This dissertation has been
microfilmed exactly as received

70-4199

SOLBERG, James Joseph, 1942-
A GRAPH THEORETIC APPROACH TO THE STUDY OF NETWORKS OF QUEUES.

The University of Michigan, Ph.D., 1969
Engineering, industrial

University Microfilms, Inc., Ann Arbor, Michigan

A GRAPH THEORETIC APPROACH TO THE STUDY OF NETWORKS OF QUEUES

by

James Joseph Solberg

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy in
The University of Michigan
1969

Doctoral Committee:

Professor Ralph L. Disney, Chairman
Associate Professor Seth Bonder
Professor Keki B. Irani
Professor Richard C. Wilson

ABSTRACT

A GRAPH THEORETIC APPROACH TO
THE STUDY OF NETWORKS OF QUEUES

by

James Joseph Solberg

Chairman: Ralph L. Disney

This paper presents a new approach to determining the steady-state probability solution to those queueing networks which can be described as finite, irreducible, discrete-state, continuous-parameter Markov processes, and demonstrates the significance of the approach by means of two examples. The approach relies upon a graphical technique derived from the flowgraph method of Coates for solving linear equations. It is shown that (provided the equations are appropriately formulated) the Coates' flowgraph of the steady-state equations of a queueing network is essentially equivalent to the transition diagram of the process. Subsequent theorems refine Coates' method to take advantage of the special properties of queueing systems. Two practical computational procedures are provided, the complete method is summarized, and a simple example is given to illustrate the method.

In addition to providing computational algorithms, the graphical approach provides theoretical tools.

The use of these theoretical tools is illustrated in two ways. First, a number of ways of decomposing a transition diagram into smaller, more manageable pieces are derived. Secondly, a number of approximation theorems are proved. The latter permit one (for queueing networks of a particular class, subject to certain traffic conditions) to determine an approximate steady-state solution from inspection of the transition diagram.

As specific examples of queueing networks, the two-server instantaneous jockey system and the two-server overflow system are considered in detail.

ACKNOWLEDGEMENTS

I cannot adequately express my gratitude to Professor R. L. Disney for the help he has provided me. Over the years since he first introduced me to the subject of queueing theory, he has been consistently generous with his time, information, and advice. I of course wish to thank the members of my committee for their most co-operative efforts. I also owe a debt of gratitude to Professor F. Harary for an inspiring education in graph theory. Finally, my wife is to be congratulated for her stalwart patience and good cheer.

TABLE OF CONTENTS

	Page
ACKNOWLEDGEMENTS	ii
LIST OF ILLUSTRATIONS	iv
PART I. PRELIMINARIES	
Chapter 1. Introduction	2
Chapter 2. Algebraic Formulation	14
Chapter 3. Transition Diagrams	24
Chapter 4. Graph Theory	34
PART II. THEORY	
Chapter 5. Solution Techniques	53
Chapter 6. Decomposition	78
Chapter 7. Approximation	90
Appendix to Chapter 7	110
PART III. APPLICATIONS	
Chapter 8. The Instantaneous Jockey System	125
Chapter 9. The Overflow System	139
PART IV. CONCLUSION	
Chapter 10. Some Informal Remarks	168
BIBLIOGRAPHY	171

LIST OF ILLUSTRATIONS

Figure		Page
1-1.	A typical queueing network	3
1-2.	Two servers in parallel	5
1-3.	Two servers in series	6
2-1.	A specific pattern	15
3-1.	The shorter queue transition diagram	27
3-2.	The instantaneous jockey transition diagram	28
3-3.	The overflow transition diagram	29
3-4.	The two-servers-in-series transition diagram	30
4-1.	A typical digraph	35
4-2.	A spanning subgraph of D	36
4-3.	A subgraph of D	37
4-4.	Another subgraph	37
4-5.	D-S	37
4-6.	S+T	38
4-7.	Adjacency	39
4-8.	A digraph with two components	39
4-9.	A path	40
4-10.	A cycle	40
4-11.	A non-path	40
4-12.	A non-path	40
4-13.	A 1-factor	41

4-14. A 1-factor connection..... 41
4-15. Two labeled digraphs..... 42
4-16. Two isomorphic weighted labeled digraphs..... 43
4-17. The flowgraph of A..... 44
4-18. The flowgraph of Ax=b..... 45
4-19. Another flowgraph of the same equations..... 46
4-20. The denominator terms..... 48
4-21. The numerator terms..... 49
5-1. Before and after modification..... 57
5-2. The proof of theorem 5.4..... 72
5-3. Modified transition diagram..... 76
5-4. G_0-P_{0_i} 76
5-5. One balance step..... 77
5-6. Many balance steps..... 77
6-1. A digraph G..... 80
6-2. The strongly connected components..... 80
6-3. Determinant decomposition at a cutpoint..... 85
7-1. The difference between H^k and H^{k+1}..... 92
7-2. The graph S..... 95
7-3. H^0..... 97
7-4. All H^1..... 97
7A-1. The difference between H^k and H^{k+1}..... 110
7A-2. The difference between H^k and H^{k+1}..... 112
7A-3. A graph S..... 113

7A-4. H^0 114
7A-5. The two H^1 114
7A-6. Theorem 7.4, case 1 121
7A-7. Theorem 7.4, case 2 122
8-1. The instantaneous jockey system 126
8-2. The modified transition diagram 128
8-3. The graphs A_i and B_i 129
8-4. The graph C when i<M 130
8-5. The graph C when i=M 131
8-6. The graphs S_i 132
8-7. The graph D when i>0 134
8-8. The graph E when i>0 134
8-9. The graph D when i=0 135
8-10. The graph E when i=0 135
9-1. The overflow system 140
9-2. The overflow transition diagram 142
9-3. The four column types 144
9-4. Balancing A_m 145
9-5. Separating the bottom point 146
9-6. Transition diagram when N=1 149
9-7. The remaining subgraph decomposed 152
9-8. Transition diagram when N=2 154
9-9. The set of paths 155
9-10. Paths having value ρ^{M+2} 156

9-11. Paths having higher values..... 156
9-12. The path and remaining subgraph..... 158
9-13. Cycles between columns..... 158
9-14. The path and remaining subgraph..... 159
9-15. The other cycle..... 159
9-16. The path and remaining subgraph..... 160
9-17. Some of the correction terms..... 161
9-18. The rest of the correction terms..... 162

PART I

PRELIMINARIES

CHAPTER 1

INTRODUCTION

Despite the rich variety of available knowledge pertaining to individual queues, to say almost anything about networks of queues has proved to be a difficult task. The absence of even an accepted approach to the problem bears testimony to its inherent difficulties, for it has existed as a recognized problem for as long as queueing theory itself has existed.

This paper will present a new approach to determining the steady-state probability solution to those queueing networks which can be described as finite, irreducible, discrete-state, continuous-parameter Markov processes, and will demonstrate the significance of the approach by means of examples. The approach is easy to describe and to apply, yet it is sufficiently powerful to provide information about systems which have not been solved in any other way.

The Problem In General

Consider, as an example of a queueing network, the system depicted in figure 1-1. Squares represent servers; the long troughs, waiting space; arrows, the traffic flow; and the circle, a decision point. In this case, an arrival to the system enters from the left. At the circle, a

decision is made by the item to be serviced (if the item is a human customer), by the system itself, or by external management intervention. As a result of the decision, the item enters the top subsystem, to be serviced in turn by servers A and B—subject of course to their respective queue disciplines—, or it enters the bottom subsystem, to be serviced by C. At the completion of the item's servicing, it departs from the system to the right.

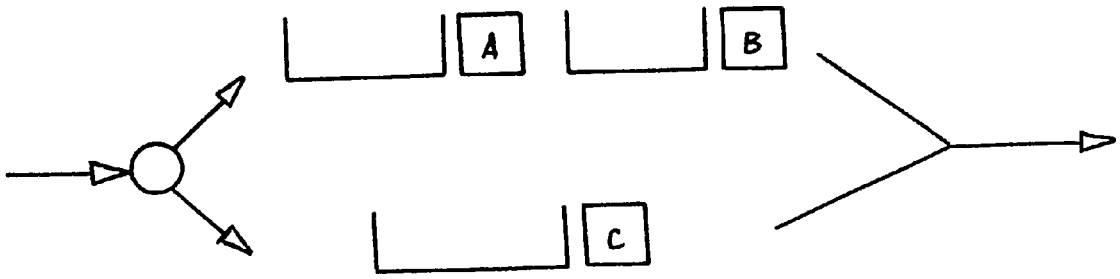


Fig. 1-1. A typical queueing network.

Some previous attempts to solve network problems have been based on the concept of studying the constituent parts separately. Thus, the decision point, which can be viewed as simply a switching device when the decision criterion is known, acts upon an arrival stream to divide it into two separate arrival streams. Server A and server C, along with their respective waiting spaces, are each just simple queues whose arrival stream emanated from the decision point. Server B, along with its waiting space, is just a simple queue whose arrivals are the output of server A.

Although this approach to the problem has some appeal, in that the individual queues are "smaller" than the network, the conceptual separation of the parts of a network may create problems which (1) are very difficult to solve, and (2) did not exist in the original network. In particular, even when the network as a whole is Markovian (with respect to some suitable defined "state-of-the-system" variable), the subsystem will not, in general, be Markovian. Speaking generally, what happens in one part of a network may depend upon what has happened in another part. The state-of-the-system variable takes these dependencies into account automatically. To extract a subsystem from the network for separate study leaves one in the position of either ignoring the dependencies or finding ways to provide for them mathematically. Experience has shown the latter to be very difficult.

In contrast to this entire approach, the approach to be taken within this paper relies upon preserving the integrity of the network. Specifically, the approach relies upon a new graphical technique (related, but not identical, to Coates' flowgraph method for solving linear equations) for obtaining the steady-state distribution of the network. Conveniently, the graph used for this purpose happens to be the transition diagram of the system, a graph which is already well established in queueing literature.

The great appeal of a graph theoretic approach lies in its intuitive value. An abstract graph, whether drawn

or only imagined, enables one to visualize concepts and relationships which might otherwise be too complicated to grasp. The method to be developed in this paper is particularly appealing because the graph is a "natural" one.

Accordingly, the significance of this new approach is not so much that it enables one to solve for the steady state probabilities—this, at least in principle, was possible before—but that it enables one to study networks of queues in the steady state. It provides theoretical tools as well as computational algorithms.

Some Example Systems

For the sake of having specific examples at hand from the outset, consider a few particular systems. Figure 1-2 depicts what will be called the two-server shorter queue system. The name derives from the way in which the switching device assigns items to server lines, namely, always to the shorter line. If both queues are of the same length, the item is sent to either with equal probability.

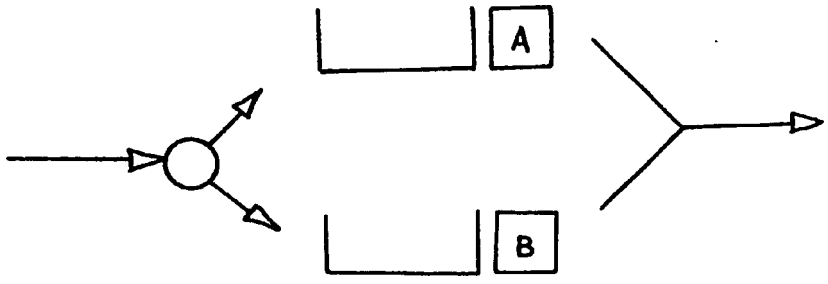


Fig. 1-2. Two servers in parallel.

A variation of this system is called the two-server instantaneous jockey system. In all respects it behaves the same, except that whenever the lines differ in length by more than one, the item at the rear of the longer line "jockeys" instantaneously to the shorter queue. The system itself looks like figure 1-2.

A different kind of switching discipline characterizes the two-server overflow system. In this case, an arrival is assigned to the first server unless his waiting space is fully occupied, in which case it "overflows" to the second server. Again figure 1-2 depicts the system.

Each of these three systems are of a class to be called parallel systems. They are characterized by the property that the cause of every change of state is either an arrival to or a departure from the system. This would rule out, for example, the two-servers in series system shown in figure 1-3. Since an item upon completing service at A enters the queue of B, the system can change state without the occurrence of either an arrival or a departure.

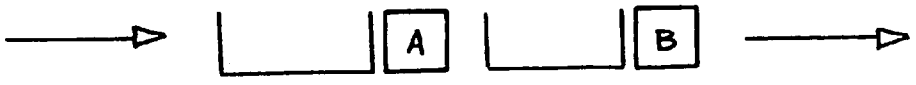


Fig. 1-3. Two servers in series.

Under the assumptions that the waiting space is finite, that the overall arrival process is Poisson, and that the service times are negative exponentially distributed, all of these systems can be solved by the methods to be developed herein. As examples, the instantaneous jockey and the overflow systems will be studied in particular detail. They are, of course, only representative of the variety of systems to which the methods will apply.

Historical Background

In discussing historical precedents for the work of this paper, it is important to distinguish between specific results obtained for particular network types (using techniques which usually cannot be generalized) and general approaches to the problems of queueing networks. Of the former, there are many. T. L. Saaty has summarized them well in his book [25] and in an address to a symposium on congestion theory held in 1964 [26].

Of general approaches taken, there are really only two. One of these, summarized by Disney in [11], is based upon the visualization of a network of queues as a combination of separate queues. But this point of view, for one thing, has not been particularly fruitful, for as Saaty concludes, "...the general problem of a network of queues remains in its infancy without substantial changes." [26, p. 86]. For another, it does not permit one to grapple with the real problems of interest, such as dependencies

within the network and the question of alternative network structures.

The other approach has received little attention until recently. R. Syski, in 1961, was apparently the first to propose seriously that a general queueing network be viewed as an integrated whole [29]. Syski's method was to define a stochastic process on a "pattern space", much as will be done in chapter 2 of this paper. His results are of theoretical interest, but are too formal to be very useful. Also regarding the problem as a whole, V. Benes has recently obtained [1] many genuinely significant results for a certain class of networks (basically, telephone switching networks) which could conceivably be emulated in a broader context.

One of the great barriers to the whole-network approach has been the absence of a good way to represent a network, a way which is both mathematically and intuitively convenient. Pointing out that substantial theory relevant to deterministic network flow exists within the mathematical discipline of graph theory, Saaty, in [26], proposed visualizing a queueing network as a directed graph (i.e., a collection of points, for servers, and directed lines between points, to indicate traffic flow) and utilizing concepts of graph theory, somehow "introducing the stochastic element." Saaty's proposed directed graphs would look like simplified versions of figures 1-1, 1-2, or 1-3.

However, the remarks of Richard Evans following Saaty's address [26, p. 102-105] seem more to the point:

> "Thinking of a network of queues as a network of expanding or collapsing pipes as suggested by Dr. Saaty has severe limitations. The customers, jobs, or what have you, which flow in a queueing network often are not homogeneous as suggested in fluid flow analogy."

But after discussing some of the other disadvantages and inadequacies of Saaty's proposed approach, Evans goes on to suggest:

> "A more complete description which does result in a graph may be obtained if one defines a state of the system variable and then graphs the single event transitions which are possible. At the moment this mode of system definition seems to be the most promising one..."

It is the approach suggested by Evans which is taken in this paper. Although graph theory concepts are utilized extensively, the focus is upon the graph of the state space of a network (i.e., the transition diagram) rather than a graph of the network itself.

Coates' flowgraph method for solving linear equations [5], upon which the method to be presented herein is based, is itself based upon the signal flow graph method of S. J. Mason [22]. Mason's work first appeared in 1953. Since that time, it has become a generally known and accepted technique for solving linear equations of the form that occur in electrical engineering (generally, equations of the form $Ax = x$).

Coates' theorem did not appear until 1959. Between 1953 and 1959, a number of papers noted the possibility

of applying signal flow graphs to stochastic systems. (See [18], [27], and [17].) Discrete Markov processes, in particular, provided a natural area of application of the technique, because the equations were of the required form. Continuous parameter Markov processes, on the other hand, did not appear to provide a convenient area of application, because Laplace transforms were required to bring the equations into the proper form.

It was Coates' clear intention, at the time he published his flowgraph theorem, to provide a simple variation of Mason's signal flow graph theorem which would improve computational efficiency. Neither he nor later writers (e.g., [8], [16]) show any awareness of the possibility that Coates' method might apply very naturally in areas where Mason's does not. Lorens, for example, dismisses Coates' method in his 1964 book with the remarks:

> "This method involves subtracting the variable from both sides of the [signal] flowgraph equation. The rules are [signal] flowgraph rules with a unity self-loop subtracted from each variable. The problem with their use is principally one of formulation." [21, p. 170]

Recognizing that the problem of solving linear equations is a mathematical problem—not just a problem encountered in electrical engineering—Wai-Kai Chen, in several more recent papers ([2], [3], and [4], among others), has elevated Coates' method to a position equal to that of Mason's. One article in particular, [3], is recommended as a summary of existing results.

As a result of these historical circumstances, it has gone unnoticed until now that Coates' method provides a very natural and convenient way to solve those linear equations which occur in connection with a continuous parameter Markov process. It was the discovery (or, more properly, observation) of this fact which led to the development of the method for studying networks of queues presented in this paper.

Organization Of The Paper

The remaining three chapters of Part I are merely preliminary to the rest of the dissertation; they contain no new results. Chapter 2 establishes the elements of the algebra of finite, irreducible, discrete-state, continuous-parameter Markov processes. Chapter 3 describes the transition diagram and discusses its known usefulness. Chapter 4 contains definitions and elementary concepts of the formal theory of directed graphs. Coates' theorem for solving linear equations by means of a flowgraph is stated and illustrated. A variation of the theorem, more suited to the needs of this paper, is also given.

The three chapters of Part II are theoretical in nature. With the exception of the first theorem in chapter 6, the results contained therein are original.

Chapter 5 contains the complete development of the solution technique. The fundamental basis for the method is revealed in theorem 5.1, the equivalence theorem. This

theorem states that, provided the equations are properly formulated, the Coates' flowgraph of the steady-state equations of a queueing network is essentially equivalent to the transition diagram of the process. This result permits the transition diagram to be used as a computational device. The remainder of chapter 5 is devoted to refining the flowgraph technique to take advantage of the special properties of those flowgraphs which originate from queueing systems. Two practical computational procedures are provided, the complete method is summarized, and an example is given to illustrate the use of the method.

Chapter 6 illustrates in one way the power of the graphic approach. Queueing networks of even moderate size or complexity are likely to produce very large sets of equations or, equivalently, very large transition diagrams. As a practical matter, such graphs must either be collapsed or decomposed in some fashion. The theorems of chapter 6 involve several kinds of decomposition.

Chapter 7 demonstrates in another way the usefulness of the graphic concepts in advancing the mathematical theory. For networks of the parallel system class, subject to certain traffic conditions, approximate solutions can be determined immediately from inspection of the transition diagram. That is, no computation is involved. The theorems which state how this is done are proved with the aid of graph theoretic concepts.

In Part III, the solution technique of chapter 5, as well as the theorems of chapters 6 and 7, are applied to the two-server instantaneous jockey system, in chapter 8, and the two-server overflow system, in chapter 9. The former solution has appeared before [20], and is therefore verifiable. The solution to the overflow system is new.

The final chapter, chapter 10, lists certain problems which, though uninvestigated, would very likely yield to any serious attempt. These include suggested generalizations and extensions of the theorems in chapters 5, 6, and 7, as well as particular queueing systems whose transition diagrams seem appealingly simple.

CHAPTER 2

ALGEBRAIC FORMULATION

In this chapter, the specific mathematical assumptions of the problem are stated and the algebraic equations which determine the steady state probabilities are formulated. For the most part, the development proceeds in the conventional manner, but there is one noteworthy exception. Ordinarily, the states of a queueing process are identified as integers. This is done for purposes of algebraic convenience—to permit the use of matrix notation, for example—but it is not really necessary. Because the method which will be introduced and used in this paper is not algebraic, but rather graphical, the identification of states by integers is both unnecessary and, usually, inconvenient. Accordingly, the first section of this chapter will discuss an alternative, more general, means of state designation.

States of the System

The discrete state, continuous time Markov processes used to describe the state of single queue systems are invariably defined on the integers. That is, the value of the random variable $X(t)$ at a fixed time t is given by an integer, interpretable as the queue length, or number of

busy servers, or some such thing. A realization of the process is mathematically viewed as a sequence of integers. This is certainly a very natural and convenient way to represent the processes for such simple systems. It is somewhat less natural and convenient when one begins to consider more complicated systems and behavior.

Consider, for example, the case of two servers in series, the output of the first becoming the input to the second. An adequate description of the state of the system requires at least a specification of the number of items waiting for the first server and the number waiting for the second server. It is not immediately obvious how to establish the correspondence between states of this kind and the integers. In this particular case, it is probably most natural to describe the state by a two-dimensional vector. In other cases, other state representations may suggest themselves.

Rather than become involved in the various means by which the states of a process may be designated, let us broaden our notion of discrete parameter stochastic processes in general along the lines suggested by R. Syski in 1961 [29]. Assuming that a physical system, which a stochastic process is to model, is given, the states of the process will correspond to the various configurations of items in the physical system. For each genuinely realizable distinct configuration of the system, let there be defined an abstract "pattern."

Take, for example, the system depicted in figure 1-1. One of the patterns of this system might be designated by the drawing shown in figure 2-1.

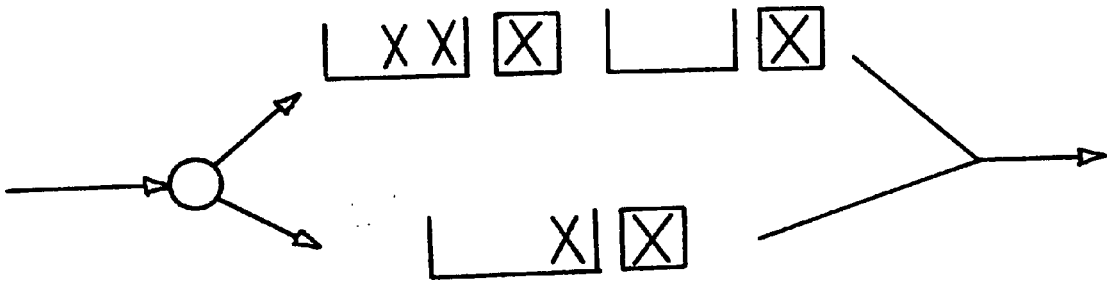


Fig. 2-1. A specific pattern.

