivalent Terminology in Databases and ISR Systems

5.2.1 Silent Scheme

The database server processes three types of requests submitted by transactions that are running on the workstations. The requests are *open-transaction*, *close-transaction*, and *abort-transaction*. With the *open-transaction* request, a transaction supplies the list of the database record keys which specifies what records the transaction is going to update. Upon receiving the request, the server gives a unique identification number, *tid*, to the transaction, saves the contents of the records requested into a data buffer, *oldbuf[tid]*. [1] The server

[1]In addition to the record ID, a version number, *vid*, may be associated with every record in the database which increments every time the record is updated. In this way we can tell if the contents of a record has changed by comparing that record's present and old *vid*'s instead of comparing the actual present and old contents of the record.

then responds with *transaction-started*, giving the *tid* and the contents of those records and telling the transaction that it may start processing the data.

With the *close-transaction* request, a transaction supplies its *tid* and the new contents of the records. The server first checks whether the present contents of the records is identical to the contents of the buffer *oldbuf[tid]*. [2] If so, the server updates the database changing the contents of the records to the new contents, replies *transaction-committed* and then releases the buffer *oldbuf[tid]*. If the present and old contents of the records differ, the server releases *oldbuf[tid]* and replies *transaction-aborted.*

With the *abort-transaction* request, a transaction supplies only its *tid*, and the server releases *oldbuf[tid]*, replying *transaction-aborted.* The server may abort a transaction even before it receives the *close-transaction* or the *abort-transaction* requests. If the concurrency control scheme is such that the server can tell if a transaction is going to be aborted even before it receives any of the two above requests, the server then sends *transaction-aborted.*

5.2.2 Broadcast Scheme

The database server processes three types of requests submitted by transactions that are running on the workstations. The requests are *open-transaction*, *close-transaction*, and *abort-transaction.* The server also maintains a list of *tid*'s associated with each database record. The list includes all those transactions

[2] See Footnote 1.

that are presently accessing the corresponding database record. With the *open-transaction* request, a transaction supplies the list of the database record keys which specifies what records the transaction is going to update. Upon receiving the request, the server gives a unique identification number, *tid*, to the transaction. Then, the server appends the *tid* to the lists of every record specified by the *open-transaction* request. After that, the server responds *transaction-started*, giving the *tid* and the contents of those records and telling the transaction that it may start processing the data.

With the *close-transaction* request, a transaction supplies its *tid* and the new contents of the records. The server updates the database changing the contents of the records to the new contents, removes the *tid* from the lists of the records updated, and replies *transaction-committed*. Then, the server compiles the list of all the transactions presently accessing the just updated records, and sends *transaction-aborted* to those transactions.

5.2.3 Locking Scheme

The database server processes three types of requests submitted by transactions that are running on the workstations. The requests are *open-transaction*, *close-transaction*, and *abort-transaction*. The server also maintains a queue of *tid*'s associated with each database record. The queue includes all those transactions that are presently accessing the corresponding database record, queued in the order of arrival of *open-transaction* requests.

With the *open-transaction* request, a transaction supplies the list of the database record keys which specifies what records the transaction is going to update. Upon receiving the request, the server gives a unique identification number, *tid*, to the transaction. Then, the server appends the *tid* to the queues of every record specified by the *open-transaction* request. After that, the server does not reply until all the transactions that were ahead in any of the queues have committed. The server saves the list of records keys requested by the transaction. Only when the *tid* moves at the head of the queues of all the records requested will the server respond with *transaction-started*, giving the *tid* and the contents of those records and telling the transaction that it may start processing the data.

With the *close-transaction* request, a transaction supplies its *tid* and the new contents of the records. The server updates the database changing the contents of the records to the new contents, removes the *tid* from the queues of the records updated, and replies *transaction-committed*. Then, the server compiles the list of all the transactions presently in the head of the queues of the just updated records. The server checks for *tid* of each of those transactions whether the *tid* is in the head of each of the records the *tid* is accessing. To all those transactions for which the above is true, the server sends *transaction-started*. This locking scheme is deadlock free.

5.2.4 Modeling Finite Servers with Extended Mapping

Consider a system with an infinite number of servers, uniform data access, and with a total of t database records. Let the transactions always access exactly one record, in which case we are talking about a single-request ISR system. The conflict measure for such a system is $c_{old} = 1/t$.[3]

Let us now replace the infinite server assumption in the above system with a finite number of servers, say m. Furthermore, let the transactions access those m servers independently of each other (imagine a random switch routing the transactions to the servers), and let the probabilities of access be the same for all the servers. Then, we can identify this system to be another infinite-server system where the servers become the resources in addition to the database records. In such a system the total number of resources increases to $t+m$, the transactions demand exactly two resources (the system is not single-request any more), and the conflict measure becomes

$$c_{new} = \frac{1}{m} + \frac{m-1}{m} c_{old} \tag{5.1}$$

5.2.5 Modeling Random Delay with Redraw

Suppose that every time a transaction runs, it encounters delays of different length. Those delays may be the communication delays, or they may be the delays due to interactive execution of the transaction. A typical interactive delay

[3]Note that for the single-request systems it is irrelevant whether the resource demand pattern is random or sequential.

would be making airplane ticket reservations in a database of an airline company. The random delay upon every transaction run may be modeled by *redrawing the service time upon every restart of the transaction.*

5.2.6 Modeling Useful Work with Partial Restarts

Consider again an airline company's database system. In the process of making a change of the ticket reservation, the clerk first accesses the database record which appears displayed on the screen. Then the clerk talks to a customer and corrects the incorrectly spelled name of the customer in the record, and then the change is submitted and sent to the remote database server. If the database aborts the transaction, it is unnecessary to go back to the interactive mode and force the clerk to manually correct the spelling of the name. Instead, the process of aborting the transaction and resubmitting the update request is kept transparent to the user (clerk). So, while the first processing of the transaction took a relatively long time due to the interactive delay, every one of the restarts take a much shorter processing time. We can say that the interactive delay was "useful work" since it will not be repeated even though the transaction run was unsuccessful. This difference in the length of the initial and restarted transaction service time may be modeled my making the restarts of the transaction *partial*, with $r < 1$. Furthermore, while the first service time has a specific distribution according to the length of the interactive delay, the restarts may have a different distribution. Thus, in addition to the restarts being partial, they could be drawn from another distribution. Such cases may be modeled with a D_qM distribution for the first service, and a $D_{q_r}M$ distribution for the restart.

5.2.7 Modeling Concurrency Control Scheme Overhead with D_qM Distribution

Assuming the overhead of a concurrency control scheme per transaction is fixed and equal to $\overline{h} = b/\mu$, the overhead may be modeled by adjusting the q-deterministic service time distribution with a mean $\overline{x} = 1/\mu$ as follows.

We define a new system with zero overhead. The average service time of the new system, $\overline{x'} = 1/\mu'$, is greater than the average service time of the initial system, $\overline{x}$, by the amount of overhead, $\overline{o}$. Thus

$$\frac{1}{\mu'} = \frac{1}{\mu} + \frac{b}{\mu} = \frac{1+b}{\mu}$$

The service of every customer in the new system is greater than in the old system by the amount of overhead, $\overline{h}$, and thus

$$q' = \frac{q+b}{1+b}$$

We now define the load for the new system as

$$\rho' = \frac{\lambda}{\mu'} = (1+b)\rho$$

If $\mathcal{P}_{q,b}(\rho)$ represents some performance measure of the system with q-deterministic service times and fixed amount of overhead per transaction equal to b/μ, then the following holds.

$$\begin{aligned} \mathcal{P}_{q,b}(\rho) &= \mathcal{P}_{q',0}(\rho') \\ &= \mathcal{P}_{\frac{q+b}{1+b},0}[(1+b)\rho] \end{aligned} \tag{5.2}$$

5.2.8 Modeling Restart Overhead with Partial Restarts

The overhead caused by restarting an unsuccessful transaction may be modeled by making the restarts of the transaction require more processing than the first transaction run. This can be achieved by using *partial restarts*, in which the restart-to-initial ratio is greater than one, $r > 1$. In addition to that, since the distribution of the restarts due to the restart overhead changes, we may also use a different distribution of the restarts. For example, if the initial service of the transaction has a $D_{0.4}M$ distribution with mean $\overline{x}$, and if every restart requires additional time equal $0.1\overline{x}$, then the restarts will have mean $\overline{x_r} = 1.1\overline{x}$, a $D_{0.5/1.1}M$ distribution. This is similar to changing the deterministic portion of the service time distribution in order to model the overhead of the concurrency control schemes, described in Section 5.2.7.

5.3 Results for Concurrency Control Schemes

In Figures 5.1 through 5.8 we show results obtained from mapping for silent, broadcast, and locking schemes, with random resource access for varying $0 < \rho < 1$. Figures 5.9 through 5.16 display results for silent, broadcast, and locking, in the case of sequential access.

The total number of records is $t = 100$. For the random case, the transaction size is varied from 2 to 20 with step $\Delta s = 2$. For the sequential case, the transaction size is varied from 5 to 50 with step $\Delta s = 5$.

Figures 5.1 through 5.16 show that in both random and sequential access the results are very close to the simulation results for the silent and broadcast schemes when the service time distribution is memoryless. Even the half-deterministic service time distribution gives results close to the simulation results for the broadcast case. However, pure deterministic service times differ considerably for both silent and broadcast, but only for $\rho > 0.6$.

For the locking scheme the situation is reversed. Only deterministic service times give results close to the simulation results, while half-deterministic and memoryless clearly differ from the simulation for $\rho > 0.7$.