This would represent the state describable in words as "three items to be serviced by A, one by B, two by C." Alternatively, the same pattern might be designated by the integer triple (3,1,2), or by the arbitrary integer 9, or in any number of other perfectly reasonable ways.

Each of these designations, however, would refer to the same underlying configuration of items in the system. The notion of patterns is introduced to raise the level of abstraction above any particular state designation convention. The set of all patterns, the "pattern space", is thus just an abstract set of mathematical entities which correspond exactly to the possible states of the system.

Lost in the process of abstraction, of course, is the linear ordering of states suggested by the integers. As already mentioned, however, such an ordering would be less than natural for all but the simplest processes. The natural ordering of patterns will be discussed in the next chapter.

In all other respects, the pattern-valued Markov process can be treated exactly as an ordinary discrete state one. In order to take full advantage of this fact, whenever it is convenient to do so, let us choose to designate the states by numbering the states arbitrarily. In doing so, we recognize the action to be convenient but unnecessary.

The Assumptions

Primarily for the reason that the graphical techniques to be used later will require finite graphs, it will be explicitly assumed here that the processes to be dealt with possess only a finite number of patterns (or states). Conveniently, the assumption of finiteness simplifies much of the probability theory which will be required. For example, we know in advance that a steady-state solution exists. But the assumption also influences the kinds of systems which can be studied. The number of servers must be finite, the capacity of each queue must be finite, and so forth.

All processes to be considered must satisfy the Markov property. This property is stated for pattern-valued processes just as it is for integer-valued processes. Specifically, a pattern-valued stochastic process $\{X(t)\}$ is Markovian if for all $t_1<t_2<\ldots<t_n$,

$$\Pr\{X(t_n)=p_n \mid X(t_{n-1})=p_{n-1},\ldots X(t_1)=p_1\} = \Pr\{X(t_n)=p_n \mid X(t_{n-1})=p_{n-1}\},$$

where the p_i are patterns. In words, the process is Markovian if knowledge of the present pattern is sufficient to predict, in the probabalistic sense, the next pattern.

The Markov assumption is fundamentally an assumption of independence. It is particularly important when speaking of networks of queues to understand the nature of whatever independence assumptions are being made. Consider, then the class of systems which might be modeled as Markov processes.

For all practical purposes, the arrivals must be generated by pure birth processes. There may, on the other hand, be more than one arrival process involved; and, for that matter, the parameters of the arrival processes may be state dependent. Similarly, the service times must be negative exponentially distributed, but the parameters may depend upon the state.

Actually, by an extension of the notion of patterns, it would be possible to include arrival or service distributions of the Erlang family, but this idea will not be explored further. The important restriction is that the intervals between transitions of the system as a whole (that is, changes in the pattern) be sampled from independent negative exponential distributions.

The Equations

In principle, a time-homogeneous discrete state, continuous parameter Markov process is defined by its transition

probability functions $p_{ij}(t)$, where these are defined as $p_{ij}(t) = Pr\{X(u+t)=j \mid X(u)=i\}$. Since these functions represent probabilities for every value of t, they must satisfy

$$0 < p_{ij}(t) < 1 \tag{2.1}$$

$$\sum_j p_{ij}(t) = 1 \tag{2.2}$$

$$p_{ij}(0) = 0 \text{ for } i \neq j \tag{2.3}$$

$$p_{ii}(0) = 1 \tag{2.4}$$

In addition, the Markov property implies

$$p_{ij}(t+u) = \sum_k p_{ik}(t)p_{kj}(u) \tag{2.5}$$

In applications, however, these functions are generally known initially for short time intervals through the transition rates. The transition rates are constants such that as $\Delta t \to 0$,

$$p_{ij}(\Delta t) = \lambda_{ij}\Delta t + o(\Delta t), \text{ for } i \neq j, \text{ and} \tag{2.6}$$

$$p_{ii}(\Delta t) = 1 + \lambda_{ii}\Delta t + o(\Delta t). \tag{2.7}$$

where λ_{ij} is the transition rate from i to j and $o(\Delta t)$ is some function such that $o(\Delta t)/\Delta t \to 0$ as $\Delta t \to 0$.

Assume that all of the transition rates are finite. Any state i which had an associated $\lambda_{ij} = \infty$ would be an instantaneous state. That is, the duration of occupancy of the state would be infinitesimal. Such states cause great mathemetical difficulties (see [19], p. 227), but as a practical matter, they may be ignored.

It is immediately obvious from (2.1) that

$$\lambda_{ij} \geq 0, \quad \text{for } i \neq j, \text{ and} \tag{2.8}$$

$$\lambda_{ii} \leq 0, \quad \text{for all } i. \tag{2.9}$$

As a matter of fact, it can easily be shown, using (2.3), (2.4), (2.6), (2.7), and the definition of the derivative, that, even for i=j,

$$\lambda_{ij} = \left. \frac{dp_{ij}(t)}{dt} \right|_{t=0}$$

Thus the equations (2.6) and (2.7) can be interpreted as Maclaurin's series expansions of the transition functions.

Using (2.6) and (2.7) in conjunction with (2.5), one obtains

$$p_{ij}(t+\Delta t) = \sum_{k \neq j} p_{ik}(t)\lambda_{kj}\Delta t + p_{ij}(t)\{1+\lambda_{jj}\Delta t\} + o(\Delta t)$$

which after minor manipulation becomes

$$\frac{p_{ij}(t+\Delta t) - p_{ij}(t)}{\Delta t} = \sum_{k} p_{ik}(t)\lambda_{kj} + \frac{o(\Delta t)}{\Delta t}$$

Taking the limit as $\Delta t \to 0$, one obtains the Chapman-Kolmogorov forward differential equation

$$\frac{dp_{ij}(t)}{dt} = \sum_{k} p_{ik}(t)\lambda_{kj} \text{ , for all } i,j. \tag{2.10}$$

Notice that, although the subscripts have been written as if they were integers, there is nothing in the above equations to prevent the subscripts from referring to patterns directly. It will, however, prove convenient at this point to impose an arbitrary ordering on the pattern space—that is, number the patterns—in order to utilize matrix notation.

Let the integer zero refer to the "empty pattern", the state of the system when it contains no items to be served. Number the other patterns in any order whatsoever. Let Λ_a represent the matrix of transition rates, written in the usual way with λ_{00} in the upper left and the rows and columns ordered in corresponding fashion. Similarly, let P(t) represent the matrix of transition functions. Then the set of differential equations represented by (2.10) may be written more simply as

$$\frac{dP(t)}{dt} = P(t)\Lambda_a \tag{2.11}$$

A process for which

$$\sum_j \lambda_{ij} = 0 \tag{2.12}$$

is said to be conservative. Not every process is conservative, but every finite one is [6]. Since the assumption of finiteness has already been made, the conservative property will necessarily hold for all of the processes to be considered here.

This property tells us several things about the matrix Λ_a. It tells us directly that the sum of the elements in every row, the "row sums", is zero. Since, from (2.8) and (2.9), only the terms on the diagonal may be negative, it tells us that for every i,

$$\lambda_{ii} = - \sum_{j \neq i} \lambda_{ij} \tag{2.13}$$

Finally, it tells that Λ_a is of less than full rank, because there exists a linear combination (namely, the sum) of the columns which equals the zero vector.

A process is <u>irreducible</u> if for any two states i and j, there exists a sequence of possible transitions taking the process from state i to state j and there exists another sequence of possible transitions taking the process from state j to state i. We will restrict our attention to such processes.

This property adds to our knowledge of Λ_a. We may be certain that at least one λ_{ij} in each row i is strictly greater than zero. If not, there is no possible transition out of the state i and the process cannot be irreducible. We can then further deduce, using (2.13), that every diagonal term λ_{ii} must be strictly negative. Similar logic enables us to conclude that at least one element in each column must be strictly greater than zero.

We are interested in the behavior of the transition probability functions as $t \to \infty$. For finite irreducible processes, these functions approach, asymptotically, limits which are independent of the initial state. Since in the limit the initial state is irrelevant, define

$$q_j = \lim_{t \to \infty} p_{ij}(t) \qquad (2.14)$$

The set of the q_j for all states j is called the equilibrium or steady state solution. For finite, irreducible processes, the equilibrium solution exists, is strictly positive, and constitutes a proper probability distribution. It can be shown (see [6], p. 184) that the vector $q = (q_0, q_1, \ldots q_N)$ is

determined by the set of equations

$$q\Lambda_a = 0, \text{ and} \tag{2.15}$$

$$\sum q_i = 1 \tag{2.16}$$

The system of equations (2.15) is simply a set of homogeneous linear equations. It has already been shown that the rank of Λ_a is less than full, so a non-trivial solution exists. The addition of (2.16), called the normalizing equation, determines the q_i uniquely. It may therefore be concluded that the rank of Λ_a is one less than the order.

CHAPTER 3

TRANSITION DIAGRAMS

The transition diagram of a discrete state stochastic process is a pictorial representation of the state space of the process which shows the relationships between states in terms of the possible transitions. It is a well-known and frequently used device, whose value as a conceptual aid is well established. In chapter 5, it will be shown that the transition diagram can also be a valuable computational tool. In chapters 6 and 7, further uses for the transition diagram will be obtained—uses which enhance its value as a conceptual aid.

In this chapter, the transition diagram will be defined, certain conventions to be followed in drawing the diagram will be established, and a number of examples will be given. For the sake of completeness, a section on the well-known use of the transition diagram to obtain the steady-state equations has also been included.

Definitions

Given a discrete state stochastic process, the transition diagram of the process is constructed by drawing a point to represent each state of the process, and drawing an arrow from point i to point j wherever a direct transition

from state i to state j is possible. In the case of continuous parameter Markov processes, such as we are interested in, a direct transition is possible if and only if $\lambda_{ij} \neq 0$. In such a case, it is common practice to write the value of λ_{ij} next to the arrow from i to j. Lines from a point to itself are not included in the transition diagram (even though λ_{ii} is defined), because such a line would not represent an actual transition, or change of state.

It will aid in the presentation of these diagrams to establish certain conventions regarding the placement of states in the drawings. It is to be understood that these are entirely arbitrary and have no influence on the theorems. But as long as the states must be placed on the paper in some manner, there is value in being as consistent as possible.

All of the systems to be considered have the characteristic that they may be "empty." In drawing a transition diagram, place the empty state at the bottom. In a horizontal row above this, place those states which correspond to a system which contains exactly one item. In a row above that, place all those states corresponding to a system containing two items, and so forth. These horizontal rows will be referred to as <u>levels</u> and for purposes of reference will be numbered such that the k-th level contains those states corresponding to k items in the system. Thus the empty state is in the zero-th level.

Three kinds of transitions may occur in a service system: a) an <u>arrival</u>, which increases the number in the system by one,

b) a departure, which reduces the number by one, and c) an internal transition, which leaves the number unchanged. In terms of the transition diagram, an arrival is represented by an arrow going generally upward to the next higher level; a departure, by an arrow going generally downward to the next lower level; and an internal transition, by an arrow going generally horizontally to a state in the same level.

Examples

Assuming these conventions, let us now look at a few examples of two-server systems. Next to each point will be written an integer pair, (i,j), in which i refers to the number of items in the queue of the first server and j refers to the number in the queue of the second. These state designations are included at this time only so that the points may be identified with their corresponding states in the physical system; in subsequent chapters they will be deleted. In addition, the arrows are not labeled with their associated transition rates, as they ordinarily would be, in order that the structure of the diagrams may be emphasized.

Figure 3-1 is the transition diagram of a two-server shorter queue system in which each server has waiting space for four items, including the one in service. The next figure, figure 3-2, shows the transition diagram of an instantaneous jockey system with storage capacity the same as in the previous example. An overflow system in which the primary server has capacity for three items and the overflow server has a capacity for four is depicted in figure 3-3.

Finally, the two-servers-in-series case with three and four item capacities for the first and second servers respectively is shown in figure 3-4.

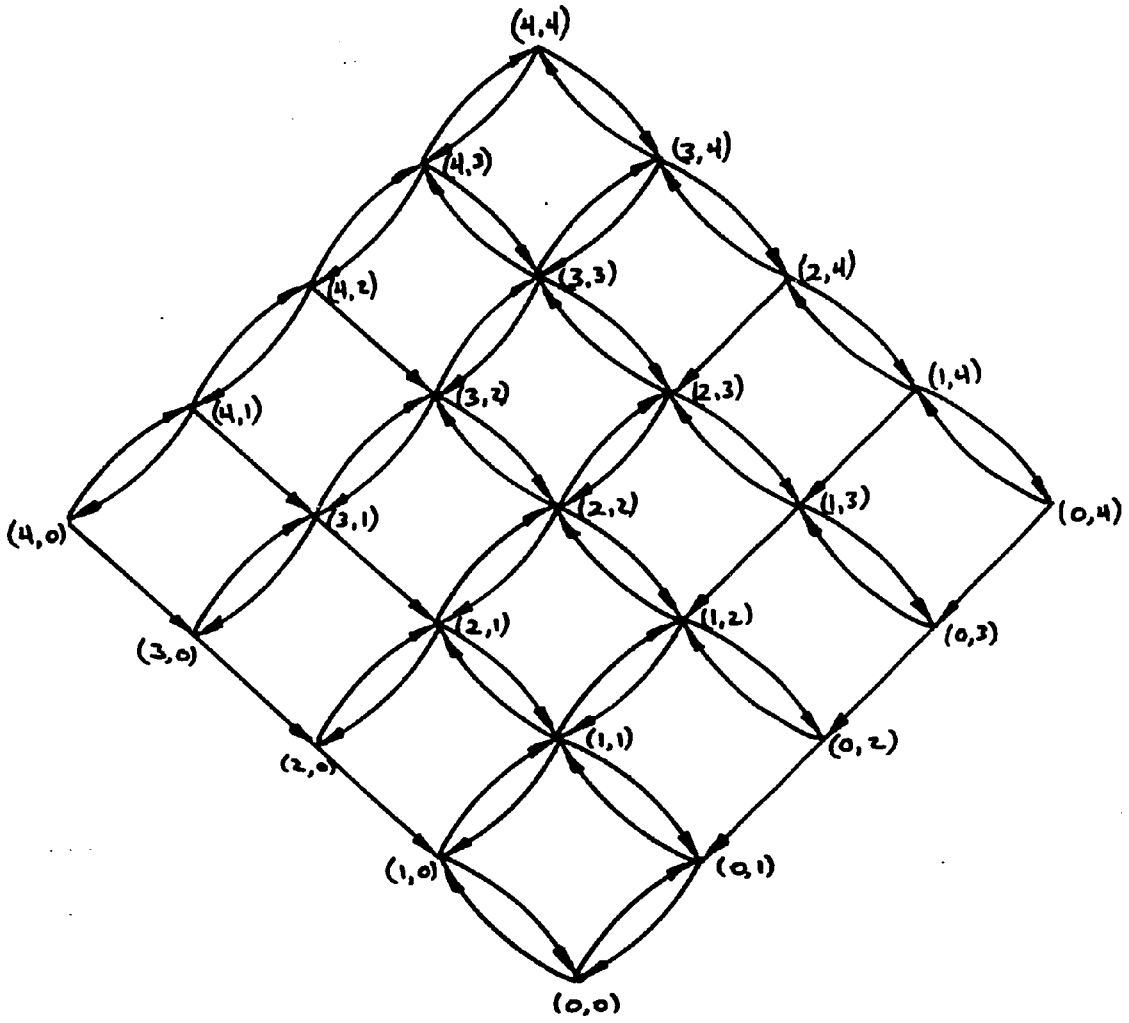


Fig. 3-1. The shorter queue transition diagram.

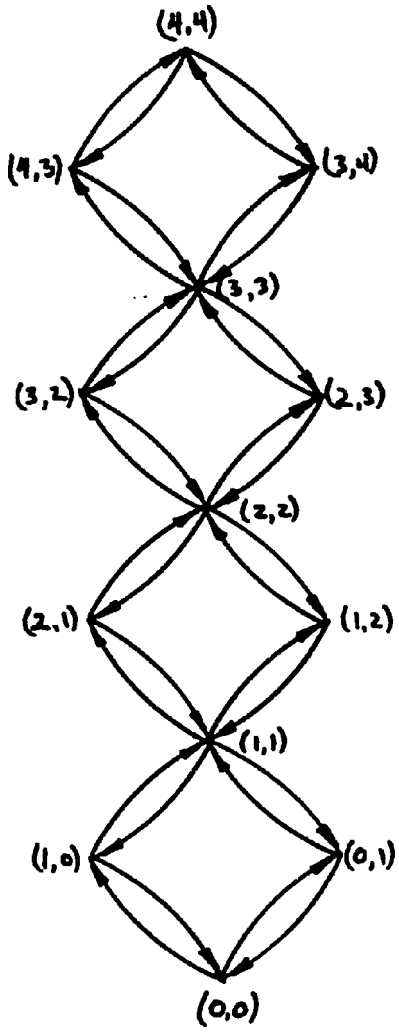


Fig. 3-2. The instantaneous jockey transition diagram.

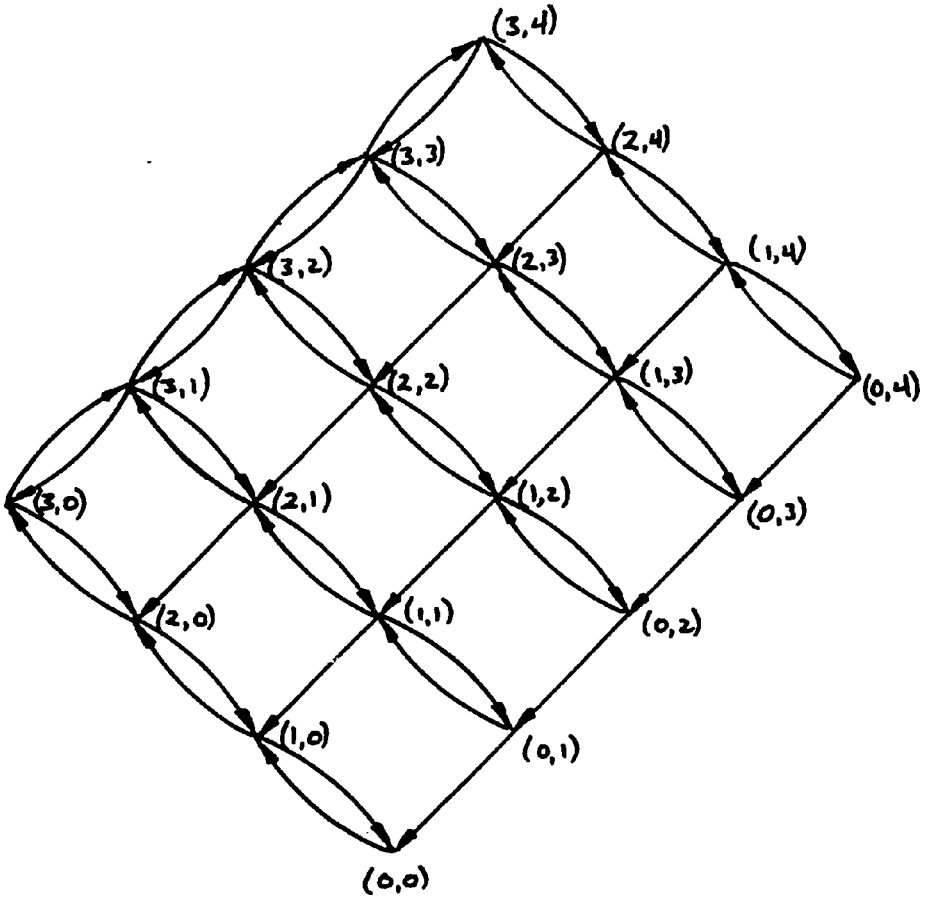


Fig. 3-3. The overflow transition diagram.

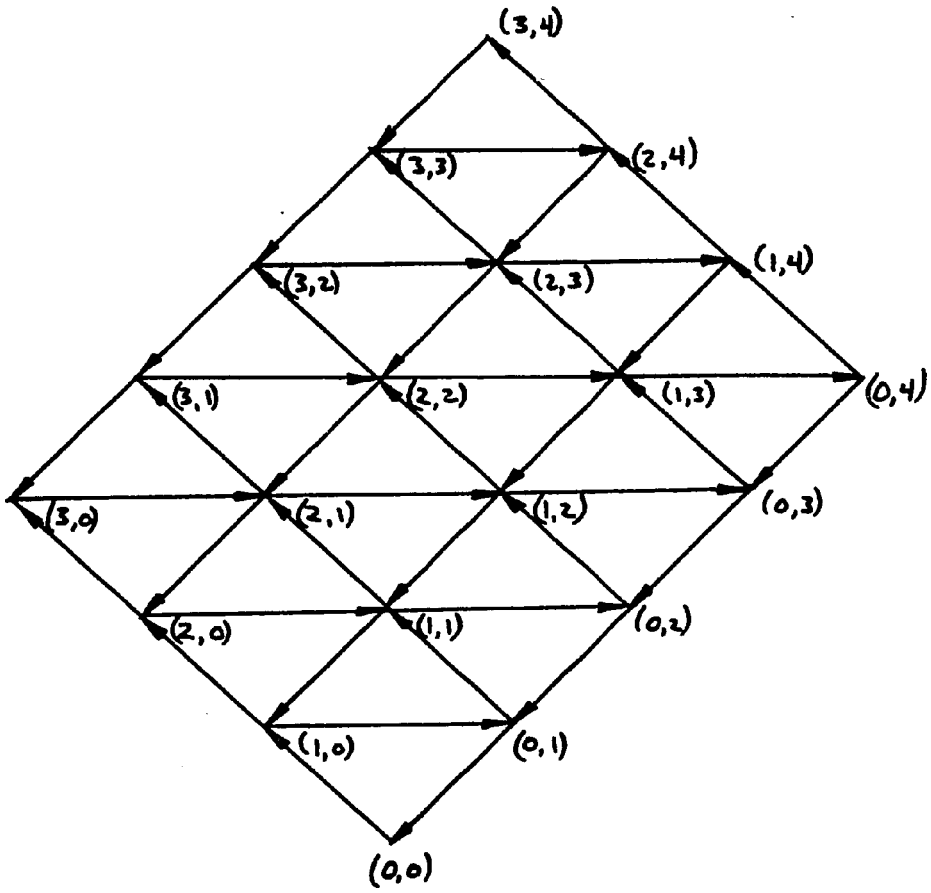


Fig. 3-4. The two-servers-in-series transition diagram.

The first three cases are members of the class of parallel systems, mentioned in chapter 1. Accordingly, they have transition diagrams with no horizontal arrows. Every transition is either an arrival or a departure; hence every arrow is between states in different levels. Use will be made of this property in chapter 7. The fourth example illustrates a system in which internal transitions are possible.

The obvious simplicity of the structure of these diagrams cannot be overlooked. Although the two-dimensional lattice-like structure is typical only of two-server systems, it is a fact that three-server systems have characteristic three-dimensional lattice-like structure, and so forth.

At least when it is neatly drawn, the transition diagram places the states of the process in what might be called their proper or natural order. Forcing the states into a linear sequence by numbering them with integers and then attempting to describe transitions in terms of that numbering results in unnecessary complexity. The transition diagram reveals that, regardless of how the states are designated, the transitions are really quite simple. Since when dealing with networks of queues it is important not to be conceptually bound to an arbitrary designation of state, the transition diagram is a convenient device to describe how the process works. Indeed, after the transition diagram is drawn, the original context of the process may be ignored, and the process may be viewed as a random walk in continuous time over the diagram.

The Principle of Statistical Equilibrium

Aside from its value as a conceptual aid, the transition diagram is useful in direct fashion for formulating the equations of equilibrium, (2.15). Define the mean transition rate from state i to state j to be the steady state probability of being in state i multiplied by the transition rate from i to j, or $q_i\lambda_{ij}$. The principle of statistical equilibrium states that, in order for the system to be in equilibrium, the mean transition rate into any state must be equaled by the mean transition rate out of that state. (See [23], [24], and [28].) Algebraically, for any j,

$$\sum_{i\neq j} q_i\lambda_{ij} = \sum_{k\neq j} q_j\lambda_{jk}$$

$$= q_j \sum_{k\neq j} \lambda_{jk}$$

or, using (2.3),

$$= q_j(-\lambda_{jj})$$

Stated differently,

$$\sum_{\text{all } i} q_i\lambda_{ij} = 0, \text{ for all } j$$

The set of these equations for all j taken together constitute just exactly the matrix equation (2.15), which was obtained through probability arguments.

In practice, this means that the principle of statistical equilibrium used in conjunction with the transition diagram (from which one can determine by inspection the mean transition rate into or out of a state) can substitute

for the tedious derivation of each equation or the equally tedious matrix multiplication indicated in (2.15). Moreover, the principle of statistical equilibrium has strong intuitive appeal and can be rigorously justified, at least in the sense that it always produces the correct result.

CHAPTER 4

GRAPH THEORY

In spite of its apparent reliance upon "pictures", the formal theory of graphs is an accepted, legitimate field of pure mathematics. It could, in fact, exist without pictures, as a special category of set theory. It is the pictures, however, which make graph theory appealing. They provide intuitive insight into concepts which might, in set theoretic terms, be very difficult to grasp. These remarks apply to the general theory of graphs and, in particular, to that subcategory, the theory of directed graphs, that will be required in this paper.

This chapter will present a large number of definitions related to directed graphs. It would be an imposing task to master all of the concepts to be presented at once were it not for the intimate relation between the concepts and real or imagined pictures. Because of the association,—provided that a concept is genuinely and correctly understood at the time it is defined—one may generally rely upon his intuition in using the concept. For the most part, the terminology will be consistent with that of Harary [15].

Definitions For Directed Graphs

A directed graph (digraph for short) consists of a non-empty set X of points and a set L of directed lines. A directed line is formally an ordered pair of points of X. In drawing a digraph, the line (x,y) is drawn as an ordinary geometric curve from x to y, with an arrowhead pointing toward y to indicate that x is the first point of the ordered pair and y is the second point. We say that the line is incident with both of its points, incident from its first point, and incident to its second point. As a part of the definition, a digraph may not have multiple lines in the same direction between the same two points. It may, however, have lines of the form (x,x). Such lines are called loops. It should be pointed out that Harary's definition of a digraph excludes loops. Figure 4-1 shows a typical digraph having three points.

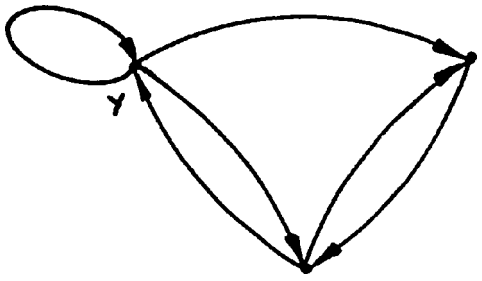


Fig. 4-1. A typical digraph.

Given a digraph D, the number of lines incident to a point x is called the <u>in-degree</u> of the point x, and is denoted id(x). Similarly, the number of lines incident from x is called the <u>out-degree</u>, and is denoted by od(x). In figure 4-1, id(y) = 2 and od(y) = 3.

A <u>subgraph</u> of a digraph D is a digraph whose points and lines are points and lines of D. A <u>spanning subgraph</u> of a digraph D is a subgraph whose set of points is the same as that of D. That is, it contains all of the points and some of the lines of D. Two subgraphs of a digraph are <u>disjoint</u> if they have no points in common. By implication, they may have no lines in common. Figure 4-2 shows a spanning subgraph of the digraph shown in figure 4-1. Figures 4-3 and 4-4 illustrate subgraphs of the same digraph which are disjoint.

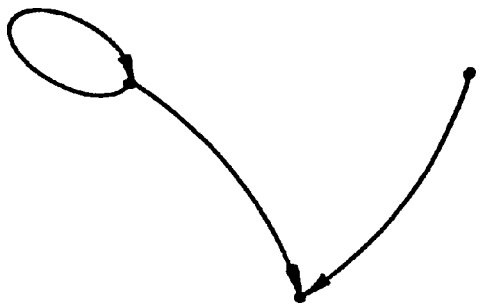


Fig. 4-2. A spanning subgraph of D.



Fig. 4-3. A subgraph of D.



Fig. 4-4. Another subgraph.

The subgraph D-x is the digraph obtained from D by removing the point x and all lines incident to or from x. More generally, if S is any subgraph of D, D-S is the digraph obtained from D by removing all of the points of S and all lines incident to or from points of S. Obviously, S and D-S are disjoint subgraphs of D. If D is the digraph shown in figure 4-1 and S is the subgraph of D shown in figure 4-4, then D-S is the digraph shown in figure 4-3. Notice, however, that if S is the subgraph shown in figure 4-3, then D-S is not the digraph shown in figure 4-4, but rather the one shown in figure 4-5.

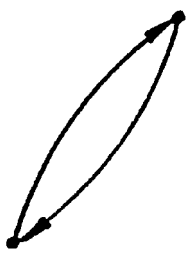


Fig. 4-5. D-S.

If S is a subgraph of D and x is a point of D which is not contained in S, then S+x is the digraph obtained from S by adding the point x and all lines in D between x and points of S. More generally, if S and T are disjoint subgraphs of D, then S+T is the digraph whose point set is the union of the point sets of S and T, and all lines of D which go from a point of S to a point of T, or from a point of T to a point of S. Again referring to the same example, if D is the digraph of figure 4-1, S is the subgraph of figure 4-3, and T is the subgraph of figure 4-4, then S+T is shown in figure 4-6.

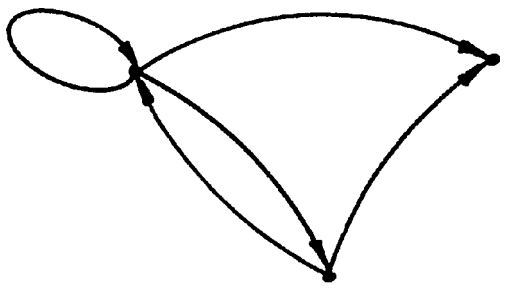


Fig. 4-6. S+T.

Two points x and y are adjacent in D if the line (x,y) or the line (y,x) is in D. They are joined in D if they are adjacent or if there exists a sequence of points of D, z_1, z_2, ... z_n, such that x and z_1 are adjacent, z_i and z_{i+1} are adjacent (for all i from 1 to n-1), and z_n and y are adjacent. In figure 4-7, x and y are joined but not adjacent. The points x and z, on the other hand, are adjacent, as are z and y.

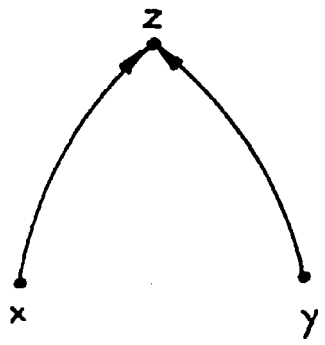


Fig. 4-7. Adjacency.

A digraph D is (weakly) connected if every two points of D are joined. Although each of the digraphs shown so far has been connected, figure 4-8 illustrates one which is not. A component of a digraph D is a maximal connected subgraph of D; that is, a connected subgraph of D such that there is no other connected subgraph of D which properly contains it. The digraph of figure 4-8 consists of two components. The components of a digraph are, of course, disjoint subgraphs.

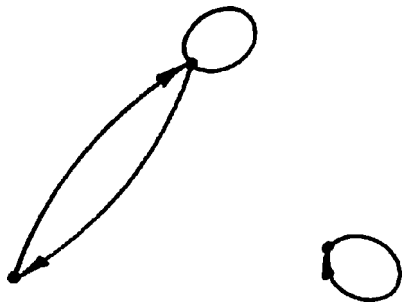


Fig. 4-8. A digraph with two components.

A <u>path</u> from x_1 to x_n is a sequence of points $x_1, x_2, \ldots x_n$, together with the lines (x_1,x_2), (x_2,x_3), ... (x_{n-1},x_n), such that no point appears more than once in the sequence. A path consisting of a single point is called <u>trivial</u>. A <u>cycle</u> consists of a path together with a line from the terminal point of the path to the initial point. A <u>trivial cycle</u>, constructed in this manner from a trivial path, is just a loop. A path or a cycle within a digraph D may be considered to be a subgraph of D. Figures 4-9 and 4-10 illustrate a path and a cycle, respectively. Figure 4-11 shows a digraph which is not a path (because one of the points appears twice in the sequence) and figure 4-12 shows another which is not a path, because one of the lines points in the wrong direction.



Fig. 4-9. A path.



Fig. 4-10. A cycle.



Fig. 4-11. A non-path.



Fig. 4-12. A non-path.

The <u>length</u> of a path or cycle is the number of lines it contains. In the case of a cycle, the number of lines will be equal to the number of points, but the two will not be equal in the case of a path. We may say, however, that the length of a path is one less than the number of points it contains.

A <u>1-factor</u> H of a digraph D is a spanning subgraph such that for every point x of H, id(x) = od(x) = 1. More intuitively, H consists of disjoint cycles which collectively span D. A <u>1-factor connection</u> H_{xy} from point x to point y of D is a spanning subgraph such that id(x) = od(y) = 0, od(x) = id(y) = 1, and for every other point z of H_{xy}, id(z) = od(z) = 1. Intuitively, a 1-factor connection from x to y consists of a path from x to y and a set of disjoint cycles— disjoint from each other and from the path—such that the path and the cycles collectively span D. Figures 4-13 and 4-14 illustrate, respectively, a 1-factor and a 1-factor connection from x to y.

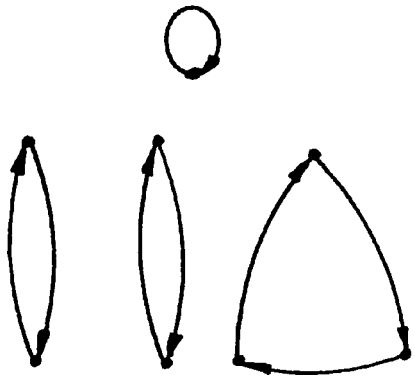


Fig. 4-13. A 1-factor.

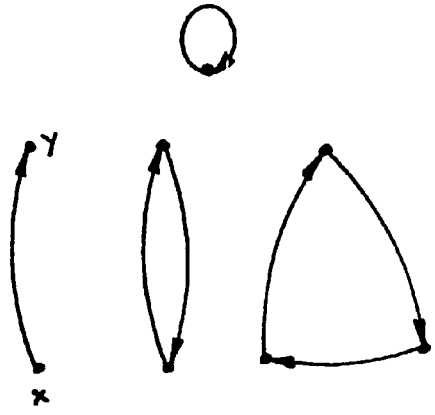


Fig. 4-14. A 1-factor connection.

A labeled digraph is a digraph whose points are distinguished by naming them in some manner. Thus the two digraphs of figure 4-15 are identical as digraphs but different as labeled digraphs. As labeled digraphs, they



Fig. 4-15. Two labeled digraphs.

are, however, isomorphic. Two labeled digraphs D_1 and D_2 are isomorphic if there exists a one-to-one mapping, say γ, of the points of D_1 onto the points of D_2 which preserves the directed lines (i.e., for (x,y) a line in D_1, $(\gamma(x),\gamma(y))$ must be a line in D_2).

A weighted labeled digraph is a labeled digraph with as associated function f from the lines of D into the real numbers. In drawing a weighted labeled digraph, the weight of a line—that is, the value of the function for that line—is written next to it. Two weighted labeled digraphs are isomorphic if they are isomorphic as label digraphs and, in addition, the isomorphism function γ preserves the weights of all lines (formally, for f(x,y) the weight of a line (x,y) in D_1, $f(x,y) = f(\gamma(x),\gamma(y))$). Figure 4-16 shows two isomorphic weighted labeled digraphs. The mapping γ is in this case defined by $\gamma(x_i) = y_i$.



Fig. 4-16. Two isomorphic weighted labeled digraphs.

The weight of a subgraph S of a weighted labeled digraph, written f(S), is the product of the weights of the lines in S. The absolute weight of a subgraph S, written f'(S), is the product of the absolute values of the weights of the lines in S.

As an alternative to the usual pictorial representation, a weighted labeled digraph can be represented by its variable adjacency matrix V. V is a square matrix, having as many rows and columns as the digraph has points. If (x_i,x_j) is a line of the digraph, the element v_{ij} of V has the value $f(x_i,x_j)$. If there is no line from x_i to x_j, $v_{ij} = 0$. It is clear that the pictorial representation can be reconstructed from V, so the two are truly alternative representations of the same structure.

In Harary's terminology, a weighted labeled digraph (with loops allowed) is called a network. Because of possible confusion with the physical systems to be studied, this term will be avoided when speaking in the context of

graph theory. Instead, a weighted labeled digraph will be referred to in exactly that manner when there is any possibility of ambiguity, and as simply a graph when there is none.

Given a square matrix A, the flowgraph of A, denoted G, is a weighted labeled digraph with as many points as A has columns, such that if $a_{ji} \neq 0$, there is a directed line from i to j with the weight a_{ji}, and if $a_{ji} = 0$, there is no line from i to j. Stated another way, it is "that weighted labeled digraph whose adjacency matrix is the transpose of A."

Figure 4-17 shows the flowgraph of the matrix

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 3 & 2 & 1 \\ 5 & 2 & 0 \end{bmatrix}$$

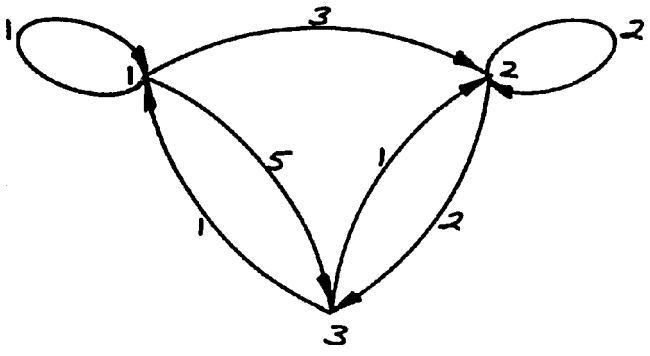


Fig. 4-17. The flowgraph of A.

Given a system of N independent linear equations in N unknowns, Ax=b, the flowgraph of the equations, denoted G_0, is the flowgraph of A with one additional point, called the source node, appended. A line is drawn from the source node to each point k for which $b_k \neq 0$, and that line is assigned the weight $-b_k$. No lines are directed toward the source node.

The flowgraph of the equations

$$\begin{bmatrix} 1 & 0 & 1 \\ 3 & 2 & 1 \\ 5 & 2 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 6 \\ 9 \end{bmatrix}$$

is shown in figure 4-18.

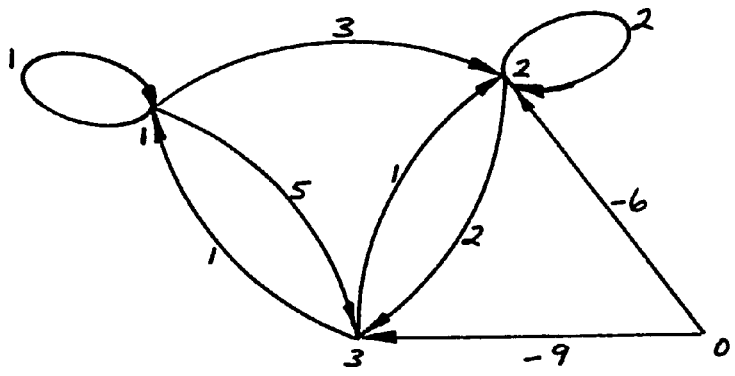


Fig. 4-18. The flowgraph of Ax = b.

It is worth noting parenthetically that the structure of the flowgraph of a matrix or a set of equations depends very much upon the order of rows and columns of A. For example, interchanging the first and second equations of the example shown, while it certainly does not affect the solution to the equations, produces the entirely different flowgraph shown in figure 4-19.

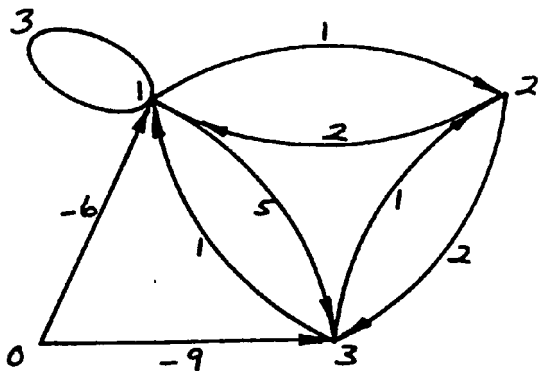


Fig. 4-19. Another flowgraph of the same equations.

That the two graphs are not isomorphic can be seen from the fact that one of them has two loops, while the other has only one.

It is also worth noting that the flowgraph defined here is that invented by Coates, not to be confused with the signal flowgraph of Mason.

Coates' Theorem

Coates' theorem provides the computational formula to solve a set of linear equations of the form Ax=b by means of the flowgraph of the equations.

Theorem 4.1. (Coates, 1959) The solution to a set of N independent linear equations in N unknowns, Ax=b, is given by

$$x_i = \frac{\sum (-1)^C f(H_{Oi})}{\sum (-1)^C f(H)} \qquad \text{for } i=1,2,\ldots N$$

where the numerator sum is over all 1-factor connections from the source node to point i in the flowgraph of the equations, the denominator sum is over all 1-factors in the flowgraph of A, and C is the number of cycles in a 1-factor or 1-factor connection.

Proof: The details of the proof are of no concern here; they may be found in [5], [8], or [16]. However, it is important to understand the fundamental basis of the association between the solution to a set of equations and these particular sums of weights of subgraphs.

By Cramer's rule,

$$x_i = \frac{\sum_k b_k \Delta_{ki}}{\Delta}$$

where Δ is the determinant of A, Δ_{ki} is the cofactor of the (k,i)-th element of A. The proof of Coates' theorem rel[illegible]s upon establishing the identities

$$\sum_k b_k \Delta_{ki} = (-1)^N \sum_{H_{0i}} (-1)^C f(H_{0i})$$

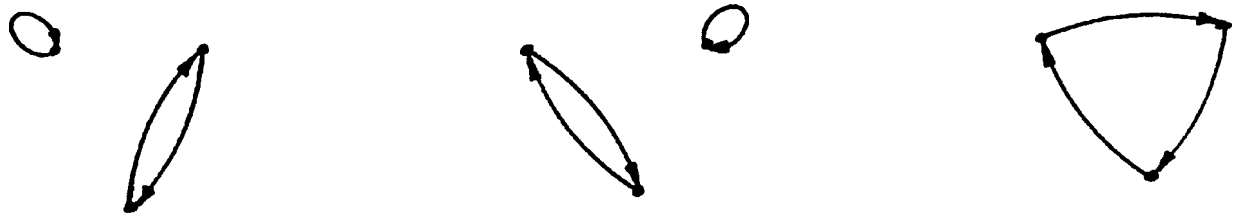
and

$$\Delta = (-1)^N \sum_H (-1)^C f(H)$$

Each of these identities is proved similarly. Without becoming enmeshed in details, the proof uses the definition of a determinant as a sum over all permutations of products of elements of A. The cycles of a permutation are directly related to the cycles of a flowgraph. A very clear presentation of this relationship is given by Hoskins [16].

Given these identities, substitution in Cramer's formula and cancellation of the common factor $(-1)^N$ produces Coates' theorem as given above.

To illustrate, using the flowgraph of figure 4-18, the denominator consists of the three terms shown in figure 4-20.



f(H) = 2
C = 2

f(H) = 10
C = 2

f(H) = 6
C = 1

Fig. 4-20. The denominator terms.