5.4 Concurrency Control Scheme Selection

In this work we do not try to propose "the best" concurrency control scheme. However, we here compare the results obtained for the different concurrency control schemes. The results show that the locking scheme gives the best results, if the optimistic schemes are no-redraw. However, there are certain service time distributions for which broadcast gives better results than locking for certain ranges of system load [15]. In the case of a pure deterministic service time distribution, the results for the broadcast-no-redraw scheme are not worse than the results for the locking. The above comparison does not include for the effect

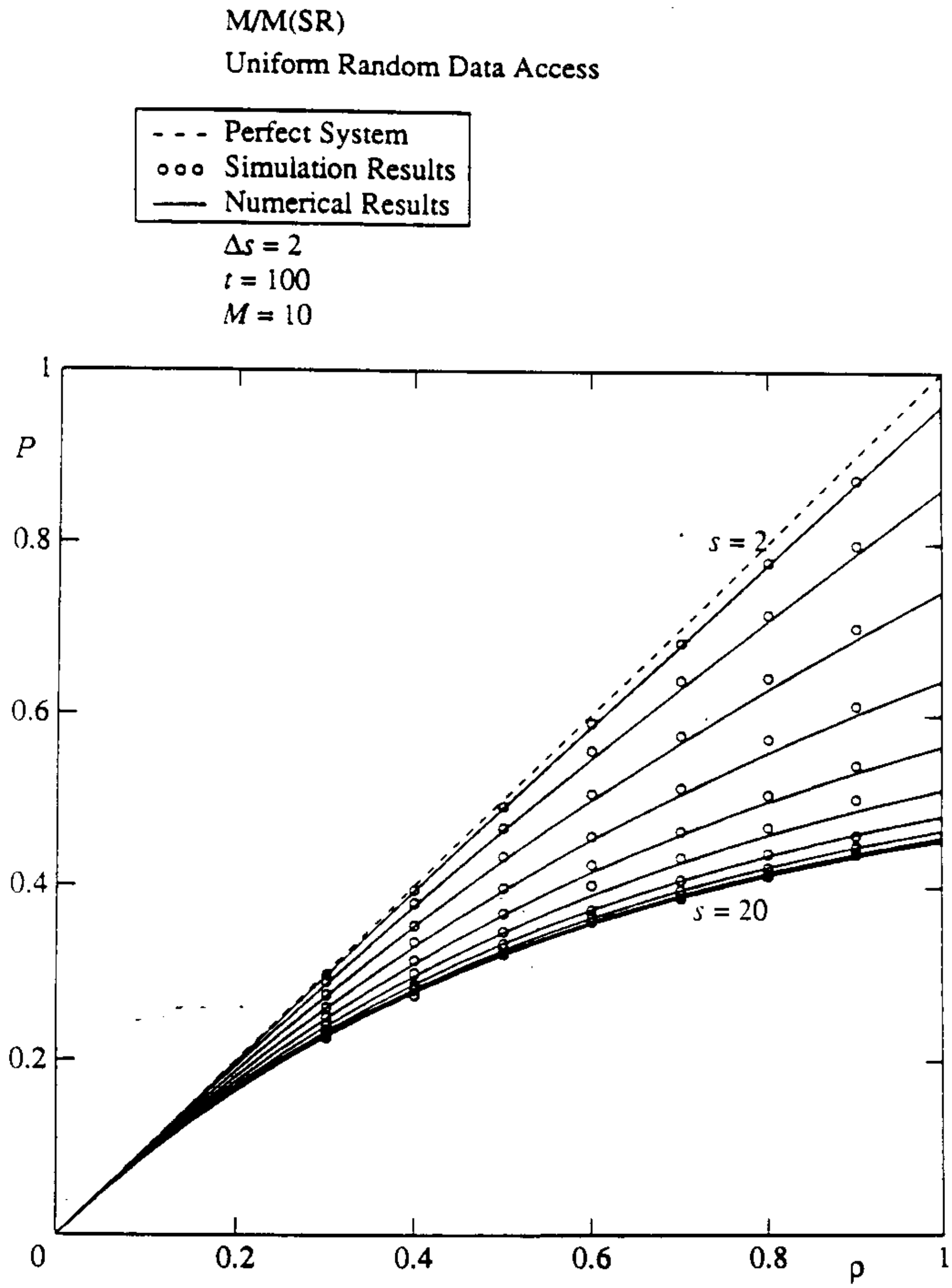


Figure 5.1: Power for Silent/Memoryless/Random

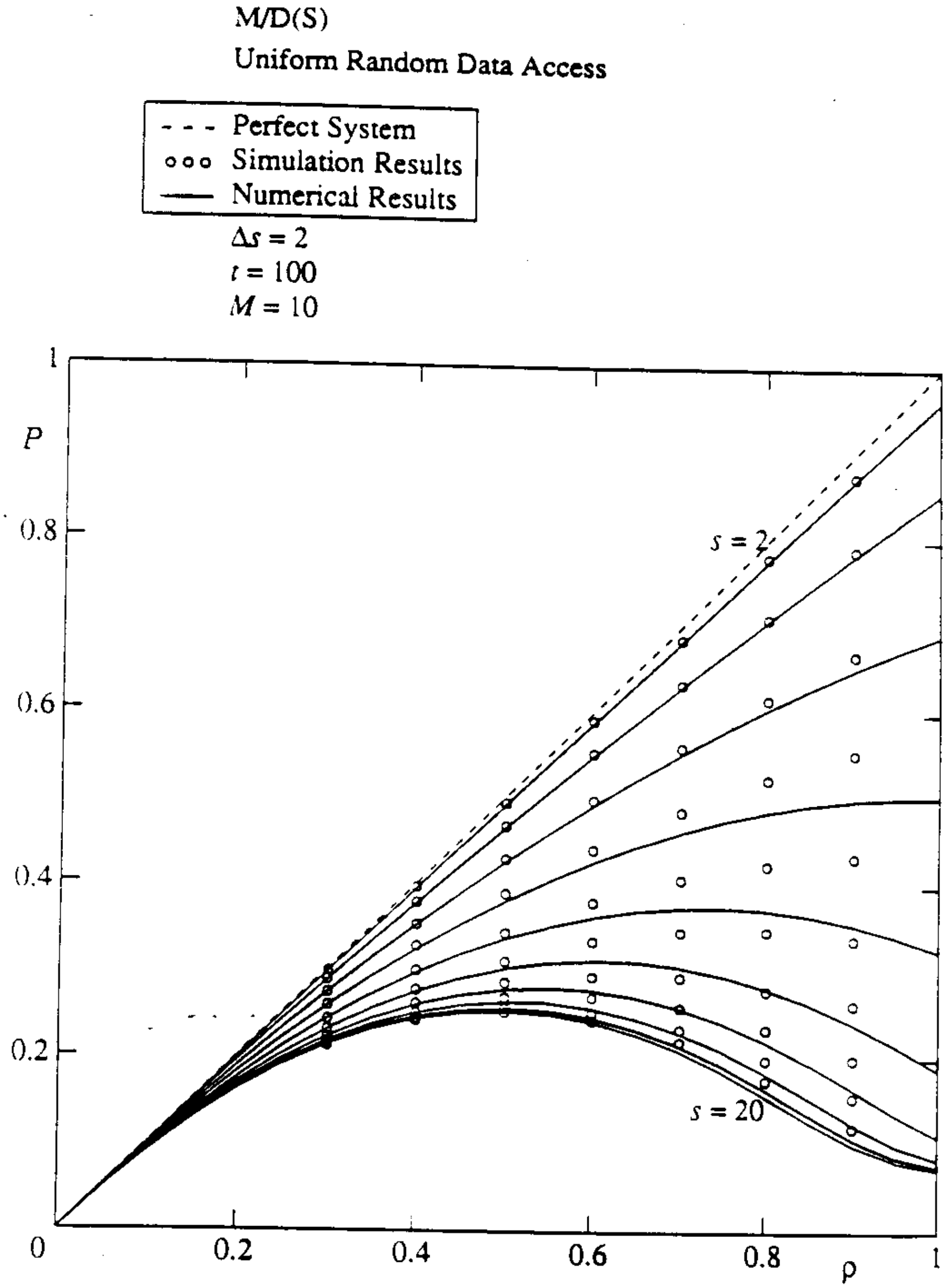


Figure 5.2: Power for Silent/Deterministic/Random

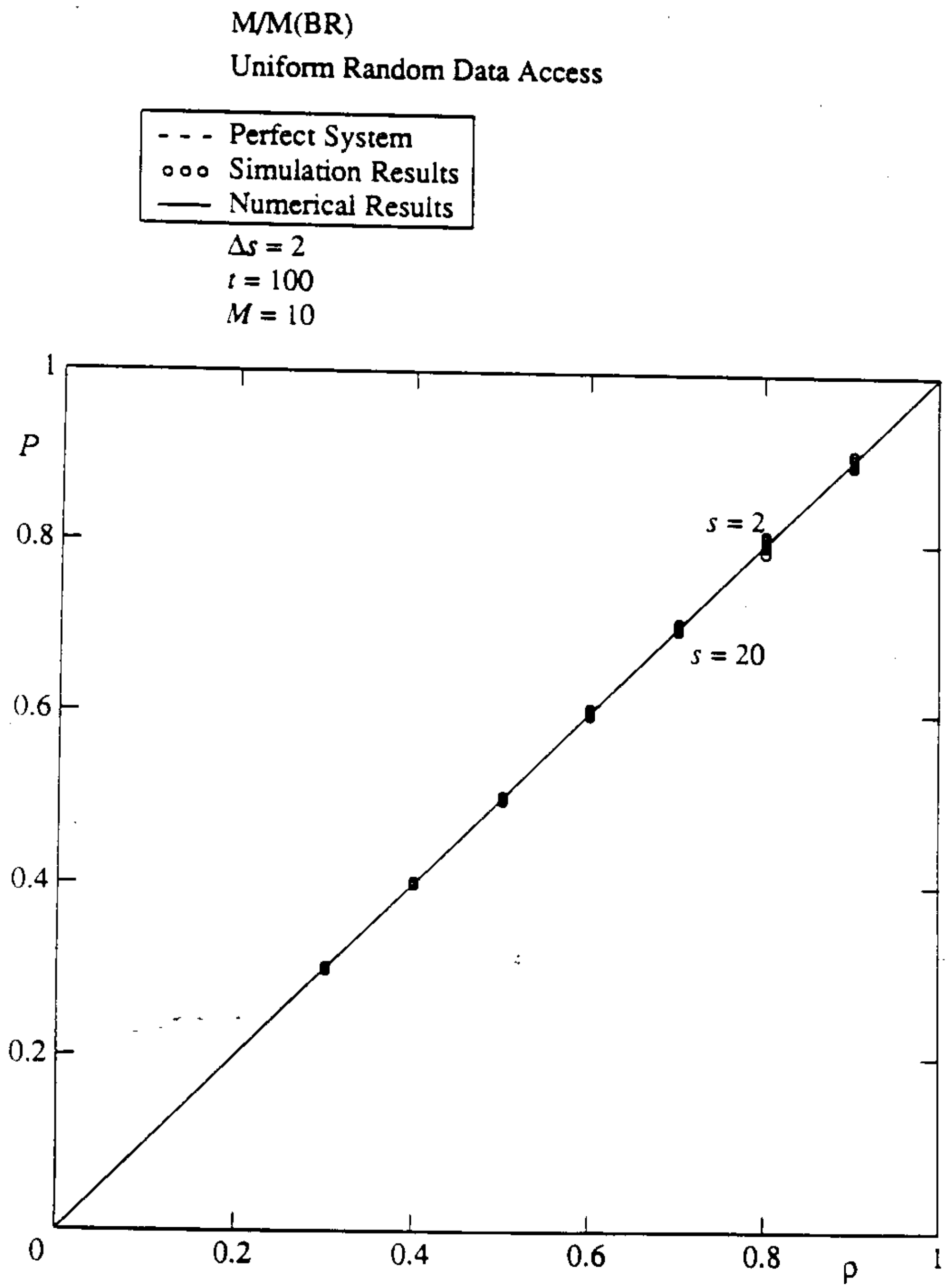


Figure 5.3: Power for Broadcast/Memoryless/Random

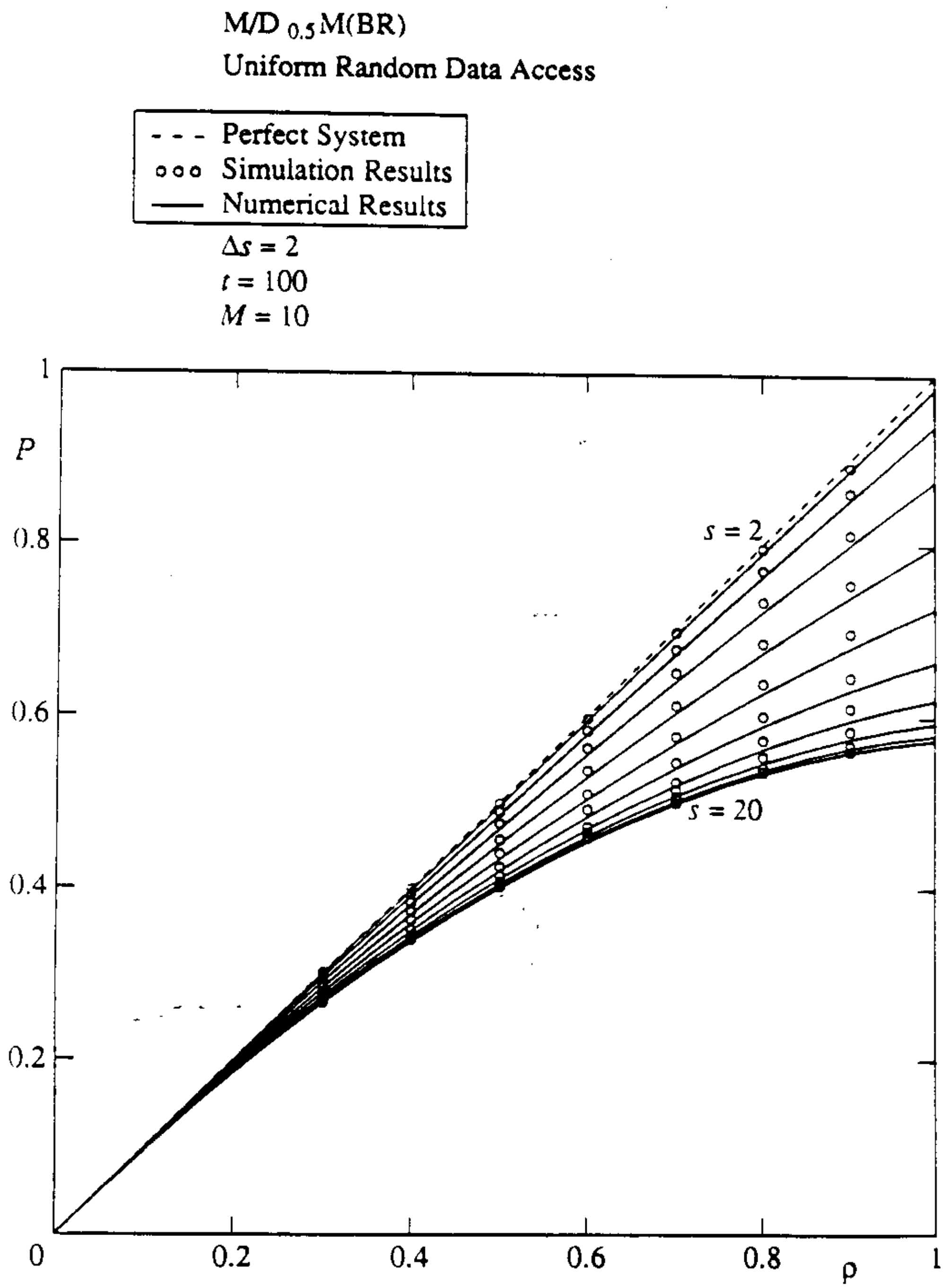