Consequently, the denominator is equal to 2+10-6 = 6. The numerator, if x_1 is to be solved for, is calculated from the terms shown in figure 4-21.



$f(H_{01}) = -18$ $\quad C = 1$ $\qquad$ $f(H_{01}) = -12$ $\quad C = 0$

Fig. 4-21. The numerator terms.

So the numerator is 18-12 = 6, and the solution is $x_1 = 1$. The determination of x_2 and x_3 is similar.

It will be convenient to define for any labeled, weighted digraph G, the graph determinant of G, denoted Δ_G, as follows:

$$\Delta_G = \sum_H (-1)^C f(H)$$

where the sum is over all 1-factors H of G, and C is the number of cycles in H.

If G is the flowgraph of a matrix A, as in the proof above, Δ_G differs from the ordinary determinant of A by at

most a change of sign. In fact, from the second identity above,

$$\Delta = (-1)^N \Delta_G$$

Because of the cancellation of $(-1)^N$, it is the graph determinant rather than the true determinant of A which appears in the denominator of Coates' formula.

The numerator can also be expressed in terms of graph determinants. The result is the following alternative version of Coates' formula.

Theorem 4.2. The solution to a set of N independent linear equations in N unknowns, Ax=b, is given by

$$x_i = \frac{\sum_{P_{0i}} f(P_{0i}) \Delta_{G_0 - P_{0i}}}{\Delta_G}$$

where the numerator sum is over all paths from 0 to i in G_0, the flowgraph of the equations, and G is the flowgraph of A.

Proof: The validity of the denominator is obvious from the above remarks. The numerator must be shown to equal the numerator in theorem 4.1.

A 1-factor connection from i to j is just a path from i to j and a 1-factor of the remaining subgraph. The number of cycles of the 1-factor connection is of course the number of cycles in the 1-factor of the remaining subgraph. Therefore, for any 1-factor connection H_{0i},

$$(-1)^C f(H_{0i}) = f(P_{0i})(-1)^C f(H_{G_0 - P_{0i}})$$

where P_{0i} is the path in H_{0i} and $H_{G_0-P_{0i}}$ is the 1-factor of the remaining subgraph.

Collecting all of the 1-factor connections which have the particular path P_{0i} in common,

$$\sum (-1)^C f(H_{0i}) = \sum f(P_{0i})(-1)^C f(H_{G_0-P_{0i}})$$

where the first sum is over all H_{0i} containing P_{0i}, and the second sum is over all 1-factors of G_0-P_{0i}. Continuing,

$$\sum (-1)^C f(H_{0i}) = f(P_{0i}) \sum (-1)^C f(H_{G_0-P_{0i}})$$

$$= f(P_{0i}) \Delta_{G_0-P_{0i}}$$

Finally, summing over all paths from 0 to i,

$$\sum_{H_{0i}} (-1)^C f(H_{0i}) = \sum_{P_{0i}} f(P_{0i}) \Delta_{G_0-P_{0i}}$$

This completes the proof of theorem 4.2.

Theorem 4.2 restates Coates' theorem in a form which clarifies the fact that the basic operation involved in solving equations by means of flowgraphs is evaluating graph determinants.

PART II

THEORY

CHAPTER 5

SOLUTION TECHNIQUES

Before undertaking the details of the solution method to be developed in this chapter, consider the problem as it presently stands. The objective is to solve the linear equations (2.15), $q\Lambda_a = 0$, and (2.16), $\sum q_i = 1$. Among the various ways to go about solving these equations is to use the flowgraph method of Coates, as presented in chapter 4. Actually, Coates' method is not immediately applicable because the equations are not all independent; but this difficulty is easily surmounted by discarding any one of the equations of $q\Lambda_a = 0$.

Because <u>any</u> of these equations may be discarded to eliminate the dependency, and because the remaining equations may be arranged in any order, the specific form of the system of equations to which Coates' method could be applied may be freely selected from a number of alternative forms. But it was illustrated by means of an example in chapter 4 that the flowgraph of a set of equations depends upon the specific form of the equations. There is no <u>a priori</u> reason to prefer one flowgraph to another; any will suffice to obtain the correct solution.

It turns out—as the first section of this chapter will prove—that, when the equations are set up in one particular way, the flowgraph is essentially equivalent to the transition diagram of the system. Now, to draw a transition diagram requires only an intuitive understanding of how the system operates. It does not require the steady state equations; quite to the contrary, the equations are frequently obtained from the transition diagram by utilizing the principle of statistical equilibrium, discussed is chapter 3.

The equivalence between the transition diagram and the flowgraph of the steady state equations (set up in some particular way, which way becoming irrelevant once the theorem is proved) suggests, as a solution method, the direct interpretation of a transition diagram as a flowgraph and the use of Coates' formula on this graph to obtain the solution. That is, the formulation of the equations and the subsequent construction of a corresponding flowgraph may be by-passed.

Actually, the method to be proposed and utilized in this paper is not precisely that just described. Although the flowgraph concepts will be retained, Coates' formula itself happens to be unnecessarily unwieldy in the context of the problems considered here. Thus the role of the equivalence theorem is only to establish the relationship between a flowgraph and the transition diagram. The

relationship is further exploited to obtain the solution technique which is actually used.

Aside from this introductory overview of the approach, the first task of this chapter is to prove the equivalence theorem, theorem 5.1. After that is done, the problem will be simplified slightly by changing variables. Next, a new formula for the value of a graph determinant will be given in theorem 5.2. This formula is less general than that of Coates, in that it applies only to those flowgraphs which are really transition diagrams; but for such flowgraphs, it is superior to Coates' formula. Subsequent to proving theorem 5.2, two entirely different algorithms for evaluating graph determinants, both dependent on theorem 5.2, will be provided. Finally, the complete solution technique will be summarized and illustrated.

The Equivalence Theorem

The fundamental principle underlying the solution technique to be developed within this paper is that the transition diagram of a finite, irreducible, discrete-state, continuous-parameter Markov process is "essentially equivalent" to the flowgraph of the steady-state equations of the process. In this statement, however, the phrase "essentially equivalent" should not be misinterpreted; the two graphs are not identical. They are "essentially equivalent" in the sense that after suitable modifications (of a simple nature)

have been made to the transition diagram, it can be used as a flowgraph to solve the equations.

In chapter 4 it was pointed out, by means of an example, that the flowgraph of a set of linear equations depends upon the way in which the equations are arranged. In other words, a set of equations can be solved using any of a number of genuinely different (i.e., non-isomorphic) flowgraphs, each of which could be said to be "the" flowgraph associated with the equations. But the modified transition diagram mentioned above cannot be isomorphic to each of several flowgraphs which are not isomorphic to each other. Thus the statement that the modified transition diagram can be regarded as a flowgraph is really the assertion of the existence of an arrangement of the steady-state equations such that the flowgraph of that arrangement is isomorphic to the modified transition diagram.

Prior, then, to proving theorem 5.1—the equivalence theorem—there are two objectives to accomplish. The first is to state the way in which the transition diagram must be modified, and the second is to present the steady-state equations in that particular form which is required by the theorem. The first objective is met within the following definition.

The <u>modified transition diagram</u> of a Markovian service system is that weighted, labeled, directed graph obtained from the ordinary transition diagram, defined in chapter 3, by effecting the following two changes, in the order given:

(a) First, a loop is attached to each point of the diagram and is given as associated weight equal to the negative of the sum of the transition rates out of the state.

(b) Next, all lines entering the zero state are deleted. An example of a transition diagram before and after modification is shown in figure 5-1.

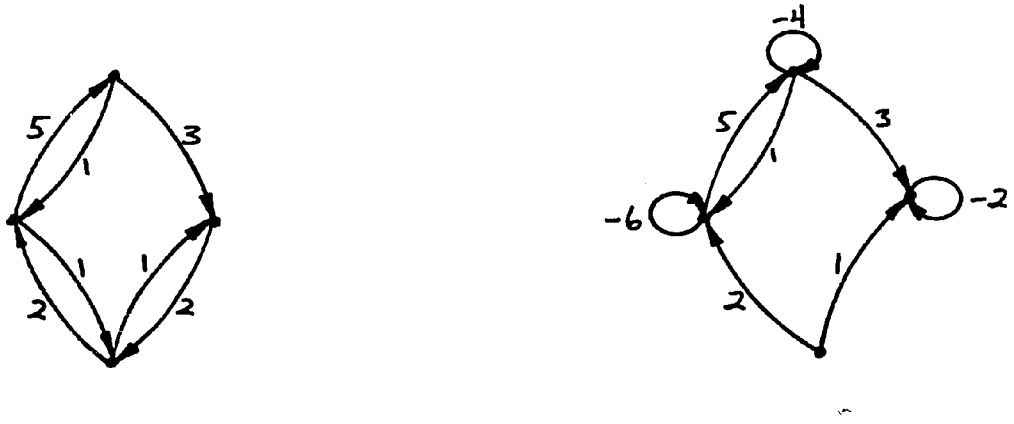


Fig. 5-1. Before and after modification.

Because only genuine changes of state were represented, the transition diagram contained no loops prior to modification. If we denote the loop weight at point i by λ_{ii}, the first step of the modification can be formally restated as

$$\lambda_{ii} = -\sum_{j \neq i} \lambda_{ij}$$

This is consistent with the notation used in chapter 2. (See, for example, equation 2.13.) Thus part (a) amounts to adding to the transition diagram loops having weights equal to the diagonal terms in λ_a. The second step deletes all lines having weights of the form λ_{i0}.

Now the steady-state equations, previously given ((2.15) and (2.16)) in the form

$$q\Lambda_a = 0 \tag{5.1}$$

$$\Sigma q_i = 1 \tag{5.2}$$

must be manipulated to the form required by the equivalence theorem. The normalizing equation, (5.2), will be temporarily disregarded. The other equations, (5.1), will be used to solve for all of the q_i in terms of q_0. Recall that the state numbered zero is the one which refers to the empty system. The choice of this state rather than some other one is arbitrary but convenient for a number of reasons which will become clear in succeeding chapters. Moreover, solving for all probabilities in terms of q_0 is the procedure which is typically followed in algebraically solving simple Markov systems, such as those which are birth-death processes.

In conformity with the usual practice of placing the matrix of known coefficients to the left of the vector of unkowns, the equations (5.1) will be transposed into the equivalent form

$$\Lambda_a^t q^t = 0 \tag{5.3}$$

or, written out in detail,

$$\begin{aligned} \lambda_{00}q_0 + \lambda_{10}q_1 + \ldots + \lambda_{N0}q_N &= 0 \\ \lambda_{01}q_0 + \lambda_{11}q_1 + \ldots + \lambda_{N1}q_N &= 0 \\ &\vdots \\ \lambda_{0N}q_0 + \lambda_{1N}q_1 + \ldots + \lambda_{NN}q_N &= 0 \end{aligned} \tag{5.4}$$

Notice that the transposition of Λ_a has caused the subscripts in (5.4) to appear in the reverse of the usual order.

These equations are not independent. By virtue of (2.12), the conservative property, any row of Λ_a^t is equal to the negative of the sum of the other rows. Hence we are free to discard <u>any</u> of the equations in (5.2). It is necessary to the theorem that follows that the first of these equations above, the one which contains λ_{00}, be the one which is discarded.

Dividing the remainder of the equations by q_0 (which, from chapter 2, is >0) and rearranging, we obtain

$$\begin{aligned} \lambda_{11}\frac{q_1}{q_0} + \lambda_{21}\frac{q_2}{q_0} + \ldots + \lambda_{N1}\frac{q_N}{q_0} &= -\lambda_{01} \\ \vdots \qquad\qquad\qquad\qquad \vdots \qquad &\vdots \\ \lambda_{1N}\frac{q_1}{q_0} + \quad . \quad . \quad . \quad + \lambda_{NN}\frac{q_N}{q_0} &= -\lambda_{0N} \end{aligned} \tag{5.5}$$

The result is a linearly independent set of equations which uniquely determine values for the q_i/q_0. Call these variables the <u>true coefficients</u> (of q_0), and denote them by the column vector c:

$$c^t = (c(1),c(2),\ldots,c(N)) = (\frac{q_1}{q_0}, \frac{q_2}{q_0}, \ldots, \frac{q_N}{q_0}) \tag{5.6}$$

From this point on, there will be no further need for the original system of equations, so let Λ refer to the matrix obtained from Λ_a by removing the first row and the first column (or all elements which have zero as either subscript). Let λ_0 be the column vector of transition

rates out of the zero state:

$$\lambda_0^t = (\lambda_{01}, \lambda_{02}, \cdots, \lambda_{0N})$$

The equations to be solved, (5.5), can now be expressed as

$$\Lambda^t c = -\lambda_0 \tag{5.7}$$

These equations substitute for (2.15) as the fundamental equations to be solved. It is to be implicitly understood that after all of the c(i) are determined, the originally desired solution will be recovered from the relation $q_i = c(i)q_0$, the value of q_0 being determined by the normalizing equation, (5.2).

The flowgraph of a Markovian service system is defined to be the flowgraph of the equilibrium equations of the system arranged as in (5.7).

Repeating for emphasis, what has just been defined as "the" flowgraph of a system is really only one of many which could be used to solve the equations (5.1) and (5.2). Any of the equations of (5.1) could have been discarded to eliminate the dependency; the normalizing equation could have been explicitly included, which would permit one to solve for q_i directly; rows and columns could have been shifted about; and so forth. Any of these changes could produce a flowgraph genuinely different from (i.e., non-isomorphic to) the one we have chosen to call "the" flowgraph. Of course, the final solution would be the same no matter which was used. But it is only "the" flowgraph

which satisfies the requirements of the equivalence theorem, as will become apparent when the theorem is proved.

Theorem 5.1. (Equivalence theorem) The flowgraph of a Markovian service system is isomorphic to the modified transition diagram of the system.

Proof: The required isomorphism is the obvious one, taking state i of the modified transition diagram onto point i of the flowgraph, for $i \neq 0$, and taking the zero state onto the source node. To show that this is indeed an isomorphism, we must verify that the lines and their weights are preserved. First of all, the loop values correspond because the modification or the ordinary transition diagram was made in such a way that the loop weights would equal the diagonal terms in Λ^t. It remains to show that the other lines and weights correspond.

Ignore for the moment the zero state of the transition diagram and source node of the flowgraph. In the transition diagram, a line is drawn from state i to state j if and only if $\lambda_{ij} \neq 0$, and the line is labeled with the transition rate λ_{ij}. In the flowgraph, a line is drawn from point i to point j if and only if $a_{ji} \neq 0$, and the line is given the weight a_{ji}. But the matrix A is in this case Λ^t, so $a_{ji} = \lambda_{ij}$ (note the convenient reversal of subscripts). This establishes the equivalence with respect to all lines except those associated with the zero state and source node. In the modified transition diagram, there is a line from

0 to k if and only if $\lambda_{0k} \neq 0$. In the flowgraph, there is a line from the source node to k if and only if $b_k \neq 0$, and this line is given the weight $-b_k$. But in this case $b_k = -\lambda_{0k}$, so $-b_k = \lambda_{0k}$. This completes the proof of theorem 5.1.

The most immediate implication of theorem 5.1 is that it is no longer necessary to formulate explicitly the steady-state equations of a process. The transition diagram, as was pointed out in chapter 3, contains the equations in implicit form. Heretofore, obtaining the solution from the transition diagram involved first explicitly formulating the equations by means of the principle of statistical equilibrium, then solving the equations by algebraic methods. Theorem 5.1 implies that the solution can be obtained directly from the transition diagram without ever seeing the equations.

A second implication of the theorem is that "the" flowgraph of a system is not just any flowgraph, but a flowgraph which has all of the properties of a (modified) transition diagram. These properties include certain sign requirements, a relation between the weight of a loop and the weights of other lines leaving the same point, and possibly a great deal of structure, to name a few. It is conceivable that some of these properties may be useful in simplifying the solution procedure.

Such, indeed, is the case. The remainder of this chapter will be devoted to the development of flowgraph

techniques which apply to those particular flowgraphs which have the properties of a transition diagram.

Relative Coefficients

Inasmuch as the usual application of Coates' theorem involves the separate evaluation of the numerator and denominator in Cramer's formula,

$$c(i) = \frac{\sum_k -\lambda_{0k}\Delta_{ki}}{\Delta}$$

and since the denominator is the same for all c(i), let us define a set of relative coefficients as follows

$$c^*(i) = c(i)\Delta \quad \text{for } i \neq 0$$

$$c^*(0) = \Delta$$

Thus $c^*(i)$ for $i \neq 0$ is just the numerator in Cramer's formula, and $c^*(0)$ is the denominator.

It immediately follows that

$$c(i) = \frac{c^*(i)}{c^*(0)} \tag{5.8}$$

To carry this further, let us define

$$q_0^* = \frac{q_0}{c^*(0)} \tag{5.9}$$

It will then be correct to write

$$q_i = c^*(i)q_0^*, \text{ for all } i=0,1,\ldots N, \tag{5.10}$$

because

$$q_i = c(i)q_0 = \frac{c^*(i)}{c^*(0)}\, q_0 = c^*(i)\frac{q_0}{c^*(0)} = c^*(i)q_0^*$$

Furthermore, from 5.10 it follows directly that

$$\sum_{i=0}^{n} c^*(i) q_0^* = 1 \qquad (5.11)$$

because

$$\sum c^*(i) q_0^* = \sum q_i = 1$$

What this means is that we can solve for the relative coefficients, $c^*(i)$, rather than the true coefficients, $c(i)$. When this is done, q_0^* can be evaluated by means of (5.11), which is just a restatement of the normalizing equation. The final solution is obtained by means of (5.10). It will not be necessary ever to know the true coefficients or to perform the division indicated in (5.8).

An Improved Graph Determinant Formula

One of the factors tending to make Coates' formula unwieldy is the necessity for care in avoiding sign errors. But this factor can be virtually eliminated by utilizing what is known about the signs of weights of lines in a flowgraph which has the properties of a transition diagram. From chapter 2, we may be certain that $\lambda_{ij}>0$ for $i\neq j$—that is, the weight of any line between two distinct points is strictly positive—because λ_{ij} is a transition rate. We may also be certain that for any i, $\lambda_{ii}<0$—the weight of any loop is strictly negative.

As a consequence, of all the subgraphs that are of interest to us, loops and only loops have negative weight. The weight of a path or of a cycle of length greater than one is necessarily positive. Hence, for those flowgraphs which are really modified transition diagrams, the sign of the product of weights of lines in a path is positive; and the sign of the product of weights of lines in a 1-factor depends only on the number of trivial cycles it contains. This latter point suggests the next theorem.

Recall from chapter 4, where it was defined, that f'(S) denoted the product of the absolute values of the weights of the lines in S.

Theorem 5.2. For G a modified transition diagram, or a subgraph thereof,

$$\Delta_G = \sum_H (-1)^k f'(H)$$

where the sum is over all 1-factors H of G, and k is the number of non-trivial cycles of H.

Proof: f(H) in the original graph determinant formula, (4.5), is a product of line weights. It is positive if an even number of the lines of H are negative; it is negative if an odd number of them are. But only loops have negative weight, so f(H) is positive if H contains an even number of loops and negative if it contains an odd number of loops. Thus if L denotes the number of loops in a 1-factor H,

$$f(H) = (-1)^L f'(H)$$

Therefore,

$$(-1)^C f(H) = (-1)^{C+L} f'(H)$$

But since C+L has the same parity as C-L,

$$(-1)^C f(H) = (-1)^{C-L} f'(H)$$

Now C-L, the number of cycles minus the number of loops, is just the number of non-trivial cycles k. Summing over all 1-factors H,

$$\sum_H (-1)^C f(H) = \sum_H (-1)^k f'(H)$$

Since the left hand side is the definition of Δ_G, (4.5), the proof is complete.

The advantage of the graph determinant evaluation formula of theorem 5.2, as opposed to that of Coates, equation (4.5), is that the determination of the sign of a term in the sum has been greatly simplified. Using theorem 5.2, one may calculate the value of f'(H) <u>ignoring</u> the signs of the weights of lines (that is, simply multiplying the weights as if they were all positive), then determine the sign to precede f'(H) by counting the non-trivial cycles in H. In contrast to this approach, Coates' formula requires that one calculate f(H) as a signed product (negative if an odd number of lines of H are negative, positive otherwise), then reverse the sign if the total number of cycles in H is odd. Of course, the simplification was acquired by using the properties of a transition diagram, so

theorem 5.2 cannot be used to calculate the graph determinant of an arbitrary flowgraph.

Throughout the remainder of this paper, modified transition diagrams will be drawn and referred to in proofs as if all weights were positive. Such a diagram is not, strictly speaking, a flowgraph; but with the understanding that the formula of theorem 5.2 is to be used consistently, graph determinants can be evaluated just as if it were a true flowgraph. In addition, path weights will be the same regardless of whether loop weights are considered to be positive or negative, because a path connot contain a loop.

Consider, now, the implications of regarding all weights as positive. In such a diagram, $f'(H) = f(H)$; that is, it is not really necessary to retain the prime, because in either case the weight is just a product of positive numbers. However, the prime <u>will</u> be retained as a reminder that the graph determinant evaluation formula to be used is that of theorem 5.2. Secondly, the loop weights, λ_{ii}, formerly stated as the negative sum of weights of lines incident from i, will now be redefined as the positive sum:

$$\lambda_{ii} = \sum_{j \neq i} \lambda_{ij}$$

If for some reason it should become desirable to use Coates' formula—though it will not throughout the remainder of this paper—it is possible to recover the original

"true" flowgraph by changing the sign of every loop weight from positive to negative.

Evaluation By Complete Enumeration

In this section and the next, two systematic procedures for calculating graph determinants will be presented. Both methods apply to modified transition diagrams whose weights are regarded as positive, in the sense discussed just above. The first method follows directly from the restatement of theorem 5.2 in the form of corollary 5.2a.

Let H^k denote a 1-factor which contains k non-trivial cycles.