Figure 5.4: Power for Broadcast/0.5-Deterministic/Random

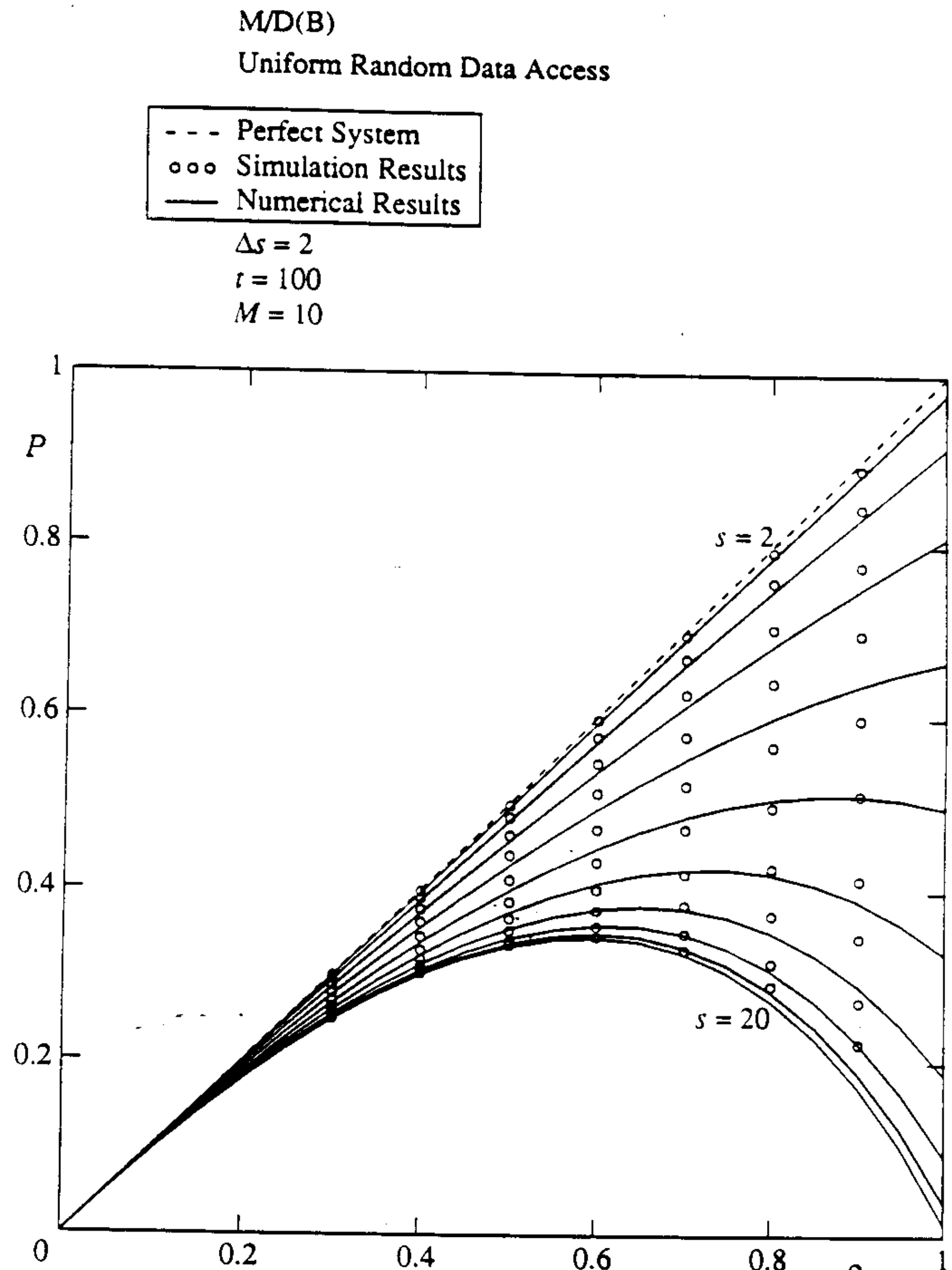


Figure 5.5: Power for Broadcast/Deterministic/Random

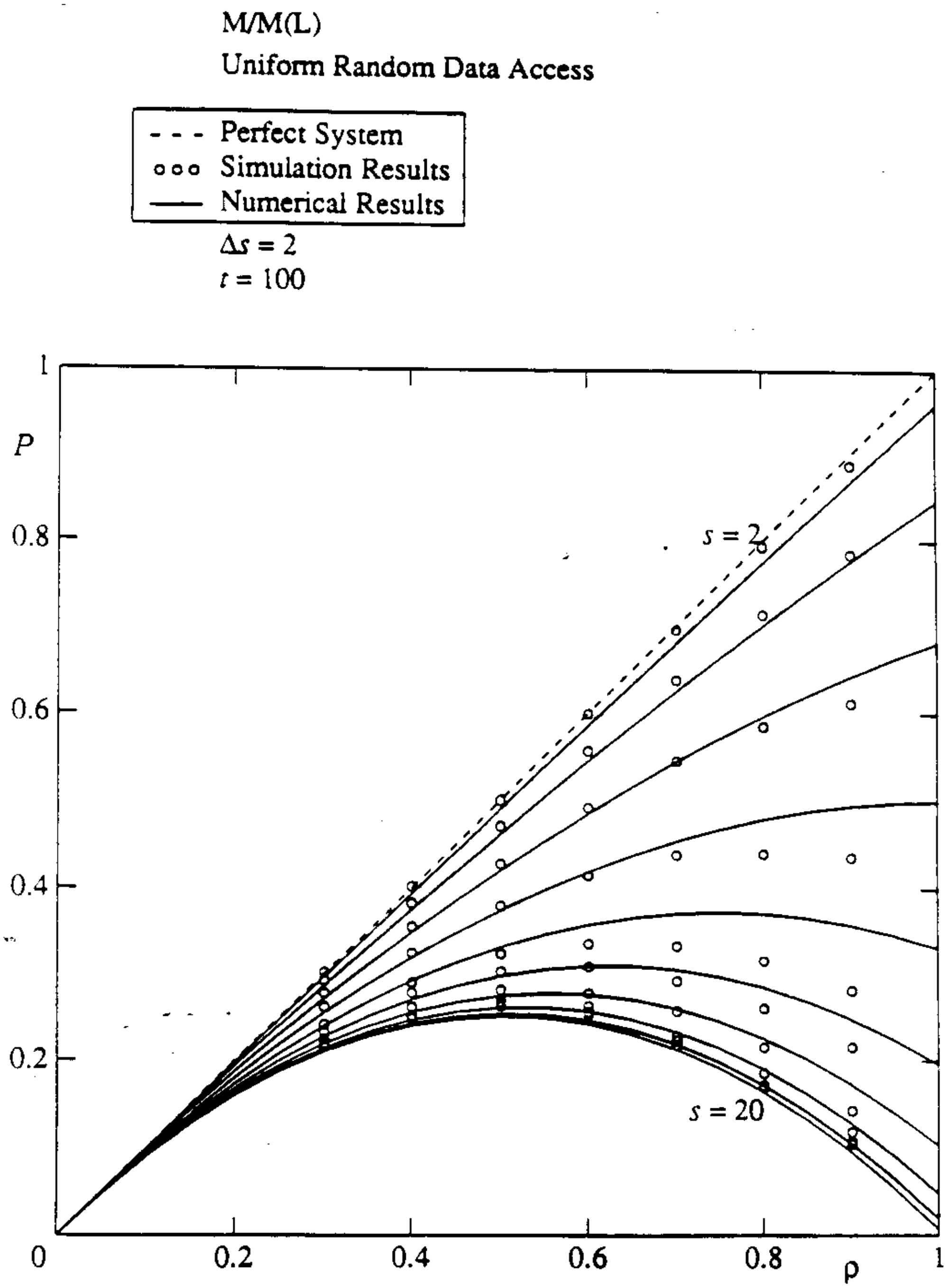


Figure 5.6: Power for Locking/Memoryless/Random

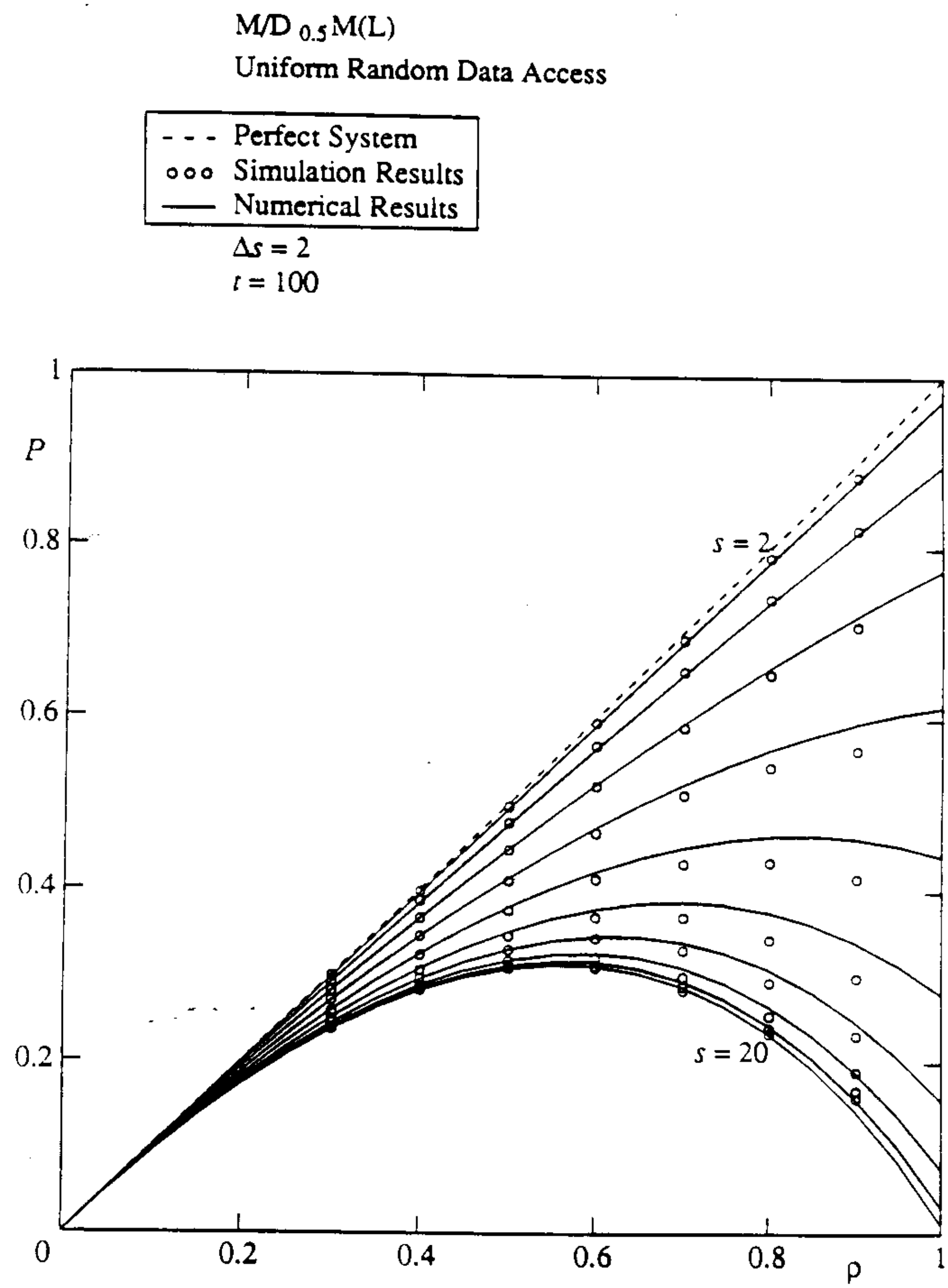


Figure 5.7: Power for Locking/0.5-Deterministic/Random

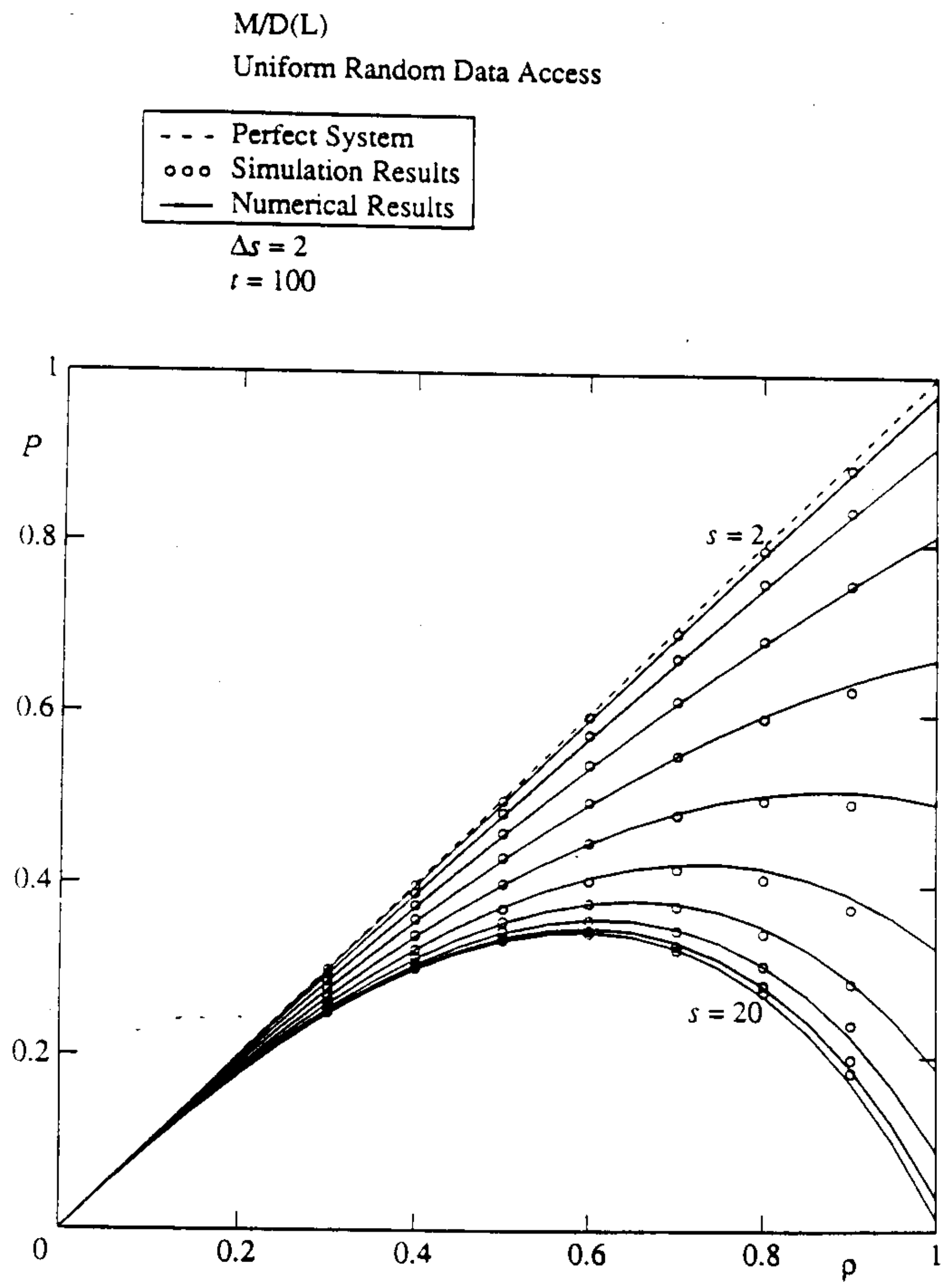


Figure 5.8: Power for Locking/Deterministic/Random

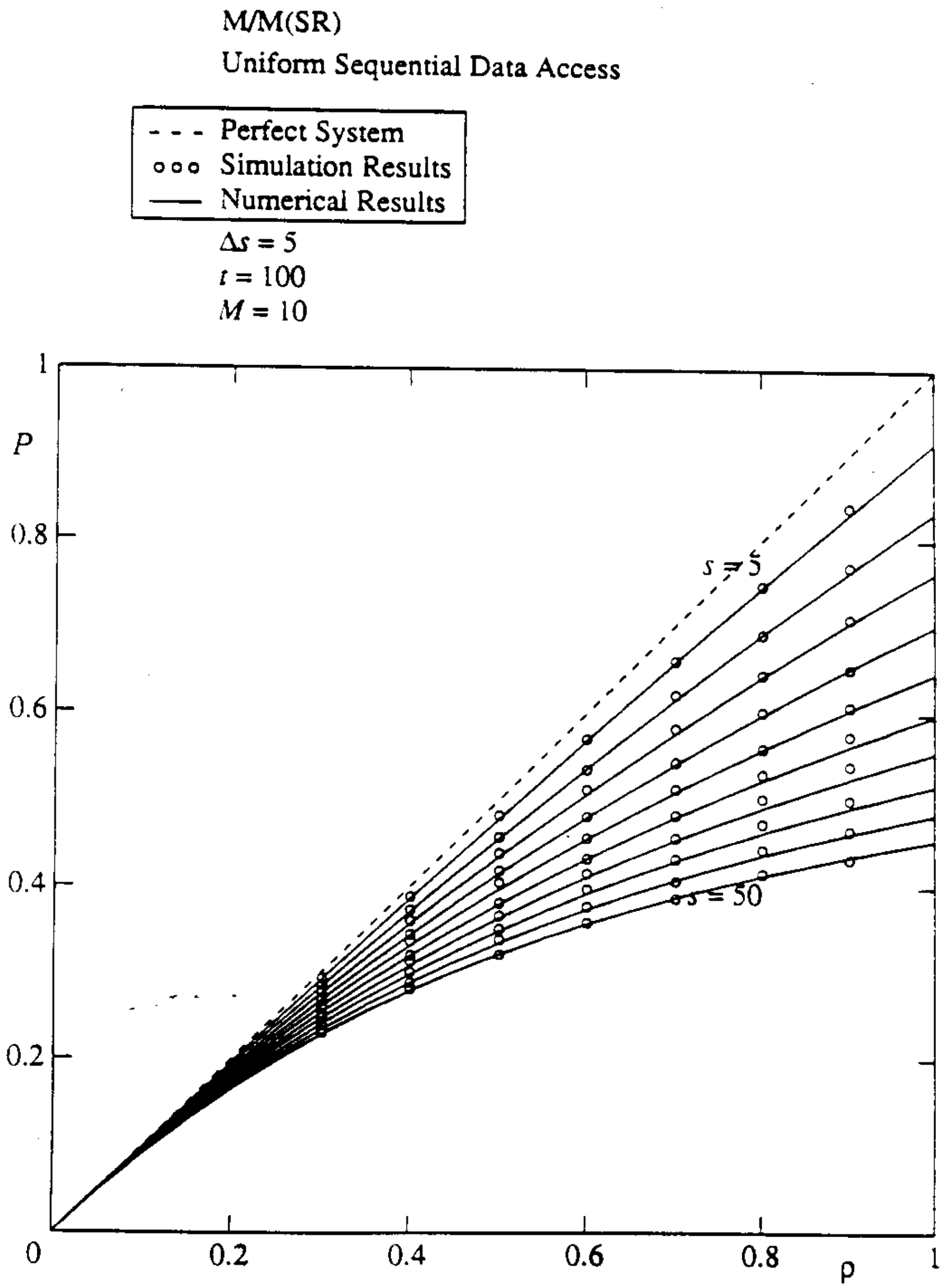


Figure 5.9: Power for Silent/Memoryless/Sequential

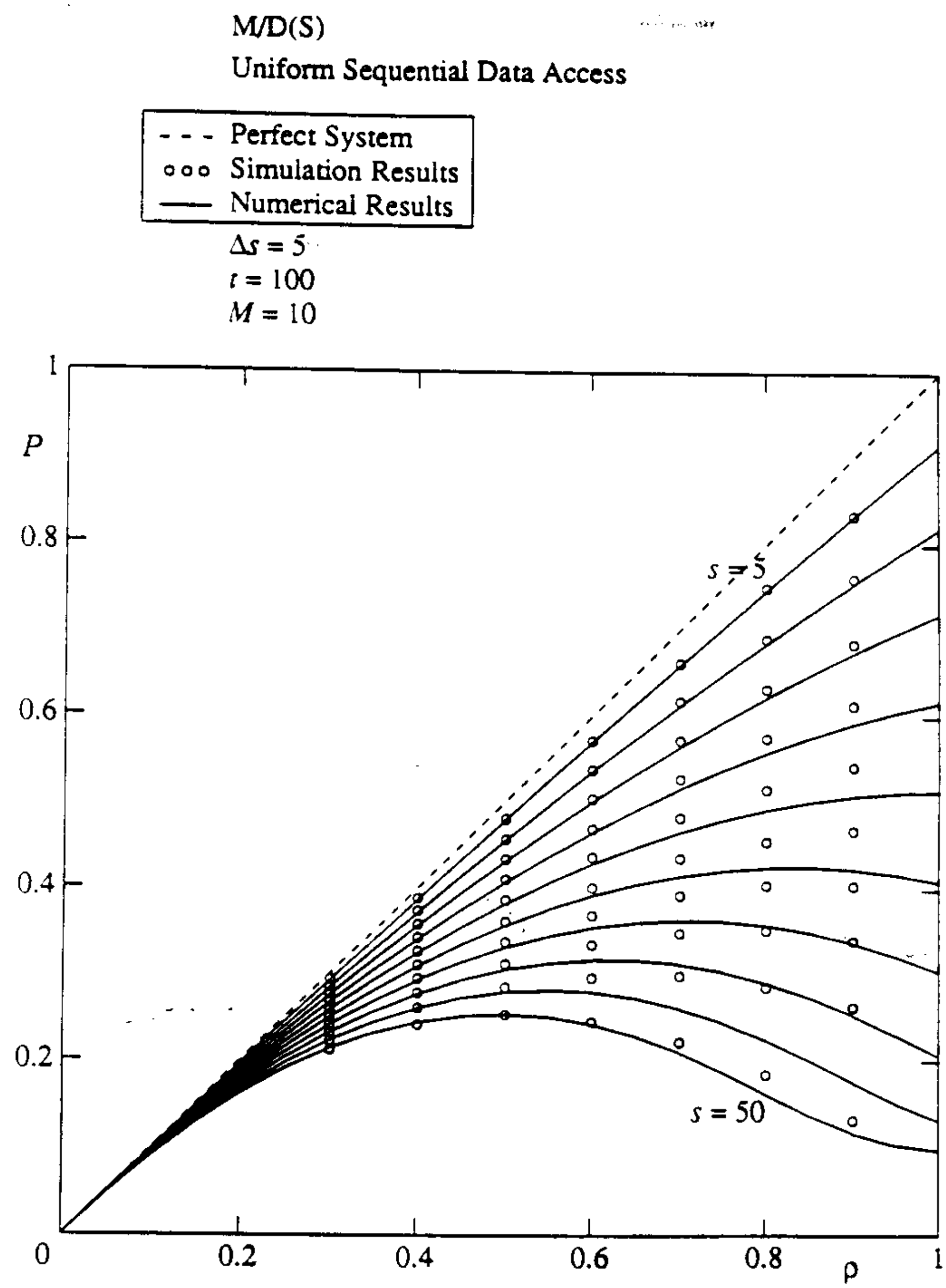


Figure 5.10: Power for Silent/Deterministic/Sequential

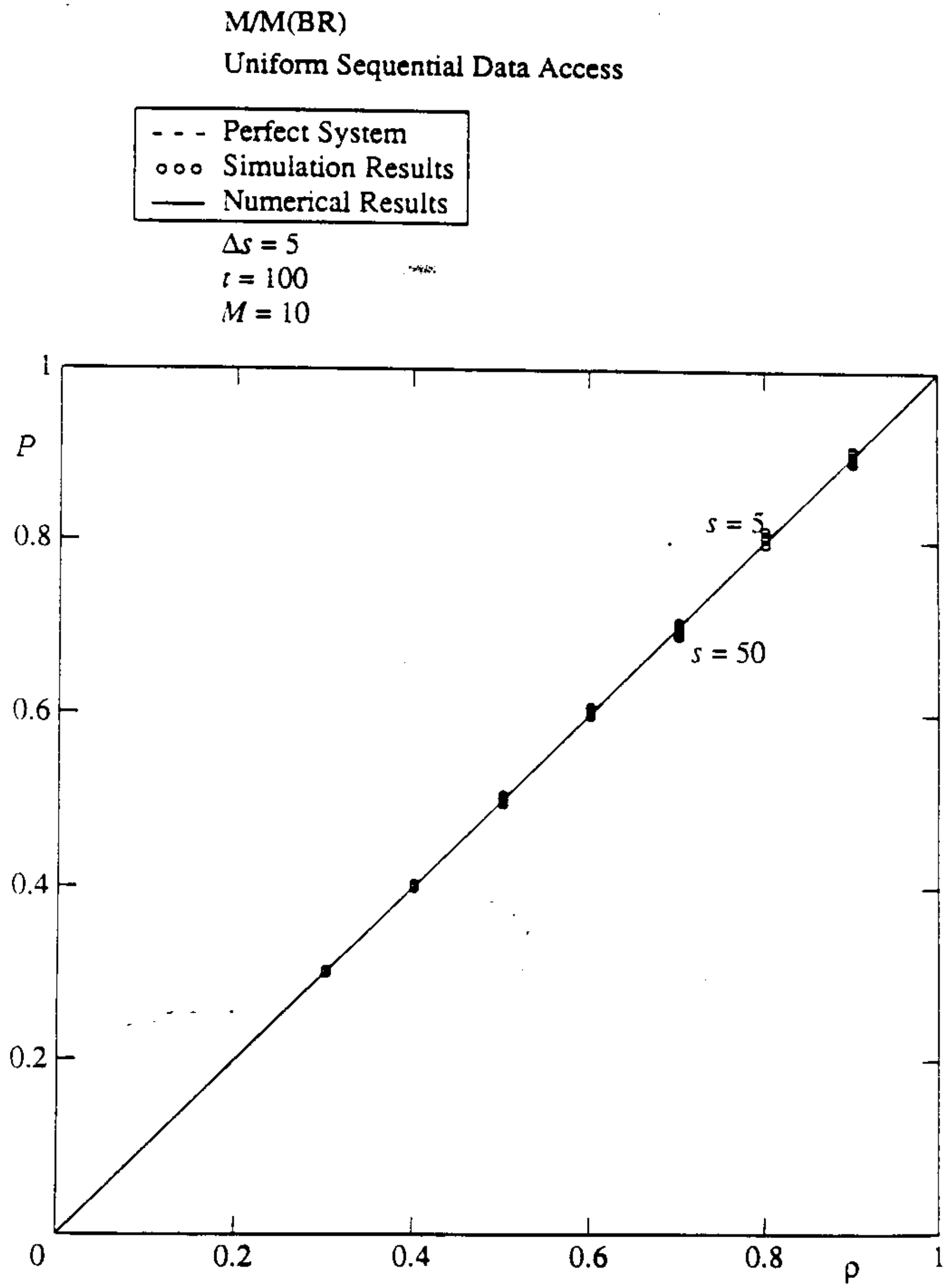


Figure 5.11: Power for Broadcast/Memoryless/Sequential

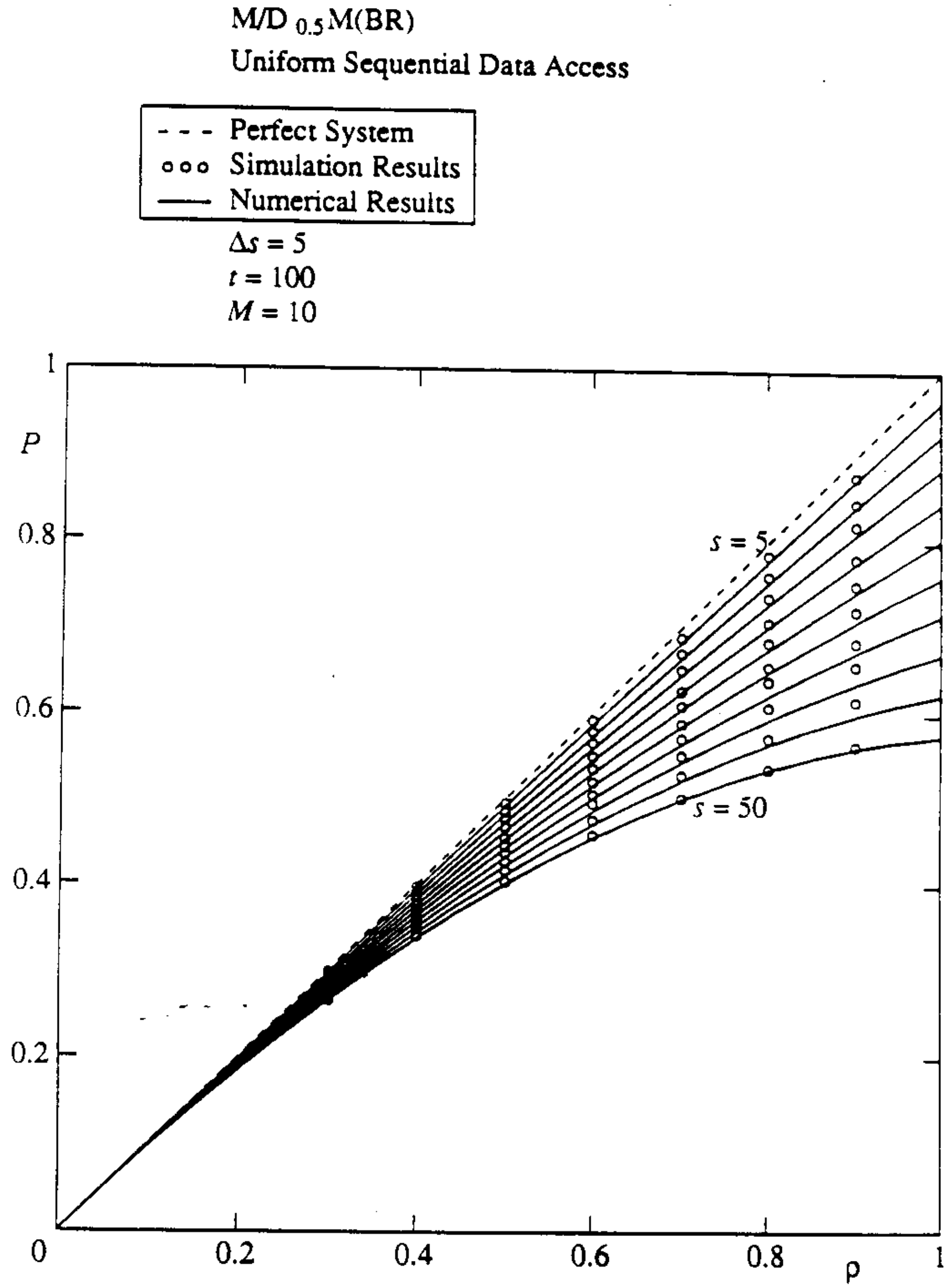


Figure 5.12: Power for Broadcast/0.5-Deterministic/Sequential

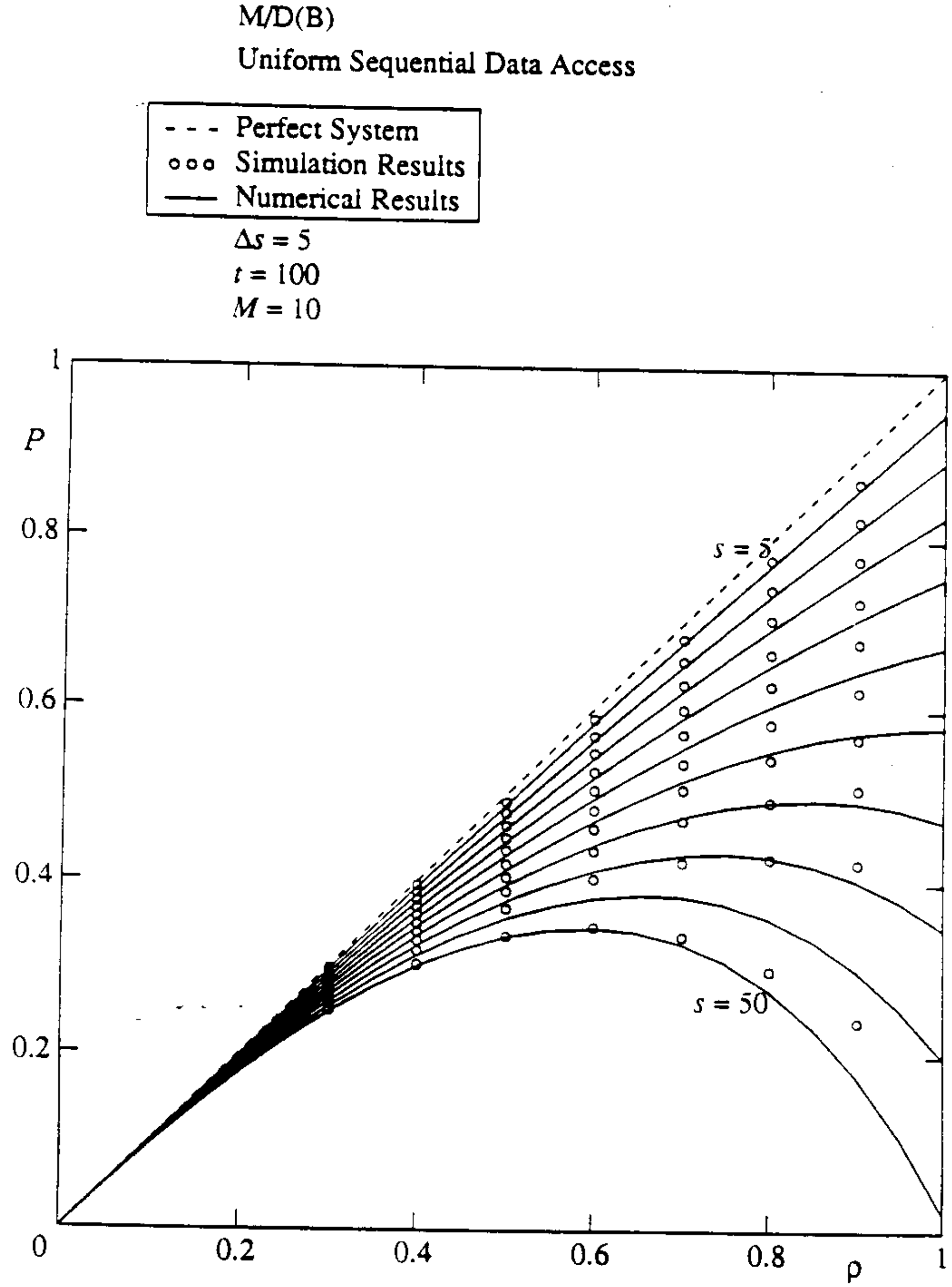


Figure 5.13: Power for Broadcast/Deterministic/Sequential

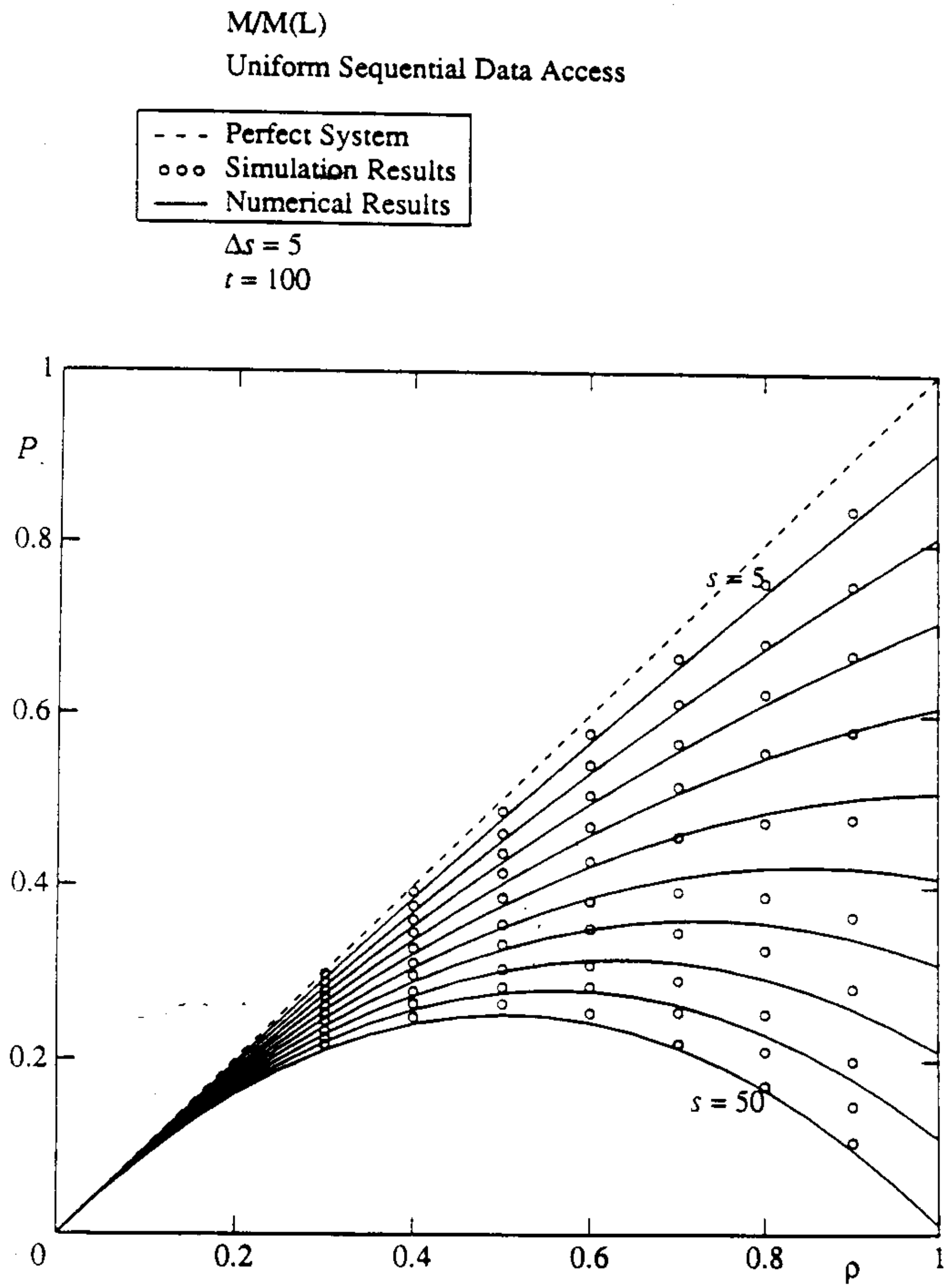