Corollary 5.2a. Under the same assumptions as theorem 5.2,

$$\Delta_G = f'(H^0) - \sum_{H^1} f'(H^1) + \ldots (-1)^{N/2} \sum_{H^{N/2}} f'(H^{N/2})$$

Proof: The formula is equivalent to the one in theorem 5.2 with all 1-factors containing the same number of non-trivial cycles collected. The only possible surprises are the absence of a summation sign before the first term, and the number of terms included. The former is accounted for by the fact that there is only one 1-factor which contains no non-trivial cycles, namely, the one consisting entirely of loops. The number of terms is determined by the fact that the maximum number of non-trivial cycles is N/2, achieved when all the non-trivial cycles are of minimum length, which is two.

Corollary 5.2a suggests an orderly procedure for evaluating graph determinants.

1. Erase (or ignore throughout) the signs of all weights; deal only with their absolute values.

2. Calculate $f'(H^0)$, the product of all loop weights. Set $\Delta_G = f'(H^0)$.

3. Find and list all non-trivial cycles. For each, calculate $f'(H^1)$, the product of the weight of the cycle and the weights of all loops at points not in the cycle. As each is calculated, subtract it from Δ_G.

4. Now using the list of all non-trivial cycles, consider all pairs of non-trivial cycles. For each <u>disjoint</u> pair, calculate $f'(H^2)$, the product of the weights of the two cycles and the weights of all loops at points not touched by either cycle. Add each such $f'(H^2)$ to Δ_G.

5. Using the list of all non-trivial cycles and the disjoint pairs determined above, find all triples of disjoint cycles. Calculate each $f'(H^3)$ and subtract from Δ_G.

6. Continue in this manner for k = 4,5,6,...until no disjoint k-tuplet of non-trivial cycles can be found. By the corollary, k cannot exceed N/2. Whenever k is even, add to Δ_G; when odd, subtract from Δ_G.

7. The final Δ_G is the correct value of Δ_G.

It is not claimed that this algorithm is either elegant or efficient, but it does provide a reliable, orderly way of going about the job. It also provides a basis for

obtaining approximate values of graph determinants. This point will be investigated further in chapter 6.

Evaluation By Balancing

The second method for evaluating graph determinants depends upon successive application of a "balancing" step. Before the algorithm can be described, a number of definitions and two theorems must be stated.

Let G be the entire modified transition diagram of some system. Let S be any subgraph of G, such as arises from G by removing a path. For any point x of S, λ_{xx} denotes the (positive) weight of the loop at x, equal in value to the sum of the weights of lines leaving x in the original unmodified transition diagram. Of course, it does not necessarily hold that λ_{xx} is equal to the sum of the weights of lines leaving x in S. Let δ_{xx} denote the sum of the weights of lines leaving x in S. We will say that a point x is in balance in S if $\lambda_{xx} = \delta_{xx}$. A graph S is said to be balanced if every point of S is in balance in S.

Theorem 5.3. The graph determinant of a balanced graph is zero.

Proof: Upon changing the signs of the loop weights, such a graph is the flowgraph of a singular matrix, since it corresponds to a matrix in which the sum of every row is zero.

If S is a graph such that every point is in balance, then the graph determinant is automatically zero and need not be calculated. Because of the way in which the λ_{xx} were defined—that is, such that the original transition diagram of the process would be in balance—many of the points of S may be in balance. It is impossible, however, for all of the points of S to be in balance; for in order for this to happen, S would have to contain all of the lines that the original transition diagram did, including the ones which enter the zero state. But these lines were removed to obtain the modified transition diagram.

In general, then, S will contain several points which are not in balance. Let θ_{xx} represent the amount of imbalance of the point x, or $\lambda_{xx}-\delta_{xx}$. By the definitions of λ_{xx} and δ_{xx}, θ_{xx} is non-negative, and is strictly greater than zero if x is not in balance.

Theorem 5.4. If S is a subgraph of a modified transition diagram and x is a point which is not in balance in S, then

$$\Delta_S = \Delta_{S'} + \theta_{xx}\Delta_{S-x}$$

where S' is the graph obtained from S by changing the weight of the loop at x from λ_{xx} to δ_{xx}.

Proof: Partition the 1-factors of S into those which contain the loop at x and those which do not. The former set has the total value $\lambda_{xx}\Delta_{S-x}$. This can be rewritten as the sum of $\delta_{xx}\Delta_{S-x}$ and $\{\lambda_{xx}-\delta_{xx}\}\Delta_{S-x}$. The first of these summands, $\delta_{xx}\Delta_{S-x}$, combined with the 1-factors of S which do

not contain the loop at x, comprise the 1-factors of S'. The second summand is just $\theta_{xx}\Delta_{S-x}$. The equation follows directly. Figure 5-2 shows visually the logical steps of the proof.

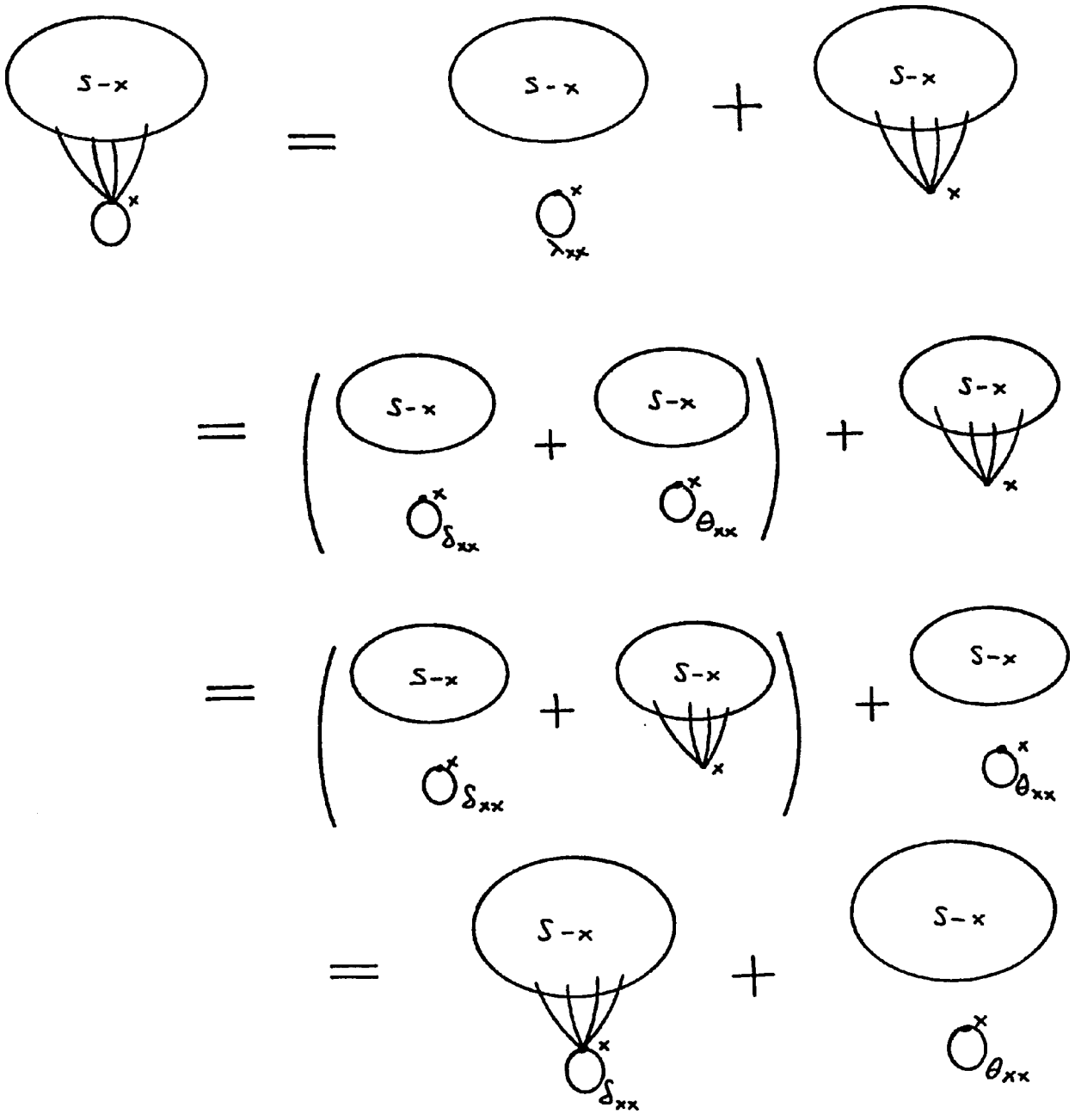


Fig. 5-2. The proof of theorem 5.4.

In words, theorem 5.4 says that the graph determinant of S can be written as the sum of the graph determinant of S with the point x brought into balance and the graph determinant of S-x multiplied by the amount of imbalance of x. By applying this "balancing" step successively to each point of S which is not in balance, S is eventually brought completely into balance. At this point, the graph determinant of S is zero, so S itself may be dropped from further consideration. At each balancing step, however, a new subgraph containing one less point than S was generated. Now each of these can be reduced in turn to a set of graphs containing one fewer point; each of these can be similarly reduced; and so forth. This procedure will be referred to as the balancing algorithm.

Theorem 5.5. If S contains n points, the balancing algorithm will require at most n! balancing steps.

Proof: Let S be any subgraph of a modified transition diagram having n points, k of which are not in balance. Of course, k cannot exceed n. Then by a single balancing step, S is reduced to two graphs, one of which contains n-1 points, some of which may not be in balance. The number of points not in balance in this graph may or may not exceed k, but is certainly no greater than n-1. The other graph contains n points, but only k-1 may be out of balance. Those which are out of balance in this graph were also out of balance in S. That is, the balancing step did not "unbalance" any points which were already in balance.

Now the balancing step can be applied to the second of these graphs to get another graph containing n-1 points (of which at most n-1 may be out of balance) and another graph containing n points, of which only k-2 are out of balance. Continuing in this manner, we obtain, after exactly k balancing steps, k graphs containing n-1 points, and one graph containing n points, none of which is out of balance. By theorem 5.3, this last graph may be disregarded.

By the same line of reasoning, each of the graphs containing n-1 points may be reduced to a series of graphs containing n-2 points. The number of steps required for any one graph will be equal to the number of points out of balance. Similarly, each graph containing n-2 points can be reduced to a series of graphs containing n-3 points, and so on. Since the number of out of balance points in any graph is at worst equal to the total number of points in the graph, the total number of steps required is at most $(n)(n-1)(n-2)\ldots(1) = n!$.

Summary Of The Method

Before considering an example, let us review in the light of presently available theorems the whole approach to the problem of finding steady state probabilities for finite Markovian queueing networks. The method is as follows:

(a) Set up the transition diagram of the process.

(b) Modify it by adding to each point a loop of weight equal to the sum of the weights of all lines leaving the point, and afterwards deleting all lines entering the empty state.

(c) Solve for the $c^*(x)$, for every state x, using the flowgraph equation

$$c^*(x) = \sum_{P_{0x}} f'(P_{0x}) \, \Delta_{G_0 - P_{0x}}$$

To evaluate the graph determinant $\Delta_{G_0 - P_{0x}}$, use either the complete enumeration algorithm or the balancing algorithm.

(d) Determine q_0^* from the normalizing equation,

$$\sum_{\text{all } x} c^*(x) q_0^* = 1$$

(e) The final solution is given by

$$q_x = c^*(x) q_0^*, \text{ for each state } x$$

It is the evaluation of the graph determinants which is likely to prove most difficult. When the graph $G_0 - P_{0x}$ is very large, complete enumeration of the 1-factors may be out of the question. Even the balancing algorithm may require too many separate balancing steps to be practical. In such a case there are two alternatives. The first is to break the graph into separate smaller graphs whose graph determinants can more easily be calculated. Chapter 6 will display a number of ways in which this may be done. The

second is to use approximations. Chapter 7 develops a number of these.

An Example

Taking a particularly simple and familiar example to illustrate the solution technique, consider the M/M/1 queue with finite storage capacity. Let N denote the maximum capacity. The modified transition diagram for this system appears in figure 5-3.

To obtain the relative coefficient, c*(i), for a general state i, it is necessary to find all paths from 0 to i. It is obvious that in this case there can be only one such path, and the weight of that path, $f'(P_{0i})$, is just ρ^i. The remaining subgraph, G_0-P_{0i}, is of the form shown in figure 5-4. The graph determinant of this graph is easily evaluated using the balancing algorithm.

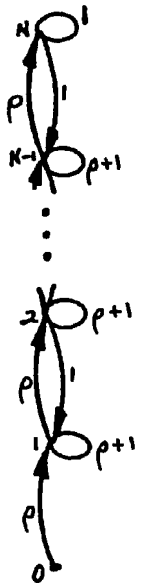


Fig. 5-3. Modified transition diagram.

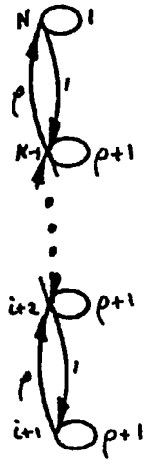


Fig. 5-4. G_0-P_{0i}.

The only point not in balance is the one at the bottom. Balancing this point, one obtains a balanced graph, whose determinant is zero, and the graph shown in figure 5-5. From this one step, it is immediately obvious that the balancing algorithm can be applied repeatedly to obtain the single graph shown in figure 5-6. Then, by inspection, $\Delta_{G_0-P_{0i}} = 1$.

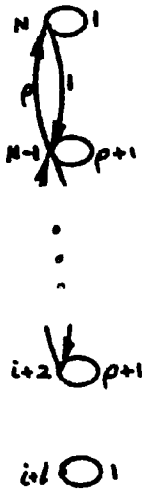


Fig. 5-5. One balance step.



Fig. 5-6. Many balance steps.

Since the same argument applies for all i, $c^*(i) = f'(P_{0i})\Delta_{G_0-P_{0i}} = \rho^i$, for all i from 0 to N. This is an unnormalized solution to the steady-state equations (which, of course, were never directly formulated). To obtain the true probabilities, q_i, we use the normalizing equation. The final result is

$$q_i = \rho^i \left(\frac{1-\rho}{1-\rho^{N+1}}\right)$$

CHAPTER 6

DECOMPOSITION

To aid in dealing with moderately large-scale modified transition diagrams, which can arise from even small-scale queueing systems, it would be advantageous to have available a variety of methods by which to decompose the graphs. "Decomposition" is used here in a general sense to refer to the broad concept of separating a graph into smaller pieces, solving the pieces independently, and compiling the results (adjusting as necessary to correct for the effect of decomposition) to obtain the complete solution. As will be demonstrated in this chapter, there are many ways to decompose a graph.

One method—balancing—has already appeared in the guise of a graph determinant evaluation algorithm. A graph with only a single point not in balance can be rewritten as some multiple of the graph determinant of a graph with one less point (because the balanced graph can be ignored). Similarly, a graph with, say, five points not in balance can be rewritten as a weighted sum of five separate graph determinants, where each of the five graphs whose determinants are left to be calculated is smaller,

by one point, than the original graph. So the complete balancing algorithm is, in fact, an evaluation procedure which is based upon repeated decomposition.

A second very general method for decomposing a graph to simplify the evaluation of the graph determinant depends upon separating the graph into a set of subgraphs called "strongly connected components." When applicable, this form of decomposition should always be used, because the graph determinants of the subgraphs can be multiplied together without correction to obtain the graph determinant of the original graph .

A third very general method is always applicable, but may or may not be convenient, since it does necessitate correction terms. Basically, the theorem, theorem 6.2, expresses the effect of separating any single point from the rest of the graph. We will find use for theorem 6.2 in proving other theorems in chapter 7 and in the appendix to chapter 7. Theorem 6.3 simply generalizes theorem 6.2; it expresses the effect of separating an arbitrary set of points from the rest of the graph.

Finally, a group of decomposition theorems which apply only when a graph contains a specific kind of point, a "cutpoint;, will be proven. Chapter 8 will illustrate the use of these theorems.

Strongly Connected Components

A directed graph is strongly connected if for any two points x and y of G there is a path in G from x to y and there is also a path from y to x. A strongly connected component of a directed graph G is a maximal strongly connected subgraph of G. Maximal is used in the sense that there is no other strongly connected subgraph of G which properly contains it. Figure 6-1 illustrates a directed graph which is not strongly connected. The strongly connected components of this graph are shown separated in figure 6-2.

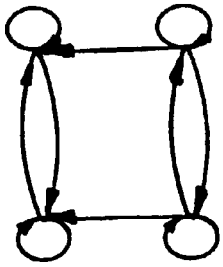


Fig. 6-1. A digraph G.

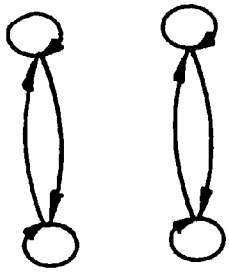


Fig. 6-2. The strongly connected components.

It is an interesting sidelight that the property of strong connectedness in the transition diagram of a Markov process corresponds exactly to irreducibility of the process (because paths correspond to sequences of transitions). Recognizing this fact, it may appear at first glance—since it was assumed at the outset that the processes to be discussed would be irreducible—that it is unnecessary to consider any flowgraphs which are not strongly connected. That conclusion would be in error, however, for although the

original transition diagram of an irreducible process is strongly connected, the modified transition diagram need not be. In fact, because the modification removes all lines incident to the zero point, one may be certain that the modified transition diagram is not strongly connected. Moreover, the solution procedure involves the evaluation of graph-determinants for subgraphs obtained by removing paths from the modified transition diagram, and these subgraphs need not be strongly connected.

Theorem 6.1. A graph determinant is the product of the graph determinants of its strongly connected components.

Proof: This theorem was proved by Harary in [14]. Details of the proof may be found there.

Point Separation

Theorem 6.2 makes explicit the effect of separating one distinguished point from the rest of the graph. In principle, it provides a method by which any graph can be reduced to arbitrarily small size. There may, of course, be better ways to accomplish the same result.

For convenience in stating and proving theorem 6.2, let $H_S^{k;x}$ denote a 1-factor of S (or whatever graph is named in the subscript) which contains k non-trivial cycles, one of which contains the point x.

Theorem 6.2. For any point x in S, a modified transition diagram or a subgraph thereof,

$$\Delta_S = \lambda_{xx}\Delta_{S-x} - \sum f'(H_S^{1;x}) + \sum f'(H_S^{2;x}) - \ldots + (-1)^{n/2} \sum f'(H_S^{n/2;x})$$

Proof: Starting from the evaluation formula of corollary 5.2a,

$$\Delta_S = f'(H^0) - \sum f'(H^1) + \sum f'(H^2) - \ldots$$

partition all of the 1-factors into two groups: those which contain a loop at point x and those which do not. In particular, H^0 is a 1-factor belonging to the former group. If H^i is a 1-factor belonging to the latter group, rewrite it as $H_S^{i,x}$, since the point x must be included in one of the non-trivial cycles of H^i. If H^i is a 1-factor belonging to the first group, rewrite $f'(H^i)$ as $\lambda_{xx}f'(H_{S-x}^i)$, since H^i contains the loop at point x and the rest of H^i must be a 1-factor of S-x.

We have, then, by grouping terms,

$$\Delta_S = \{\lambda_{xx}f'(H_{S-x}^0) - \sum \lambda_{xx}f'(H_{S-x}^1) + \ldots\} + \{-\sum f'(H_S^{1,x}) + \sum f'(H_S^{2,x}) - \ldots\}$$

But, factoring λ_{xx} out of terms in the first group, what is left is just the graph determinant of S-x. Hence,

$$\Delta_S = \lambda_{xx}\Delta_{S-x} - \sum f'(H_S^{1,x}) + \sum f'(H_S^{2,x}) - \ldots$$

and the theorem is proved.

An analogous, but more general, theorem can be stated and proved in a similar way. First, however, some new terminology and notation must be defined.

A complete subgraph, T, of S consists of a subset Y of the points of S and every line of S between points of Y. It is the "every line" requirement which distinguishes a complete subgraph from an ordinary subgraph. For the following theorem, let S be a modified transition diagram or a subgraph thereof, let T be a complete subgraph of S, and let R be the graph S-T. By the definition of S-T, R must be complete. Let the notation $H_S^{k;rt}$ mean a 1-factor of S which contains exactly k non-trivial cycles, at least one of which includes both a point of R and a point of T.

Theorem 6.3. If S, T, and R are as described above,

$$\Delta_S = \Delta_T\Delta_R - \sum f'(H_S^{1;rt}) + \sum f'(H_S^{2;rt}) - \ldots$$

Proof: Partition the 1-factors of S into those which are such that every cycle is contained wholly within either T or R, and those which include cycles having points in both T and R. By the definition of the terminology, the latter set is represented by the correction terms. It remains to show that the former set is equivalent to $\Delta_T\Delta_R$. Since none of these 1-factors of S contains any line between a point of T and a point of R, remove these lines from S. But the resulting graph is not connected; it consists of the two components T and R. If T and R are not even weakly connected, there cannot be a strongly connected component which contains points of both T and R. The required equivalence follows directly from theorem 6.1.

Cutpoint Decomposition

A cutpoint of a digraph D is a point whose removal separates D into two or more components; that is, a point x such that D-x is not a connected digraph. If a modified transition diagram, or a subgraph thereof, contains a cutpoint, the evaluation of graph determinants can be simplified through the use of theorem 6.4. Notice in particular that a subgraph of a modified transition diagram may contain a cutpoint even if the original modified transition diagram does not. Theorem 6.4 could be proved as a corollary to theorem 6.3, but the proof presented here will be independent of that result.

If x is a cutpoint of S, then S-x consists of two or more components. Let R be one of these components, and let T represent the graph consisting of the rest of the components of S-x. That is, T = S-x-R.

Theorem 6.4. For S a modified transition diagram or a subgraph thereof, and x a cutpoint of S,

$$\Delta_S = \Delta_{R+x}\Delta_T + \Delta_R\Delta_{T+x} - \lambda_{xx}\Delta_R\Delta_T$$

Proof: The graph determinant of S is a sum of the weights of all 1-factors of S. The first term on the right hand side represents the total weight of all 1-factors of S which do not contain any cycle consisting of x and points of T. The second term represents the weight of those 1-factors of S which do not contain any cycle consisting of x and points of R. (See figure 6-3.)

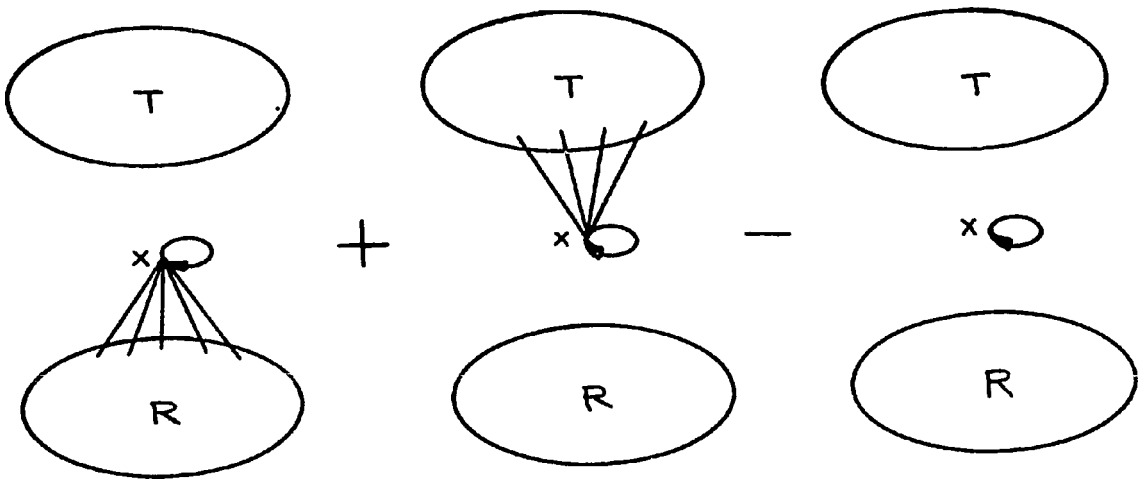


Fig. 6-3. Determinant decomposition at a cutpoint.

The final term is a correction factor, compensating for the fact that those 1-factors which contain a loop at x have been counted twice. Because x is a cutpoint, there are no 1-factors other than those already mentioned; no cycle may contain points of both R and T.

Theorem 6.4 provides a way to simplify the evaluation of graph determinants. But when a modified transition diagram contains a cutpoint, other kinds of simplification also are possible. Theorem 6.5 shows that paths can be decomposed.

For theorem 6.5, S must contain the zero point, and R must be the component of S-x (where x is a cutpoint of S) which contains the zero point.

Theorem 6.5. For any point y is R,

$$\sum_{P_{0y} \text{ in } S} f'(P_{0y}) = \sum_{P_{0y} \text{ in } R} f'(P_{0y}),$$

and for any point y in T,

$$\sum_{P_{0y} \text{ in } S} f'(P_{0y}) = \left(\sum_{P_{0x} \text{ in } R} f'(P_{0x})\right)\left(\sum_{P_{xy} \text{ in } T} f'(P_{xy})\right)$$

Proof: If y is in R, no path from 0 to y may contain any points of T, because once T has been reached, the cutpoint x must have been crossed. There can be no way to return to R. Hence the set of paths from 0 to y in S is just the set of paths from 0 to y in R.

If y is in T, every path from 0 to y must necessarily include the point x. Thus every path can be considered to consist of a path segment from 0 to x and a path segment from x to y. The first must lie entirely within R; the second, entirely within T. To <u>any</u> path segment from 0 to x, P_{0x}, <u>every</u> possible path segment from x to y may be appended to produce a distinct path from 0 to y. Thus the total weight of all those paths from 0 to y which begin with the path segment P_{0x} is $f'(P_{0x})\sum f'(P_{xy})$. Summing over all possible path segments from 0 to x gives the total weight of all paths.

By itself, theorem 6.5 is not of great help, because the determination of the c*(y) does not require only the sum of path weights. However, this result is useful in proving theorems 6.6 and 6.7, which are of direct aid in calculating the c*(y).

Again, let x be a cutpoint of S, and let R be the component of S-x which contains the zero point. Let $c^*_S(y)$ denote the relative coefficient of y in the graph S (or whatever graph is named in the subscript). That is, $c^*_S(y)$ is to be calculated in S as if S were the entire modified transition diagram.

Theorem 6.6. For all y in R,

$$c^*_S(y) = c^*_{R+x}(y)\Delta_T + c^*_R(y)\{\Delta_{T+x} - \lambda_{xx}\Delta_T\}$$

Proof:

$$c^*_S(y) = \sum_{P_{0y} \text{ in } S} f'(P_{0y})\Delta_{S-P_{0y}}$$

But if y is in R,

$$c^*_S(y) = \sum_{P_{0y} \text{ in } R+x} f'(P_{0y})\Delta_{S-P_{0y}}$$

and by theorem 6.4,

$$c^*_S(y) = \sum_{P_{0y} \text{ in } R+x} f'(P_{0y})[\Delta_{(R+x)-P_{0y}}\Delta_T + \Delta_{R-P_{0y}}\Delta_{T+x} - \lambda_{xx}\Delta_{R-P_{0y}}\Delta_T]$$

Rewriting,

$$c^*_S(y) = \Delta_T \sum_{P_{0y} \text{ in } R+x} f'(P_{0y})\Delta_{(R+x)-P_{0y}} + \{\Delta_{T+x} - \lambda_{xx}\Delta_T\} \sum_{P_{0y} \text{ in } R} f'(P_{0y})\Delta_{R-P_{0y}}$$

$$= \Delta_T c^*_{R+x}(y) + \{\Delta_{T+x} - \lambda_{xx}\Delta_T\} c^*_R(y).$$

Corollary 6.6a. Under the assumptions of theorem 6.6,

$$c^*_S(x) = c^*_{R+x}(x)\Delta_T$$

Proof: $c^*_R(x) = 0$ because there is no path from 0 to x in R alone. Applying theorem 6.6, the corollary follows directly.

Although it may not be obvious that theorem 6.6 actually simplifies the calculation of $c^*_S(y)$, for y in R, there is no doubt that corollary 6.6a does reduce the effort required to find $c^*_S(x)$, for x a cutpoint. Theorem 6.7 will provide an equally attractive formula for $c^*_S(y)$ when y is in T.

Although $c^*_S(y)$ and $c^*_R(y)$ are well defined, an explanation of the terminology $c^*_{T+x}(y)$ is required, because T+x does not contain the zero point. Let $c^*_{T+x}(y)$ be the relative coefficient of y in T computed as if x were the zero point. Formally,

$$c^*_{T+x}(y) = \sum_{P_{xy}} f'(P_{xy})\Delta_{(T+x)-P_{xy}}.$$

Theorem 6.7. For any y in T,

$$c^*_S(y) = c^*_{R+x}(x)c^*_{T+x}(y)$$

Proof: Each path in S from 0 to y can be expressed as a path segment in R+x from 0 to x and a path segment in T+x from x to y. Similarly, the 1-factors of $S-P_{0y}$ consist of a set of cycles lying wholly within R+x and a set wholly within T+x. Because any path from 0 to x and any 1-factor of $(R+x)-P_{0x}$ can be combined with any independently chosen path from x to y and any 1-factor of $(T+x)-P_{0y}$ to obtain a path from 0 to y and a 1-factor of $S-P_{0y}$,

$$f'(P_{0y})\Delta_{S-P_{0y}} = f'(P_{0x})\Delta_{R+x-P_{0x}}\, f'(P_{xy})\Delta_{T+x-P_{xy}}$$

and by the respective definitions,

$$c^*_S(y) = c^*_{R+x}(x)c^*_{T+x}(y)$$

In words, theorem 6.7 reduces the calculation of $c^*_S(y)$, for any point y not in the same component of S-x as the zero point, to the separate calculation of the relative coefficient of the cutpoint x in the graph R+x and of the relative coefficient of y in the graph T+x, using x as the zero point. Separately, the graphs R+x and T+x are each smaller than S.

Inasmuch as these cutpoint decomposition theorems will be used extensively in chapter 8, they will not be illustrated here.

CHAPTER 7

APPROXIMATION

This chapter is concerned with approximate solutions to the equations of equilibrium. A distinction should be made between the kind of approximation made after a solution is known for the purpose of simplifying the result (Morse, for example, in [24], uses such approximations extensively), and the kind which is made before the exact solution is known to avoid the effort of exact calculation. It is the latter kind which are discussed here. Of course, such approximations are of value only if they are much easier to obtain than exact solutions. The theorems of this chapter supply a variety of easily applied approximation techniques.

There are basically two classes of techniques presented here. The first simplifies the calculation of graph determinants by, in essence, providing information about the error resulting from allowing the evaluation to be only partially carried out. Although the relative coefficients must still be calculated, the effort required to do so is reduced. Moreover, the theorems apply with complete generality.

The second class of techniques applies only to the category of queueing networks called parallel systems. For these systems, however, the techniques permit direct approximations to the relative coefficients and to the steady state probabilities. These techniques are particularly convenient because the approximations are obtained merely by looking at the transition diagram. No calculation whatsoever is required.

Not because it is unimportant, but rather for the sake of clarity in the presentation, a portion of the theory relating to the second class of approximations has been separately placed in an appendix to the chapter. The theorems presented there are quite significant in their own right.

Approximation by Truncation

As was mentioned in chapter 5, the graph determinant evaluation algorithm which depends upon the complete enumeration of the 1-factors—the one suggested by corollary 5.2a—lends itself to approximation by truncation. The basis for truncating is contained within the next two theorems.

The notation H^k has been used to denote any 1-factor which contains k non-trivial cycles. For the purposes of the next two theorems, let H^{k+1} denote any 1-factor which contains the same non-trivial cycles as H^k plus one additional one. We will say that H^{k+1} is derived from H^k.

Theorem 7.1. Let S be any subgraph of a modified transition diagram. Then for all k,

$$f'(H^k) \geq f'(H^{k+1})$$

where H^k is any 1-factor of S containing k non-trivial cycles and H^{k+1} is derived from H^k.

Proof: Since the only difference between H^k and H^{k+1} is that H^{k+1} has a cycle through points where H^k has loops, the difference $f'(H^k) - f'(H^{k+1})$ must be the difference between the product of certain loop weights and the weight of a cycle through the same points. Suppose for concreteness that three points, x, y, and z, are involved in this difference, as in figure 7-1.

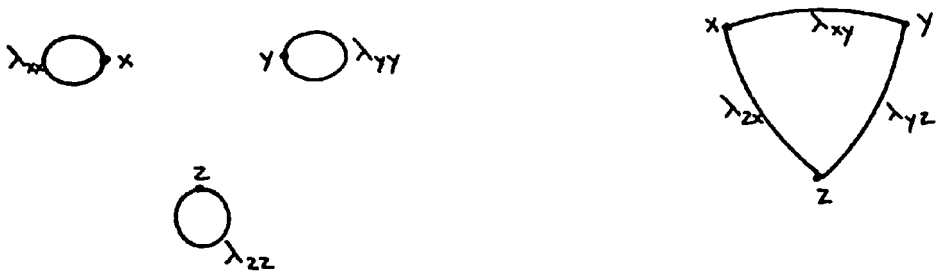


Fig. 7-1. The difference between H^k and H^{k+1}.

In this case, the above difference is given by

$$f'(H^k) - f'(H^{k+1}) = \lambda_{xx}\lambda_{yy}\lambda_{zz} - \lambda_{xy}\lambda_{yz}\lambda_{zx}$$

(Recall that we are now treating loop weights as if they were positive.) But each loop weight equals the sum of all weights of lines leaving the point, so

$$\lambda_{xx} \geq \lambda_{xy}$$

$$\lambda_{yy} \geq \lambda_{yz}$$

$$\lambda_{zz} \geq \lambda_{zx}$$

Since all weights are positive,

$$\lambda_{xx}\lambda_{yy}\lambda_{zz} \geq \lambda_{xy}\lambda_{yz}\lambda_{zx}$$

Hence the difference $f'(H^k) - f'(H^{k+1})$ is non-negative, and $f'(H^k) \geq f'(H^{k+1})$. It is obvious that the theorem is equally valid for cases in which the number of points involved in the difference is other than three.

Now let us relate this theorem to the process of evaluating Δ_S by complete enumeration of the 1-factors. The procedure involved finding, first, the unique H^0, then all of the H^1, then the H^2, and so forth. Each H^{k+1} is derived from some H^k. By theorem 7.1, knowledge of $f'(H^k)$ tells us something in advance about the contribution that $f'(H^{k+1})$ can make to Δ_S.

Unfortunately, theorem 7.1 does not directly imply that for all k,

$$f'(H^k) \geq \sum f'(H^{k+1})$$

for the simple reason that the right hand sum may contain more terms than the left hand sum. If this inequality <u>were</u> true, we could truncate the series of corollary 5.2a,

$$\Delta_S = f'(H^0) - \sum f'(H^1) + \sum f'(H^2) - \ldots$$

anywhere, with the assurance that the error would be no greater than the first omitted sum (because the signs alternate and the sums diminish).

Even though theorem 7.1 does not directly imply it, the inequality above is correct, as shown by the next theorem.

Theorem 7.2. Under the assumptions of theorem 7.1, for all k,

$$f'(H^k) \geq \sum f'(H^{k+1})$$

Proof: For each H^{k+1} on the right hand side, there is at least one H^k on the left hand side (in fact, there are k+1) such that the specified H^{k+1} is derivable from it. It will therefore certainly suffice to show that for each H^k,

$$f'(H^k) \geq \sum f'(H^{k+1}), \qquad (7.1)$$

where now the right hand sum is over all H^{k+1} which can be derived from H^k.

Now H^k contains k non-trivial cycles and a number of loops. Let R denote the subgraph of S consisting of these k non-trivial cycles. Each of the H^{k+1} in the sum contains the same k non-trivial cycles, or R, and one additional cycle over some or all of the points of S-R. In other words, each H^{k+1} consists of R and an H^1 of S-R. Therefore, the inequality (7.1) can be rewritten, for k>0, as

$$f'(R)f'(H^0_{S-R}) \geq \sum f'(R)f'(H^1_{S-R}),$$

or,

$$f'(H^0_{S-R}) \geq \sum f'(H^1_{S-R}) \tag{7.2}$$

For k=0, the inequality (7.1) is the same as (7.2) without any manipulation. So to prove (7.2)—and thereby the theorem—it will be sufficient to show that for any S,

$$f'(H^0_S) \geq \sum f'(H^1_S) \tag{7.3}$$

where now the sum is over all H^1_S, because every H^1_S is derived from H^0_S.

To prove the inequality (7.3) in complete generality requires terminology so complicated that it tends to obscure the logic. For this reason, the proof will first be given for a particular example. Subsequently, it will be generalized. Suppose, then, that S consists of three points, as in figure 7-2.

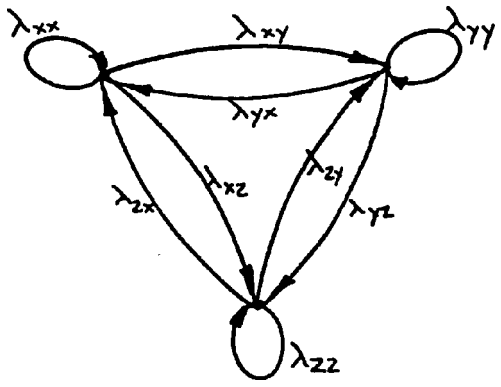


Fig. 7-2. The graph S.

All possible lines have been included and have been given weights of general form. Every weight between

distinct points must be non-negative, but we will permit some of them to be of value zero to allow for the cases in which the graph does not contain all possible lines. The loop values must be strictly positive.

Using this example, what has to be proved is that the weight of the 1-factor shown in figure 7-3 is no less than the sum of the weights of the 1-factors shown in figure 7-4.

Algebraically, it must be shown that

$$\lambda_{xx}\lambda_{yy}\lambda_{zz} \geq \lambda_{xx}\lambda_{yz}\lambda_{zy} + \lambda_{yy}\lambda_{xz}\lambda_{zx} + \lambda_{zz}\lambda_{xy}\lambda_{yx} + \lambda_{xy}\lambda_{yz}\lambda_{zx} + \lambda_{xz}\lambda_{zy}\lambda_{yx} \quad (7.4)$$

The loop weight λ_{xx} is the sum of the weights of all lines leaving the point x in the original transition diagram. Let θ_{xx} be the sum of the weights of lines leaving x which do not appear in S. (This notation is consistent with the previous use of θ as the amount of imbalance of a point.) Then

$$\lambda_{xx} = \lambda_{xy} + \lambda_{xz} + \theta_{xx}$$

Defining θ_{yy} and θ_{zz} similarly,

$$\lambda_{yy} = \lambda_{yx} + \lambda_{yz} + \theta_{yy}$$

$$\lambda_{zz} = \lambda_{zx} + \lambda_{xy} + \theta_{zz}$$

Substituting, the left hand side of (7.4) becomes

$$(\lambda_{xy} + \lambda_{xz} + \theta_{xx})(\lambda_{yx} + \lambda_{yz} + \theta_{yy})(\lambda_{zx} + \lambda_{zy} + \theta_{zz}) \quad (7.5)$$

If this were to be completely multiplied out, the result would be a sum of 27 terms, among which would be the 11 following terms:

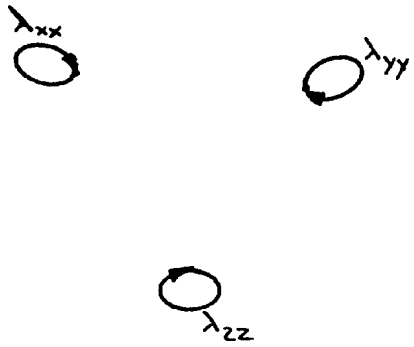


Fig. 7-3. H^0.

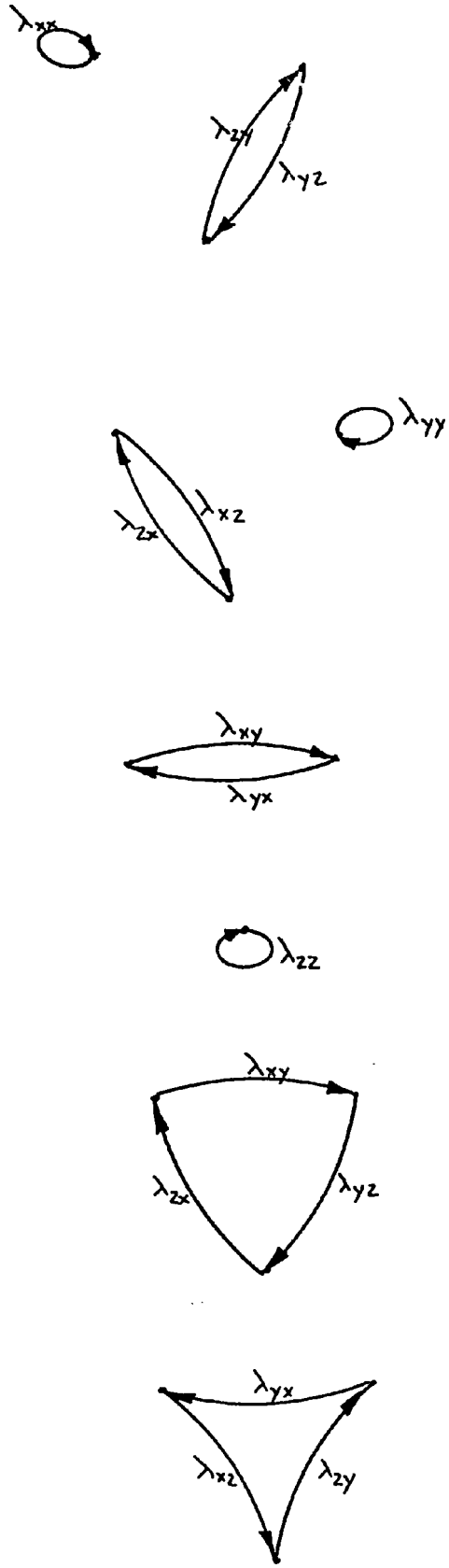


Fig. 7-4. All H^1.

$$(\lambda_{xy}\lambda_{yz}\lambda_{zy}+\lambda_{xz}\lambda_{yz}\lambda_{zy}+\theta_{xx}\lambda_{yz}\lambda_{zy}) \tag{7.6}$$
$$+ (\lambda_{xz}\lambda_{yx}\lambda_{zx}+\lambda_{xz}\lambda_{yz}\lambda_{zx}+\lambda_{xz}\theta_{yy}\lambda_{zx})$$
$$+ (\lambda_{xy}\lambda_{yx}\lambda_{zx}+\lambda_{xy}\lambda_{yx}\lambda_{zy}+\lambda_{xy}\lambda_{yx}\theta_{zz})$$
$$+ (\lambda_{xy}\lambda_{yz}\lambda_{zx})$$
$$+ (\lambda_{xz}\lambda_{yx}\lambda_{zy})$$

The complete product is obtained by selecting in every possible way one term from each of the sums in (7.5). It is easily verified that each of the 11 terms of (7.6) can be obtained in that way. Rewriting (7.6) after factoring,

$$(\lambda_{xy}+\lambda_{xz}+\theta_{xx})\lambda_{yz}\lambda_{zy} + (\lambda_{yx}+\lambda_{yz}+\theta_{yy})\lambda_{xz}\lambda_{zx}$$
$$+ (\lambda_{zx}+\lambda_{zy}+\theta_{zz})\lambda_{xy}\lambda_{yx} + \lambda_{xy}\lambda_{yz}\lambda_{zx} + \lambda_{xz}\lambda_{zy}\lambda_{yx}$$

and substituting back,

$$\lambda_{xx}\lambda_{yz}\lambda_{zy}+\lambda_{yy}\lambda_{xz}\lambda_{zx}+\lambda_{zz}\lambda_{xy}\lambda_{yx}+\lambda_{xy}\lambda_{yz}\lambda_{zx}+\lambda_{xz}\lambda_{zy}\lambda_{yx}$$

we obtain the entire right hand side of (7.4). So the left hand side of (7.4) consists of all of the terms which appear on the right hand side and 16 additional terms, all of which are non-negative. Hence the inequality is valid, and the theorem is proved for this particualr example.