Figure 5.14: Power for Locking/Memoryless/Sequential

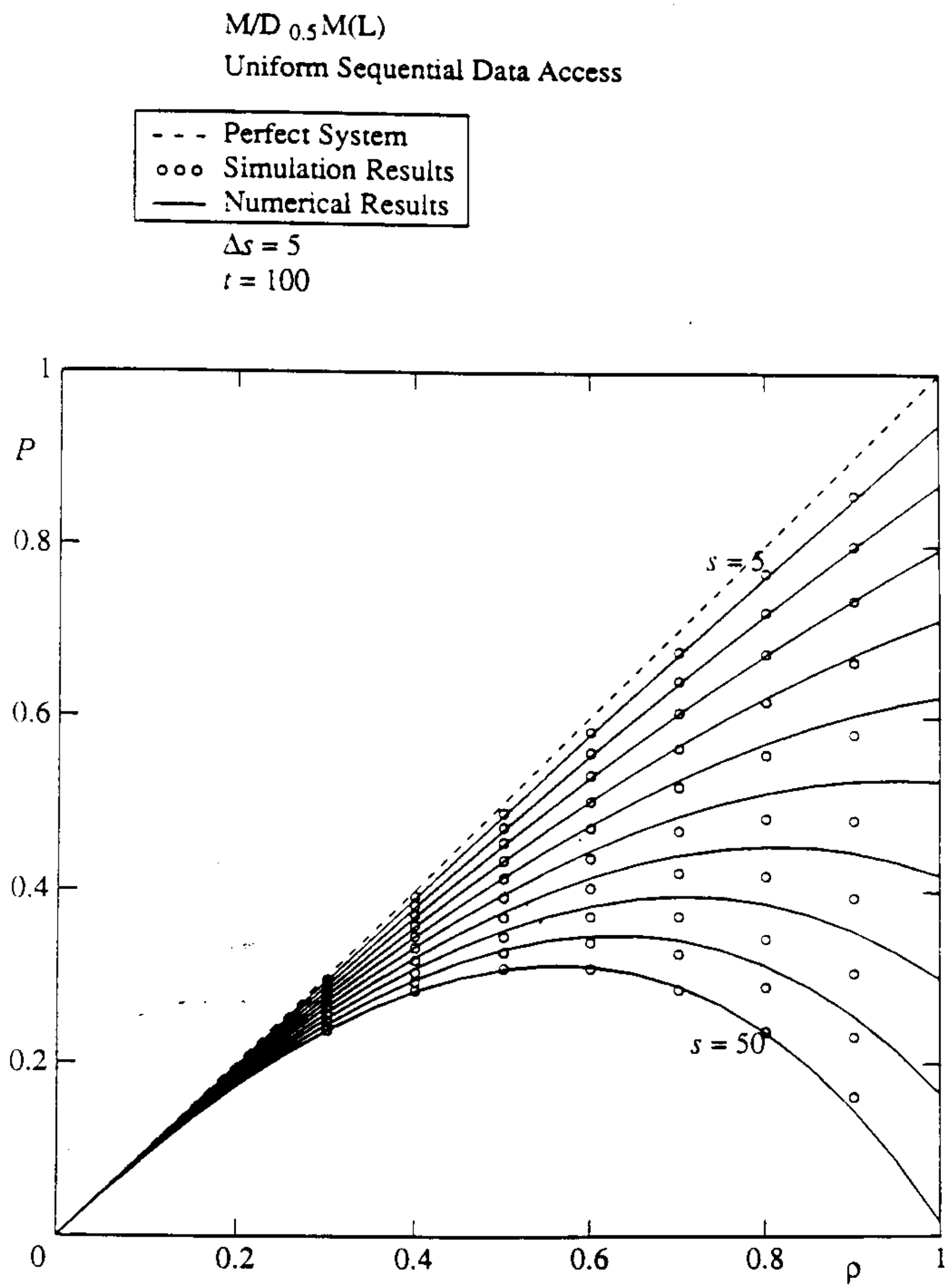


Figure 5.15: Power for Locking/0.5-Deterministic/Sequential

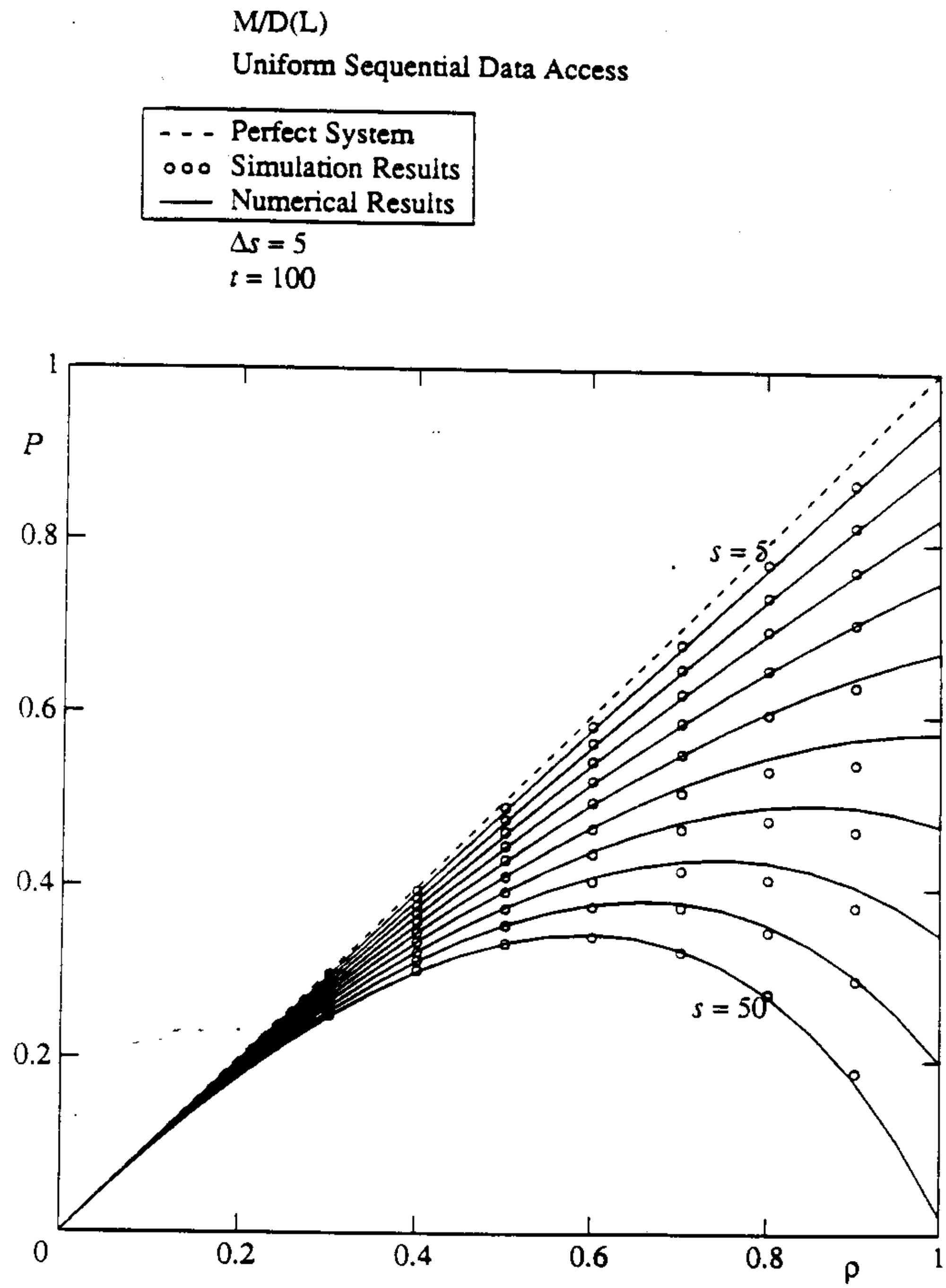


Figure 5.16: Power for Locking/Deterministic/Sequential

of overhead caused by the concurrency control schemes. If the overhead of locking is higher than the overhead for the broadcast scheme, then broadcast will give better results for certain ranges of the system load. The same can be said for the silent scheme if it were to have the lowest overhead. The schemes described in Section 5.2 have highest overhead in the locking scheme and lowest in the silent scheme.

5.5 Conclusion

Concurrency control must be enforced in order to preserve consistency of a database. In this chapter we viewed a database system as a partial-conflict system with independently shared resources. The results for the partial ISR systems, in turn, were obtained through mapping from the results obtained for the full-conflict systems.

The mapping showed for both random and sequential data access that the results are very close to the simulation results for the silent and broadcast schemes when the service time distribution is memoryless. Even the half-deterministic service time distribution gives results close to the simulation results for the broadcast case. However, pure deterministic service times differ considerably for both silent and broadcast, but only for high load.

For the locking scheme the situation is reversed. Only deterministic service times give results close to the simulation results, while half-deterministic and memoryless clearly differ from the simulation for high load.

CHAPTER 6

Conclusion

The motivation to the research described in this dissertation was to find a simple and different approach to modeling the performance of concurrency control in databases. In order to achieve that, we considered resource sharing systems with customers accessing one or more of the system resources. We defined and described those systems giving the analysis of the special case where customers access in a way that any two concurrent customers always conflict (full conflict systems). The results obtained were then mapped to realistic cases where concurrent customers not necessarily conflict (partial conflict systems). In some cases the mapping was exact, and in other it was approximate and gave results which in certain domains closely matched those obtained through simulation.

The response time and power, defined as system load divided by response time, were found for six different optimistic concurrency control scheme models, silent-redraw, silent-noredraw, silent/broadcast-redraw, silent/broadcast-noredraw, broadcast-redraw, and broadcast-noredraw, as well as for locking, a pessimistic concurrency control scheme.

The models considered have infinite number of servers. The service times of customers were modeled as consisting of a deterministic part and an exponential part. This type of service time distribution includes pure deterministic and pure exponential service times as special cases.

In this chapter we give review of the results. Then we give pros and cons of the approach described in the dissertation, and finally give possible directions for further research.

6.1 Review of the Results

Figure 3.85 gives a review of the results found for ISR systems with different characteristics. Each row in Figure 3.85 represents a different model for the conflict resolution scheme. Columns represent different service time distributions: memoryless, D_qM, for $0 < q < 1$, and deterministic. Resource demand patterns are specified in the lower left corner of the table, in a third dimension. R and S stand for random and sequential resource demands, respectively. As the table shows, some results are calculated numerically through formulas for transition probabilities, some results are found numerically through an integral expression for response time, and other results are analytic.