Proceeding now to the general case, suppose S contains n points. Without loss of generality, let them be numbered from 1 through n. Then it must be shown that

$$\prod_{i=1}^{n}\lambda_{ii} \geq \sum_{\sigma}\prod_{i=1}^{n}\lambda_{i\sigma(i)} \tag{7.7}$$

where the right hand sum is over all permutations σ of n elements containing exactly one cycle.

Consider first the left hand side of (7.7). Writing the loop weight λ_{ii} as done in the case for n=3

$$\lambda_{ii} = \sum_{\substack{j=1 \\ j\neq i}} \lambda_{ij} + \theta_{ii}$$

For notational convenience, make a temporary substitution of λ_{ii} for θ_{ii}. For this argument only, λ_{ii} will represent the amount of imbalance, rather than the whole loop weight. The loop weight can then be written very simply as

$$\sum_{j=1}^{n} \lambda_{ij}$$

The left hand side of (7.7) becomes

$$\prod_{i=1}^{n} \sum_{j=1}^{n} \lambda_{ij} \tag{7.8}$$

This is a product of sums. To write (7.8) as a sum of products, multiply it out. Any one of the terms in the sum will be a product of the form

$$\lambda_{1.}\lambda_{2.} \cdots \lambda_{n.} \tag{7.9}$$

where a dot as a second subscript represents a number from 1 to n. Notice that because a $\lambda_{i.}$ is selected from each of the sums in (7.8) in turn, each number from 1 to n appears exactly once as a first subscript.

How many terms of the form (7.9) could possibly exist? Because the second subscript of $\lambda_{1.}$ could be chosen in n ways, the second subscript of $\lambda_{2.}$ could also be chosen in

n ways, and so forth, the number of terms which could be constructed in the form (7.9) is n^n. But because for every i, the sum $\sum_{j=1}^{n} \lambda_{ij}$ contains every number from 1 to n as a second subscript, the complete expansion of (7.8) must contain every one of the n^n conceivable terms of the form (7.9).

Because we require only the weak inequality (7.7), it will suffice to show that the right hand side can be written as a sum of terms of the form (7.9); for if it can, the left hand side must be at least as great because it is the sum of all possible terms of this form.

Consider one of the terms in the right hand sum of (7.7). It is of the form

$$\lambda_{1\sigma(1)}\lambda_{2\sigma(2)} \cdots \lambda_{n\sigma(n)} \tag{7.10}$$

where σ is a permutation having exactly one cycle. But some of these weights may be loop weights, namely, those for which $\sigma(i)=i$. Without loss of generality suppose the first k weights are loop weights. Again using λ_{ii} to represent the amount of imbalance, rather than the whole loop weight, (7.10) can be written

$$\left(\sum \lambda_{1j}\right)\left(\sum \lambda_{2j}\right) \cdots \left(\sum \lambda_{kj}\right) \lambda_{k+1,\sigma(k+1)} \cdots \lambda_{n\sigma(n)}$$

It is apparent that when this expression is multiplied out, it will consist of a sum of products, each one of which is of the form (7.9).

Because the argument applies to each term in the sum on the right hand side of (7.7), the entire right hand side

is a sum of terms of the form (7.9). This is all that was required, so the inequality (7.7) has been proved, and the proof of theorem 7.2 is complete.

Corollary 7.2a. For S any subgraph of a modified transition diagram, $\Delta_S > 0$.

Proof: Using corollary 5.2a,

$$\Delta_S = f'(H^0) - \sum f'(H^1) + \sum f'(H^2) - \sum f'(H^3) + \ldots$$

compare the first two terms, the next two, and so forth. By theorem 7.2, $f'(H^0) - \sum f'(H^1)$ is non-negative $\sum f_i(H^2) - \sum f'(H^3)$ is non-negative, and so forth. So Δ_S cannot be negative. It cannot equal zero unless S is balanced, but no subgraph of a modified transition diagram is balanced.

Corollary 7.2a, along with the observation that path weights are always positive, provides assurance that the c*(x) are all positive, as they should be to obtain a probability distribution from them by normalizing.

Approximations For Parallel Systems

The approximation theorems of this section will apply only to the class of parallel systems for which all servers have equal mean service time. Attention is restricted to this class in order that a single parameter, the traffic intensity, will suffice to describe all of the stochastic processes which constitute the entire system. Let ρ be the traffic intensity—that is, the average number of arrivals

to the system per unit of mean service time. The measurement of time in units of mean service time will imply that the parameter of the negative exponential distribution describing the service time (for any server) is unity.

Although these restrictions appear to be more restrictive than is necessary, they permit clear, simple, and attractive results because there is only one parameter, ρ. In addition, they <u>are</u> broad enough to cover many interesting systems, including those to be solved in chapters 8 and 9.

Let us first establish what the transition diagrams of such systems look like. Since they are parallel systems, the diagrams will contain only upward and downward arrows—no horizontal arrows. Since upward arrows represent arrivals, each will be labeled with some constant times ρ. In the case of a deterministic switching rule which sends the system into a different state for each kind of arrival, each upward arrow will be labeled with just ρ. In the general case, however, if k different arrivals each send the system into the same state, the arrow corresponding to that transition will have the label $k\rho$; or if a probabalistic rule dictates that, given a particular arrival, the probability that the system is sent into a particular state is p, then the arrow corresponding to that transition will have the label $p\rho$. Since downward arrows represent departures, each will be labeled with just some number. Restated, no

upward arrow is labeled with anything other than a multiple of ρ, and no downward arrow is labeled with anything other than a constant.

Because the flowgraph of a parallel system contains no horizontal lines, the non-trivial cycles in such a graph must contain an even number of lines. In fact, they must contain an equal number of upward and downward lines. This fact is obvious if one thinks of a cycle as a closed path. In order to return to the starting point, every upward step in the path must be matched by a downward step, and vice versa. The product of the weights of lines in a non-trivial cycle will, of course, be some coefficient times ρ^k, where k is the number of upward lines in the cycle. Stated another way, it will be a polynomial in ρ of degree equal to half the length of the cycle. Similarly, the product of weights of lines in a path will be some coefficient times ρ^k, where k is the number of upward arrows in the path.

The loop value at any point x, being equal to the sum of the weights of all lines leaving x, is given by

$$\lambda_{xx} = \alpha(x)\rho + \beta(x)$$

where $\alpha(x)\rho$ is the sum of the weights of lines going upward from x and $\beta(x)$ is the sum of lines going downward from x.

Any state x for which $\alpha(x) = 0$ is called a <u>blocking state</u>; if an item attempts to enter the system when it is in such a state, it will be rejected. Although one could construct a queueing system in which departures are

forbidden while the system is in certain states, it would be artificial to do so. Thus $\beta(x) \neq 0$ except when x is the empty state. But since the weight of the loop at the zero state is never used (in fact, it is removed when all lines entering the zero state are), we may say that for all non-blocking states,

$$\lambda_{xx} = \alpha(x)\rho + \beta(x), \quad \alpha(x),\beta(x) \neq 0$$

and for all blocking states,

$$\lambda_{xx} = \beta(x)$$

We next establish that the form of $c^*(x)$ for these systems will be a polynomial in ρ. This is readily apparent from the definition of $c^*(x)$, page 63, and the realization that each term of the sum is just a product of weights involving only ρ and constants. Since it is a polynomial, let us use the notation

$$c^*(x) = c_0^*(x) + c_1^*(x)\rho + \ldots + c_R^*(x)\rho^R$$

or

$$c^*(x) = \sum_{r=0}^{R} c_r^*(x)\rho^r$$

Now we wish to know how much can be said about this polynomial without calculating. The first theorem of this section will enable us to know, merely by looking at the transition diagram, the smallest r for which $c_r^*(x)$ is not zero, for any state x. Later, the largest value of r for which $c_r^*(x)$ is not zero (i.e., the degree of the polynomial $c_r^*(x)$) will be given by a similar theorem.

These theorems are significant in themselves as approximations under special traffic conditions. For example, in very light traffic (i.e., ρ close to zero), it is the smallest power of ρ which essentially determines $c^*(x)$. They may also be used as checks on the accuracy of detailed calculation of the $c^*(x)$. Finally, they contribute to our understanding of how states of a particular system relate to one another.

Assuming, then, that our discussion is limited to that class of parallel systems described above—those which can be described in terms of the single parameter ρ—we have the following theorem:

Theorem 7.3. The smallest r for which $c_r^*(x)$ does not equal zero is equal to the number of arrival transitions (upward arrows) in a shortest path from zero to x.

Proof: Let P_{0x} be any path from 0 to x, and suppose it contains t lines, r of which are upward. Then $f'(P_{0x})$ is a constant multiple of ρ^r. Consider now the determinant of the remaining subgraph. Because each loop contains a constant term, the lowest power of ρ in $\Delta_{G_0-P_{0x}}$ is zero, contributed by $f'(H^0)$. Hence the lowest power of ρ in $f'(P_{0x})\Delta_{G_0-P_{0x}}$ is r, the number of upward lines in P_{0x}. The smallest r, over all paths from 0 to x, is obtained by a shortest path from 0 to x.

Special cases and applications of this theorem will be demonstrated later in this chapter.

The next major theorem will apply to the same parallel systems as the last, but with the additional requirement that the system have only one blocking state. The shorter-queue, instantaneous jockey, and overflow examples all satisfy this additional requirement.

The unique blocking state must necessarily appear at the top of the transition diagram, because it is the only state from which no lines go upward. Notice that it is not assumed that for all but one state any arrival is acceptable. The assumption is that for all but one state there is at least one kind of arrival which is acceptable.

If discussion is limited to such systems, the following theorem will be true.

Theorem 7.4. If x is any state at level k, the largest r for which $c_r^*(x)$ does not equal zero (i.e., the degree of the polynomial $c^*(x)$) is N-L+k, where N is the total number of states not counting the zero state, and L is the number of levels.

Proof: The proof of this theorem is rather involved. Rather than digress as far as would be necessary from the principal objectives of this chapter, let the truth of theorem 7.4 be granted. Details of the proof may be found in the appendix to this chapter.

Theorems 7.3 and 7.4 determine, respectively, low and high traffic approximations to the relative coefficients, the set of which constitutes an unnormalized solution to the steady state equations. Approximations to the

true state probabilities can be obtained by, in effect, normalizing the approximations to the relative coefficients. Theorem 7.5 does this for those parallel systems having a single blocking state.

Theorem 7.5. For $\rho<<1$ (low traffic), q_x is of the order of ρ^j, where j is the number of upward transitions in a shortest path from 0 to x. For $\rho>>1$ (high traffic), q_x is of the order of ρ^{k-L}, where L is the total number of levels and k is the level of state x.

Proof: From (5.10),

$$q_x = c^*(x)q_0^*$$

and from (5.11),

$$q_0^* = \frac{1}{\sum_{x=0}^{N} c^*(x)}$$

So

$$q_x = \frac{c^*(x)}{\sum_{x=0}^{N} c^*(x)}$$

Since for small ρ, $\rho^{i+1}<<\rho^i$, q_x is essentially determined by the lowest powers of ρ in the numerator and denominator of this quotient. The lowest power in the numerator is given by theorem 7.3 as j. By the same theorem, the lowest power in the denominator is 0, contributed to the sum by $c^*(0)$. Hence the lowest power of ρ in the quotient is as the theorem states.

For large ρ, $\rho^{i+1}>>\rho^i$, and it is the highest powers of ρ that are of interest. By theorem 7.4, the highest

power in the numerator is N-L+k. In the denominator, the highest power of ρ is contributed to the sum by the relative coefficient of the blocking state. According to theorem 7.4, this power is N. Hence the highest power of ρ in the quotient is as the theorem states, and the theorem is proved.

In addition to providing approximations which are directly useful, theorem 7.5 gives a precise indication of the relation between the probabilities of states at various positions in the transition diagram. One can say, for example, that under conditions of heavy traffic, states at the same level have steady-state probabilities of the same order of magnitude. In light traffic, however, this statement may not hold true.

On the other hand, theorem 7.5 suggests how to characterize those queueing systems which do have the property that, even in light traffic, states at the same level have steady-state probabilities of the same order of magnitude. The class consists of those parallel, single-blocking-state systems for which there is a direct path, consisting only of arrivals, from the zero state to every state x. To such systems, the following corollary applies.

Corollary 7.5a. For $\rho \ll 1$, q_x is of the order of ρ^k, where k is the level of state x.

Proof: The number of upward transitions in a shortest path from zero to x is equal to the level of state x.

Not every parallel, single-blocking-state system is of the type described above. The overflow queue, for example, is not. (See figure 3-3, page 29.) This case will be considered later.

A familiar example of a system to which this corollary does apply is the finite capacity M/M/1 queue. We have seen in chapter 5 that the exact relative coefficients are given by $c^*(k)=\rho^k$ for i=0,1,...N. Here k refers to the state "k customers in the queue" and N refers to the maximum possible number of customers in the queue. However, referring to the transition diagram, figure 5-3, we see that k also corresponds to the level of the state, and N is the total number of states not counting the empty state, and also the number of levels. Thus, using theorems 7.3 and 7.4, we could have determined without knowledge of the exact solution that the highest and lowest powers of ρ in the polynomial $c^*(k)$ coincide; that is, that $c^*(k)$ consists of the single term ρ^k (possibly multiplied by some coefficient.) Using corollary 7.5a, we could have determined that for small ρ, q_k was of the order of ρ^k and for large ρ, q_k was of the order of ρ^{k-N}. This is confirmed by the exact solution.

APPENDIX TO CHAPTER 7

This appendix is presented with the objective of proving theorem 7.4 and thereby demonstrating in detail the use of flowgraph concepts in a theoretical context. In order to facilitate the proof, a number of lemmas will first be stated and proved. These lemmas are of interest in themselves.

Lemma 7A.1. If S is a subgraph of an absolute flowgraph of a parallel system, and S does not contain a blocking state, then

$$\deg\{f'(H^k)\} < \deg\{f'(H^{k+1})\},$$

where $\deg\{f'(H^k)\}$ denotes the degree of the polynomial of any 1-factor H^k of S containing k non-trivial cycles, and H^{k+1} is any 1-factor of S which contains the same non-trivial cycles as H^k plus one additional one.

Proof: As in theorem 7.1, the only difference between H^k and H^{k+1} is that H^{k+1} has a cycle where H^k has loops. Without loss of generality, assume that four points are involved in this difference, as in figure 7A-1.

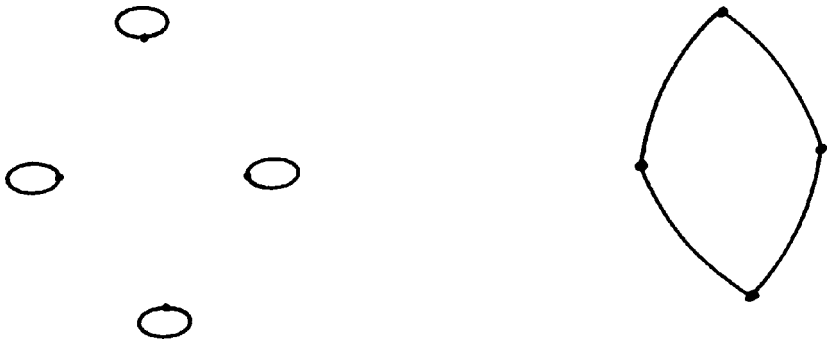


Fig. 7A-1. The difference between H^k and H^{k+1}.

Let r denote the degree of the polynomial which is the product of all of the weights of lines which H^k and H^{k+1} have in common. Then, since none of the points are blocking, the degree of the weight of each loop must be 1, and therefore $f'(H^k)$ must be of degree r+4. Now the degree of $f'(H^{k+1})$ is equal to r plus the degree of the product of the weights of the lines in the above cycle. Since we are dealing with parallel systems there must be an equal number of upward and downward transitions in a non-trivial cycle, and hence the degree of the product must in this case be 2. It should be clear from this example that if the cycle which appears in H^{k+1} but not in H^k contains p points (p must be even and greater than zero), then the degree of $f(H^k)$ will be r+p and the degree of $f(H^{k+1})$ will be r+(p/2). In short, $\deg\{f'(H^k)\} > \deg\{f'(H^{k+1})\}$.

The significant point of this lemma is that the inequality is strict. The weak inequality $\deg\{f'(H^k)\} \geq \deg\{f'(H^{k+1})\}$ follows by contradiction from theorem 7.1; for if $\deg\{f'(H^k)\} < \deg\{f'(H^{k+1})\}$, then for sufficiently large ρ, $f'(H^k) < f'(H^{k+1})$, which contradicts theorem 7.1. The strict inequality is required to prove lemma 7A.2.

Lemma 7A.2. If S is a subgraph of the modified transition diagram of a parallel system, and S does not contain a blocking state, then the degree of Δ_S is equal to the number of points contained in S.

Proof: By lemma 7A.1,

$$\deg\{f'(H^0)\} > \deg\{f'(H^1)\} > \cdots$$

so the degree of Δ_S is equal to the degree of $f'(H^0)$. But H^0 is just a set of loops, each having a weight of degree 1. The product of weights has degree equal to the sum of the degrees, which is just the number of points.

Neither lemma 7A.1 nor lemma 7A.2 is necessarily true if S contains the blocking point. Suppose, for example, that H^k is a 1-factor of S which contains a loop at the blocking point and H^{k+1} is a 1-factor which is identical to H^k, except that it has a cycle of length two containing the blocking point. Figure 7A-2 illustrates the difference between the two 1-factors.



Fig. 7A-2. The difference between H^k and H^{k+1}.

If r is the degree of the product of weights of lines that H^k and H^{k+1} have in common, then $f'(H^k)$ must be of degree r+1, because one of the above points is the blocking point, whose loop is of degree zero. But $f'(H^{k+1})$ is also of degree r+1, because the cycle of length two contains exactly one upward line. Therefore, $\deg\{f'(H^k)\} = \deg\{f'(H^{k+1})\}$ for this particular pair of 1-factors, and a counter-example has been found to the conjecture that lemma 7A.1 is true when S contains the blocking point.

One can, however, say that the case mentioned is the only one in which the degrees can possibly be equal. If, say, four points were involved in the difference between H^k and H^{k+1}, instead of just two, then an argument similar to the one just given would prove that $\deg\{f'(H^k)\} = r+3$ and $\deg\{f'(H^{k+1})\} = r+2$. Or if the blocking point were not one of the two points involved in the difference, then the argument of lemma 7A.1 would prove that $\deg\{f'(H^k)\} > \deg\{f'(H^{k+1})\}$.

But if lemma 7A.1 is not true when S contains a blocking point, then neither is lemma 7A.2. For in spite of the fact that $\deg\{f'(H^k)\} \geq \deg\{f'(H^{k+1})\}$ under any circumstances, and the equality holds only in the circumstances just described, one cannot say that $\deg\{\Delta_S\} = \deg\{f'(H^0)\}$. It is possible that the leading coefficient in the polynomial $f'(H^0)$ will be cancelled out by the sum of the leading coefficients of the $f'(H^1)$. See, for example, figure 7A-3, in which the state y is blocking.

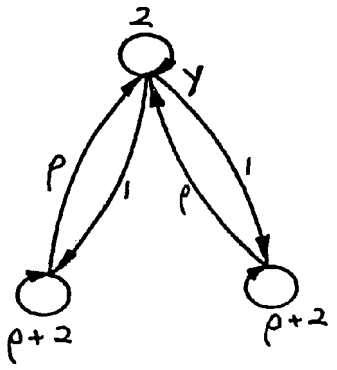


Fig. 7A-3. A graph S.

$f'(H^0)$ in this example is of degree 2, as illustrated graphically in figure 7A-4.





Fig. 7A-4. H^0.

But figure 7A-5 illustrates the two $f'(H^1)$, both of which are also of degree 2.



Fig. 7A-5. The two H^1.

In fact, when Δ_S is calculated, we find that because the terms of degree 2 cancel, Δ_S is only of degree 1.

$$\Delta_S = f'(H^0) - \sum f'(H^1)$$
$$= 2\rho^2 + 8\rho + 8 - \rho^2 - 2\rho - \rho^2 - 2\rho$$
$$= 4\rho + 8$$

With this example, the need for lemma 7A.3 is established.

Lemma 7A.3. If S contains the blocking state x, then the degree of Δ_S is n-L+k-1, where n is the number of points in S, L is the total number of levels in the transition diagram, and k is the level of the highest unbalanced point in S.

Proof: The proof will be by induction on k.

First, suppose the blocking state is not in balance. Then k=L, and it must be shown that $\deg\{\Delta_S\} = n-1$. This can be done by separating the blocking state x from the rest of the graph, using theorem 6.2.

$$\Delta_S = \lambda_{xx}\Delta_{S-x} - f'(H_S^{1;x}) + \ldots$$
$$= \beta(x)\Delta_{S-x} - f'(H_S^{1;x}) + \ldots$$

But Δ_{S-x} does not contain the blocking state, and therefore has degree equal to the number of points in S-x, which is n-1. Provided that some of the terms in the correction factor do not cancel the coefficient of ρ^{n-1} in $\beta(x)\Delta_{S-x}$, this part of the theorem will be proved. That the coefficient of ρ^{n-1} is <u>not</u> cancelled can be proved immediately, but it will be more instructive to go on to the next case—a case in which cancellation does occur. It will then be apparent that under the conditions of this first case, complete cancellation could not occur.

For the second case, suppose that the highest unbalanced point is in level L-1. Then it must be shown that $\deg\{\Delta_S\} = n-2$. To verify that the degree is at least n-2, apply the

balancing step to y, as given in theorem 5.4.

$$\Delta_S = \Delta_{S'} + \theta_{yy}\Delta_{S-y}$$

Because y is the highest unbalanced point in S, the amount of imbalance must be a sum of weights of <u>downward</u> lines. In other words, θ_{yy} must be just a constant. By corollary 7.2a, both $\Delta_{S'}$ and Δ_{S-y} are positive, so the degree of Δ_S must be at least equal to the degree of Δ_{S-y}. But the graph S-y fits the description of case 1; that is, the blocking point is unbalanced. So the degree of Δ_{S-y} equals the number of points in S-y less one. That number is n-2. We have so far shown that $\deg\{\Delta_S\} \geq n-2$.