In both random and sequential access the results are very close to the simulation results for the silent and broadcast schemes when the service time distribution is memoryless. Even the half-deterministic service time distribution gives results close to the simulation results for the broadcast case. However, pure deterministic service times differ considerably for both silent and broadcast, but only for $\rho > 0.6$.

For the locking scheme the situation is reversed. Only deterministic service times give results close to the simulation results, while half-deterministic and memoryless clearly differ from the simulation for $\rho > 0.7$.

6.2 Pros and Cons of the Approach

Table 6.1 contains pros and cons of the approach to modeling concurrency control given in this dissertation. The pros are as follows. The approach separates the issues of queueing from the issues of overlapping data access patterns of transactions. It simplifies the view of concurrency control issues, narrowing it down to a single value of conflict measure. The separation of the queueing and data access issues leaves room for applying different measures of conflict in the future, as well as easy introduction of other data access patterns. The D_qM distribution of the transaction service times gives possibilities of simplified modeling of the concurrency control overhead. The systems with redraw allow modeling of random delays in transaction restarts. The partial restarts allow for simplified modeling of useful work and restart overhead.

The ISR systems were considered in the full conflict case. Intuitively, we expect that systems performing better in the full conflict case will probably do so in the partial conflict case as well. This can be used to compare realistic partial conflict systems by comparing their full-conflict counterparts, which are much simpler to analyze.

The cons of the approach are as follows. The error of mapping is high for

Pros of the Approach	
1	Separation of queueing from data access.
2	Simplicity of the approach.
3	Possibility of different conflict measures.
4	Easy introduction of other data access patterns.
5	Simple modeling of finite servers. [1]
6	Simple modeling of CC overhead with D_qM distribution.
7	Simple modeling of random delays with redraw systems.
8	Simple modeling of useful work with partial restarts.
9	Simple modeling of restart overhead with partial restarts.
10	Simplified comparison of different systems.

Cons of the Approach	
1	The error of mapping in some cases.
2	Infinite servers only. [2]
3	Static data access only.
4	Lack of results for noredraw systems.

Table 6.1: Pros and Cons of the Approach

[1] Assuming independent access to the servers.
[2] Assuming access to the servers on the basis of availability.

some concurrency control schemes and domains of system load. The model covers only the systems with infinite servers and static data access. We have no results for noredraw except for deterministic service times in which case redraw and noredraw are equivalent.

6.3 Further Research

The issues of further research are grouped in two areas. The first covers the cons of the approach. This would include looking for a more accurate mapping and/or a more suitable measure of conflict, obtaining results for the noredraw systems, and considering both finite servers and dynamic data access.

The other area of further research would use the pros of the approach. This means to consider finite resources, assuming independent access to servers, to introduce other data access patterns, to actually model concurrency control overhead with the D_qM distribution and random delays with redraw systems, as well as useful work and restart overhead with partial restarts. The modeling of other data access patterns would include variable transaction sizes and non-uniform record access.

An investigation on the behavior of the conflict measure should be carried out. One of the data access pattern of interest would be hybrid random/sequential access, in which the data set of size s is divided into h groups. The data items in each group are sequential, while the groups themselves are randomly scattered over the database.

Aside from the approach, it may be possible to find closed forms for some of the results for the winner queues, and to take a closer look at the departure processes in winner queues, as specified in Table 2 and suggested in Section 2.

Bibliography

[1] Rakesh Agrawal, Michael J. Carey, and Miron Livny. *"Models for Studying Concurrency Control Performance: Alternatives and Implications"*. SIGMOD Record, Vol. 14, No. 4, pp. 108-121, December 1985.

[2] Rakesh Agrawal, Michael J. Carey, and Lawrence W. McVoy. *"Performance Study of Two Phase Locking in Single-Site Database Systems"*. IEEE Transactions on Software Engineering, Vol. SE-13, No. 12, pp. 1348-1363, December 1987.

[3] Divyakant Agrawal, Arthur J. Bernstein, Pankaj Gupta, and Soumitra Sengupta. *"Distributed Optimistic Concurrency Control with Reduced Rollback"*. Distributed Computing 2, pp. 45-59, 1987.

[4] Francois Baccelli, C. A. Courcoubetis, and M. I. Reiman. *"Construction of the Stationary Regime of Queues with Locking"*. Unite de Recherche INRIA-Rocquencourt, Rapports de Recherche, No. 640, March 1987.

[5] D. Z. Badal. *"Correctness of Concurrency Control and Implications in Distributed Databases"*. Proceedings 1979 IEEE COMPSAC, Chicago, November 1979.

[6] P. A. Bernstein and N. Goodman. *"Concurrency Control in Distributed Database Systems"*. ACM Computing Surveys, Vol. 13, No. 2, pp. 185-221, June 1981.

[7] Michael J. Carey and Michael R. Stonebraker. *"The Performance of Concurrency Control Algorithms for Database Management Systems"*. Proceedings VLDB, Singapore, pp. 107-118, August 1984.

[8] M. A. Casanova. *"The Concurrency Control Problem for Database Systems"*. Springer Verlag, 1981.

[9] A. Chesnais, E. Gelenbe, and I. Mitrani. *"On the Modeling of Parallel Access to Shared Data"*. Communications of the ACM, Vol. 26, No. 3, pp. 196-202, March 1983.

[10] C. J. Date. *"An Introduction to Database Systems"*. Vol. II, Addison-Wesley, 1983.

[11] K. P. Eswaran, J. N. Gray, R. A. Lorie, and I. L. Traiger. *"The Notions of Consistency and Predicate Locks in a Database System"*. Communications of the ACM, Vol. 19, No. 11, pp. 624-633, November 1976.

[12] Leonard Kleinrock. *Queueing Systems Volume I: Theory.*, John Wiley & Sons, 1975.

[13] Leonard Kleinrock. *Analysis of the Dead/Alive Model for Optimistic Concurrency Control.* UCLA, Computer Science Department, Office of L.K., 1986.

[14] Leonard Kleinrock, Christopher Ferguson. *Simulation of "Dead/Alive" System.* UCLA, Computer Science Department, Office of L.K., 1986.

[15] Leonard Kleinrock and Farid Mehović. *Broadcast vs. Locking.* UCLA, Computer Science Department, Technical Report, 1989.

[16] H. T. Kung and John T. Robinson. *"On Optimistic Methods for Concurrency Control"*. ACM Transactions on Database Systems, Vol. 6, No. 2, pp. 213-226, June 1981.

[17] Pei-Jyun Leu and Bharat Bhargava. *"Multidimensional Timestamp Protocols for Concurrency Control"*. IEEE Transactions on Software Engineering, Vol. SE-13, No. 12, pp. 1238-1253, December 1987.

[18] Daniel Menascé and Tatuo Nakanishi. *Optimistic versus Pessimistic Concurrency Control Mechanisms is Database Management Systems.* Information Systems, Vol.7, No.1, pp. 13-27, 1982.

[19] R. J. T. Morris and W. S. Wong. *"Performance of Concurrency Control Algorithms with Nonexclusive Access"*. Performance '84, pp. 87-101, 1984.

[20] R. J. T. Morris and W. S. Wong. *"Performance Analysis of Locking and Optimistic Concurrency Control Algorithms"*. Performance Evaluation 5, pp. 105-118, 1985.

[21] Arturo Andrea Pizano. *"Response Time Approximation in a Distributed Database System"*. Master Theses, UCLA Computer Science Department, 1985.

[22] K. H. Pun and Geneva G. Belford. *"Performance Study of Two Phase Locking in Single-Site Database Systems"*. IEEE Transactions on Software Engineering, Vol. SE-13, No. 12, pp. 1311-1328, December 1987.

[23] In Kyung Ryu. *"Performance Evaluation of Concurrency Control in Database Systems"*. PhD Dissertation, USC Electrical Engineering Department, September 1985.

[24] In Kyung Ryu and Alexander Thomasian. *"Performance Analysis of Data Contention with Dynamic Locking"*. IBM Research Report, RC 11399 (#51319) Computer Science, 31 pages, September 1985.

[25] In Kyung Ryu and Alexander Thomasian. *"Performance Analysis of Centralized Databases with Optimistic Concurrency Control"*. IBM Research Report, RC 11420 (#51353) Computer Science, 36 pages, October 1985.

[26] In Kyung Ryu and Alexander Thomasian. *"Performance Analysis of Dynamic Locking with the No-Waiting Policy"*. IBM Research Report, RC 11929 (#53738) Computer Science, 37 pages, June 1986.

[27] In Kyung Ryu and Alexander Thomasian. *"Performance Analysis of Dynamic Locking"*. Proceedings ACM-IEEE 1986 Fall Joint Computer Conference, Dallas, Texas, November 1986.

[28] Kenneth C. Sevcik. *"Comparison of Concurrency Control Methods Using Analytic Models"*. Information Processing 83, pp. 847-858, 1983.

[29] Y. C. Tay and Rajan Suri. *"Choice and Performance in Locking for Databases"*. Proceedings VLDB, Singapore, pp. 119-128, August 1984.

[30] Y. C. Tay, Nathan Goodman, and Rajan Suri. *"Integrated Concurrency Control and Recovery Mechanisms: Design and Performance Evaluation"*. ACM Transactions on Database Systems, Vol. 10, No. 4, pp. 529-564, December 1985.

[31] Y. C. Tay, Rajan Suri, and Nathan Goodman. *"A Mean Value Performance Model for Locking in Databases: The No-Waiting Case"*. Journal of the ACM, Vol. 32, No. 3, pp. 618-651, July 1985.

[32] Y. C. Tay, Nathan Goodman, and Rajan Suri. *"Locking Performance in Centralized Databases"*. ACM Transactions on Database Systems, Vol. 10, No. 4, pp. 415-651, December 1985.

[33] Alexander Thomasian and In Kyung Ryu. *"A Decomposition Solution to the Queueing Network Model of the Centralized DBMS with Static Locking"*. Proceedings 1983 SIGMETRICS Conference, Minneapolis, MN, August 1983.

[34] Alexander Thomasian and In Kyung Ryu. *"Analysis of Some Optimistic Concurrency Control Schemes Based on Certification"*. ACM Sigmetrics Conference, Austin, Texas, August 1985.

[35] Alexander Thomasian and In Kyung Ryu. *"On Analyzing Lock Contention in Databases"*. IBM Research Report, RC 12666 (#56814) Computer Science, 25 pages, March 1987.

[36] John N. Tsitsiklis, Christos H. Papadimitriou, and Pierre Humblet. *"The Performance of a Precedence-Based Queuing Discipline"*. Journal of the ACM, Vol. 33, No. 3, pp. 593-602, July 1986.

[37] J. Ullman. *"Principles of Database Systems"*. Second Edition, Computer Science Press, Rockville, Maryland, 1983.