Equality will be proved if it can be shown that the degree of S'—recall that S' is the graph which is identical to S except that the point y is in balance—is no greater than n-2. To prove this, separate the blocking point x from the rest of the graph

$$\Delta_{S'} = \beta(x)\Delta_{S'-x} - \sum f'(H^{1;x}_{S'}) + \cdots$$

From this expression, it is clear that the degree of $\Delta_{S'}$ is at most equal to the degree of $\Delta_{S'-x}$, which by lemma 7A.2 is n-1. But this is not enough.

The coefficient of ρ^{n-1} in $\beta(x)\Delta_{S'-x}$ can be stated exactly. It is

$$\beta(x) \prod_{z \text{ in } S'-x} \alpha(z)$$

because, from lemma 7A.1, the highest power of ρ in $\Delta_{S'-x}$ comes from H^0, the set of loops. Now in the sum $f'(H^{1;x}_{S'})$, there are some terms of degree n-1; namely, those weights

of 1-factors, $H_S^{1;x}$, which contain cycles of length two containing the blocking point. From the discussion earlier, these are the only terms of degree n-1.

So it is sufficient to look at the terms

$$\sum_{y_i \text{ in level } L-1} \lambda_{xy_i} \lambda_{y_i x} f'(H^0_{S'-x-y_i})$$

And of these terms, it is sufficient to look at the leading coefficients

$$\sum_{y_i \text{ in level } L-1} \left(\lambda_{xy_i} \alpha(y_i) \prod_{z \text{ in } S'-x-y_i} \alpha(z)\right) \rho^{n-1}$$

In writing this, the uniqueness of the blocking state x was invoked to conclude that the coefficient of ρ in $\lambda_{y_i x}$ is just $\alpha(y_i)$. But this expression can be rewritten

$$\sum_{y_i \text{ in } L-1} \lambda_{xy_i} \prod_{z \text{ in } S-x} \alpha(z) \rho^{n-1}$$

$$= \left(\prod_{z \text{ in } S-x} \alpha(z)\right) \left(\sum_{y_i \text{ in } L-1} \lambda_{xy_i}\right) \rho^{n-1}$$

$$= \left(\prod_{z \text{ in } S-x} \alpha(z)\right) \beta(x) \rho^{n-1}$$

This is precisely the highest order term in $\beta(x)\Delta_{S'-x}$. Consequently, the coefficients cancel, and Δ_S, is of degree less than n-1. It is conceivable that terms of

lower order also cancel, but all that was required was that $\deg\{\Delta_{S'}\} \leq n-2$.

The argument is now complete for the first two cases. To take the induction step, assume that the theorem is true for graphs in which the highest unbalanced point is at level k+1, or higher. Let S be a graph having n points in which the highest unbalanced point, y, is in level k, where $k \leq L-2$. Balance the point y

$$\Delta_S = \Delta_{S'} + \theta_{yy}\Delta_{S-y}$$

As before, θ_{yy} is just a constant, so $\deg\{\Delta_S\} \geq \deg\{\Delta_{S-y}\}$. The graph S-y has n-1 points and is such that the highest unbalanced point is in level k+1. Applying the induction hypothesis, Δ_{S-y} is of degree $(n-1)-L+(k+1)-1 = n-L+k-1$.

The graph S' is identical to S, except that the point y is in balance. If any points other than y are out of balance in S, S' is still a graph of the class being considered. That is, the highest unbalanced point in S' is still in level k. Successively balancing all points z in level k which are not in balance in S', $\Delta_{S'}$, can be written as a weighted sum of graph determinants of the form $\Delta_{S'-z}$, where the weights are just constants, and the graph determinant of a graph which contains no out of balance points in level k. By logic similar to that previously stated, each $\Delta_{S'-z}$ is of degree n-L+k-1. The graph determinant of the remaining graph is certainly of no greater degree than this.

In summary, then, Δ_S is of degree n-L+k-1, the induction step is established, and lemma 7A.3 is proved.

The objective of this appendix is to prove theorem 7.4. The proof will be by induction on the levels. For the blocking state, the proof is relatively straightforward and does not require lemma 7A.3. For these reasons, it is stated separately as lemma 7A.4.

Lemma 7A.4. If x is the blocking state, the largest r for which $c_r^*(x)$ does not equal zero (i.e., the degree of c(X)) is equal to N, the total number of states not counting the empty state.

Proof: Using the evaluation form (5.21),

$$c(x) = \sum_{P_{0x}} f'(P_{0x}) \, \Delta_{G_0 P_{0x}}$$

consider a shortest path from zero to x. This will consist of upward transitions only; since no other state is blocking, there is such a path and any which contained downward transitions would be longer. This path will contribute ρ^L to $c^*(x)$, where L is the number of levels of the transition diagram. The remaining subgraph, $G_0 - P_{0x}$, will contain N-L points, none of which is blocking. By lemma 7A.2, $\Delta_{G_0 - P_{0x}}$ is of degree N-L. Together, then, a shortest path P_{0x} and the remaining subgraph $G_0 - P_{0x}$ will contribute to the sum a polynomial of degree N. It remains to be shown there is no path and remaining subgraph contributing a polynomial of higher degree.

Consider a path from 0 to x which is not of minimum length—one which contains one or more downward transitions.

Suppose it contains just one. Such a path must contain L+1 upward transitions, and must therefore contribute a polynomial of degree L+1. However, the path will contain L+2 points (other than the zero point), so the remaining subgraph will contain N-L-2 points, none of which is blocking. By lemma 7A.2, the remaining subgraph contributes a polynomial of degree N-L-2. Hence the product $f'(P_{0x})\Delta_{G_0-P_{0x}}$ will be of degree (L+1) + (N-L-2) = N-1. From this example, it is clear that there can be no polynomial of degree higher than N in the sum $\sum f'(P_{0x})\Delta_{G_0-P_{0x}}$. This completes the proof.

Theorem 7.4. If x is any state at level k, then the largest r for which $c_r^*(x)$ does not equal zero (i.e., the degree of c*(x)) is N-L+k, where N is the total number of states not counting the zero state, and L is the number of levels.

Proof: For the single state in level L—the blocking state—the theorem is proved by lemma B.4. Assume the theorem is true for all states in level k+1, or higher, and let x be a state in level k. There are two cases to consider.

Suppose first that a shortest path from 0 to x consists of upward transitions only. Then it must contain k lines, and the degree of $f'(P_{0x})$ must be k. But then G_0-P_{0x} must contain N-k points, and the highest unbalanced point must be in level k+1. See figure 7A-6.



Fig. 7A-6. Theorem 7.4, case 1.

By lemma 7A,3, $\deg\{G_0-P_{0x}\} = (N-k)-L+(k+1)-1 = N-L$. So the product of the path weight and the graph determinant of the remaining subgraph is of degree N-L+k, as required by the theorem.

Now considering the second case, suppose that a shortest path from 0 to x contains t downward lines. Then it must contain a total of (k+2t) lines, and the degree of $f'(P_{0x})$ must be k+t. But then G_0-P_{0x} must contain (N-k-2t) points. The highest unbalanced point could be in a level as low as k+1, if all of the downward lines in the path are below x, or as high as k+t+1, if all of the downward lines are above x. See figure 7A-7.

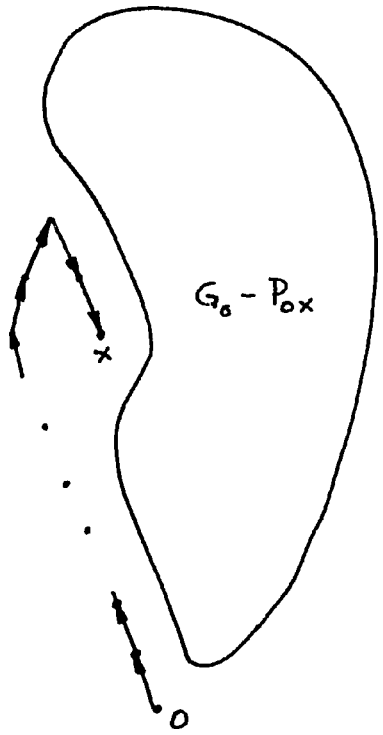


Fig. 7A-7. Theorem 7.4, case 2.

Suppose the level of the highest unbalanced state is k+t+1. Then by lemma 7A.3,

$$\deg\{\Delta_{G_0-P_{0x}}\} = (N-k-2t)-L+(k+t+1)-1 = N-L-t$$

So in this case, the product of the path weight and the graph determinant of the remaining subgraph is of degree (k+t) + (N-L-t) = N-L+k, as required by the theorem.

If the level of the highest unbalanced state is, say, only k+1, then the path weight and remaining graph determinant will be of degree N-L+k-t, which is smaller than is required. However, this poses no difficulty, because whenever a shortest path from 0 to x containing t downward lines exists, then there exists some path of equal length from 0

to x containing t downward lines such that all downward lines are above x. The existence of the latter follows from the assumption that only the topmost point is blocking and that departures are unrestricted.

It has been shown that a shortest path and the determinant of the remaining subgraph will contribute to Δ_S a polynomial of degree N-L+k. It must still be shown that paths of longer length do not contribute polynomials of higher degree. The logic required to prove this is identical to that already used. For every path from 0 to x, $f'(P_{0x})\Delta_{G_0-P_{0x}}$ is of degree at most N-L+k. Every term of this form is positive, by corollary 7.2a; there is no possibility of elimination of the highest order term by cancellation.

This concludes the proof of theorem 7.4, and the appendix.

PART III

APPLICATIONS

CHAPTER 8

THE INSTANTANEOUS JOCKEY SYSTEM

In this chapter, the two-server instantaneous jockey system will be completely solved by the methods given in chapter 5. In addition, the process of finding the solution will provide opportunities to use the approximation theorems of chapter 7 and some of the decomposition theorems of chapter 6.

The instantaneous jockey system was considered by Saaty in [25] (pp. 284-5), but was left unsolved. The solution was obtained by Koenigsberg and was presented in [20]. His results assume that both queues have infinite capacity. Since Koenigsberg's method involved solving for the probabilities in terms of q_{00}, the solution for the finite case to be considered here can be obtained from Koenigsberg's by simply performing the final normalization over a finite, instead of an infinite, sum.

The System Description

The system itself is represented in figure 8-1. Arrivals to the system enter the shorter queue, if there is one, or either queue (with equal likelihood) if they are of equal length. If a departure from one of the

servers results in a difference of queue length of more than one, one of the items in the longer queue "jockeys", or moves from the longer to the shorter queue. This change is assumed to occur instantaneously.

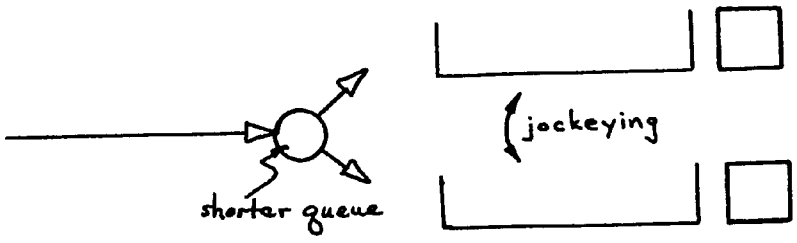


Fig. 8-1. The instantaneous jockey system.

If states are designated by the ordered pair (m,n), where m denotes the number in server A's queue, including the one in service, and n denotes the number in server B's queue, then the jockeying behavior implies that all states are of the form (m-1,m), (m,m), or (m,m-1). The maximum number in either server's queue—it is assumed that this number is the same for both queues—will be denoted by M. Thus the unique blocking state is the state (M,M).

In order that the approximation theorems be applicable, it will be assumed that the service rates are the same for the two servers. The inter-arrival and service time distributions are, of course, negative exponential. The rate of arrivals to the system will be given by 2ρ, expressed in units of arrivals per mean service time. The

use of 2ρ, instead of just ρ, is to conform to Koenigsberg's notation.

The transition diagram for the two server instantaneous jockey system was illustrated in figure 3-2 of chapter 3. The modified transition diagram, obtained by adding the loops and deleting the lines into the zero point, is shown in figure 8-2. Since the evaluation of graph determinants will use only the absolute values of weights, all weights have been recorded as positive. The loop weights have been abbreviated by defining $a=(2\rho+1)$ and $b = (2\rho+2)$.

Observe that all states of the form (i,i) are cutpoints of the graph. Thus any of the cutpoint theorems can be applied at any such point.

Define the graph B_{M-i} to be the component of $G_0-(i,i)$ which does not contain the zero point, and the graph A_{M-i} to be the graph $B_{M-i}+(i,i)$. If the convention is assumed that B_0 is a loop having weight 1, these graphs are defined for all i from 0 to M. Of course, A_M is just the entire modified transition diagram. Figure 8-3 illustrates these graphs.

For the sake of simplifying the notation, let $A_i = \Delta_{A_i}$ and let $B_i = \Delta_{B_i}$. That is, we will use the same notation for both the graph and its graph determinant, relying upon the context to distinguish the two.

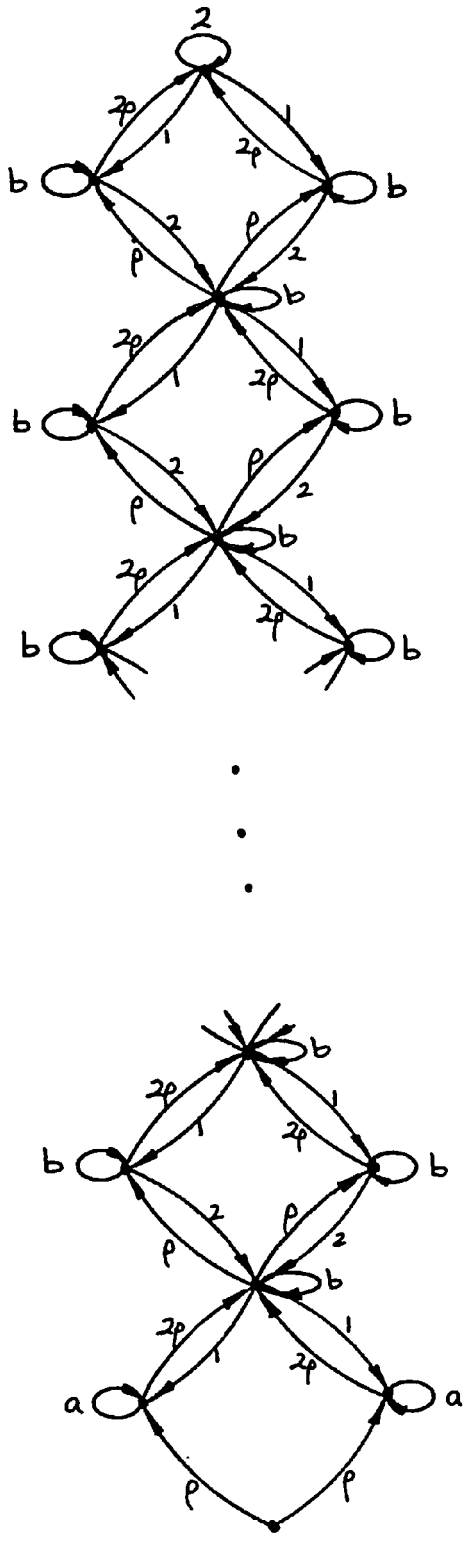


Fig. 8-2. The modified transition diagram.

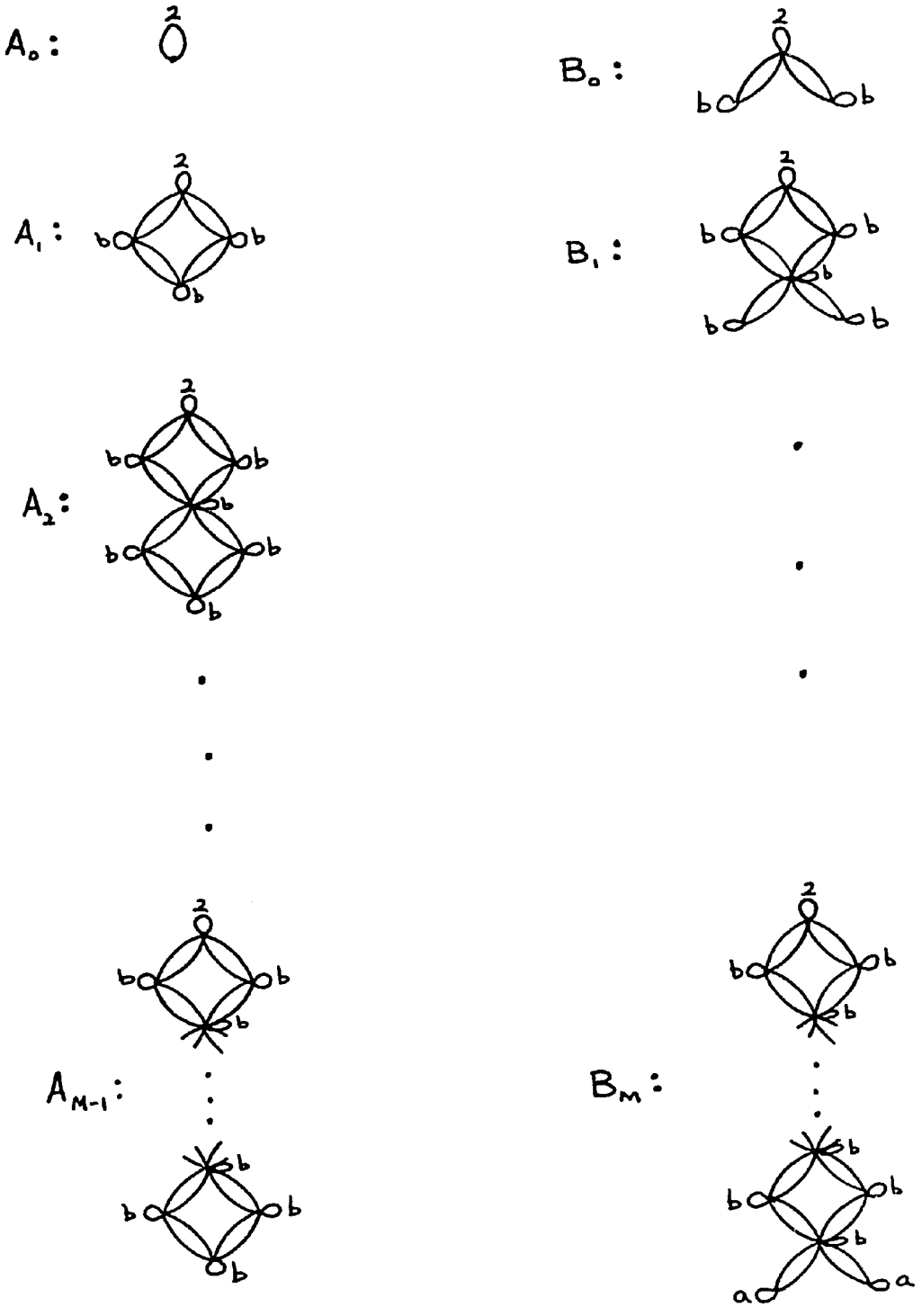


Fig. 8-3. The graphs A_i and B_i.

Theorem 8.1.

$$A_i = 2(4b)^i \quad \text{for } i=0,1,\ldots M-1$$

$$B_i = (4b)^i \quad \text{for } i=0,1,\ldots M-1$$

$$B_M = 2a(4b)^{M-1}$$

Proof: Consider any A_i. By balancing the bottom point, which is the only point not in balance, $A_i = 2B_i$, so we have only to calculate B_i.

Using the determinant decomposition method of theorem 6.4,

$$B_i = CB_{i-1} + b^2A_{i-1} - b^3B_{i-1}$$

where C is the graph shown in figure 8-4.

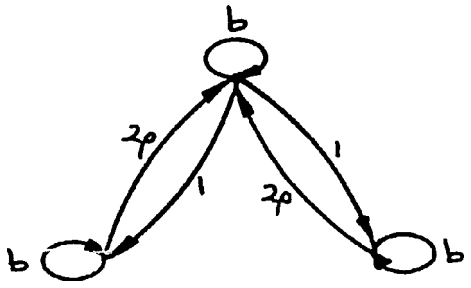


Fig. 8-4. The graph C when $i \leq M$.

Substituting values,

$$B_i = (b^3-4\rho b)B_{i-1} + 2b^2B_{i-1} - b^3B_{i-1}$$

$$= (2b^2-4\rho b)B_{i-1}$$

$$= 4bB_{i-1}$$

But by direct calculation, $B_1 = 4b$, so $B_i = (4b)^i$. This formula is correct for i=1,2,3,...M-1. For i=M, the graph C is slightly different; it is the graph shown in figure 8-5.

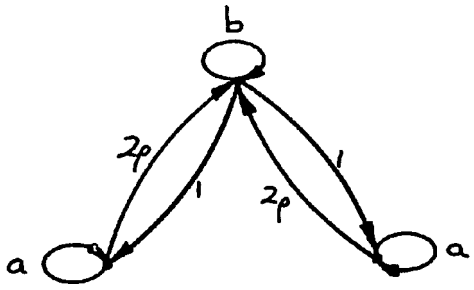


Fig. 8-5. The graph C when i=M.

Substituting values in the decomposition formula,

$$B_M = (a^2-4\rho a)B_{M-1} + 2a^2B_{M-1} - a^2bB_{M-1}$$

$$= (2a^2-4\rho a)B_{M-1}$$

$$= 2a(4b)^{M-1}$$

This concludes the proof of theorem 8.1.

Now let S_i be the graph consisting of the cutpoint (i,i) and all point below (i,i). Figure 8-6 illustrates these graphs. As the next step in the process of obtaining the relative coefficients for the graph as a whole, we wish to obtain the relative coefficient in S_i of the point (i,i).

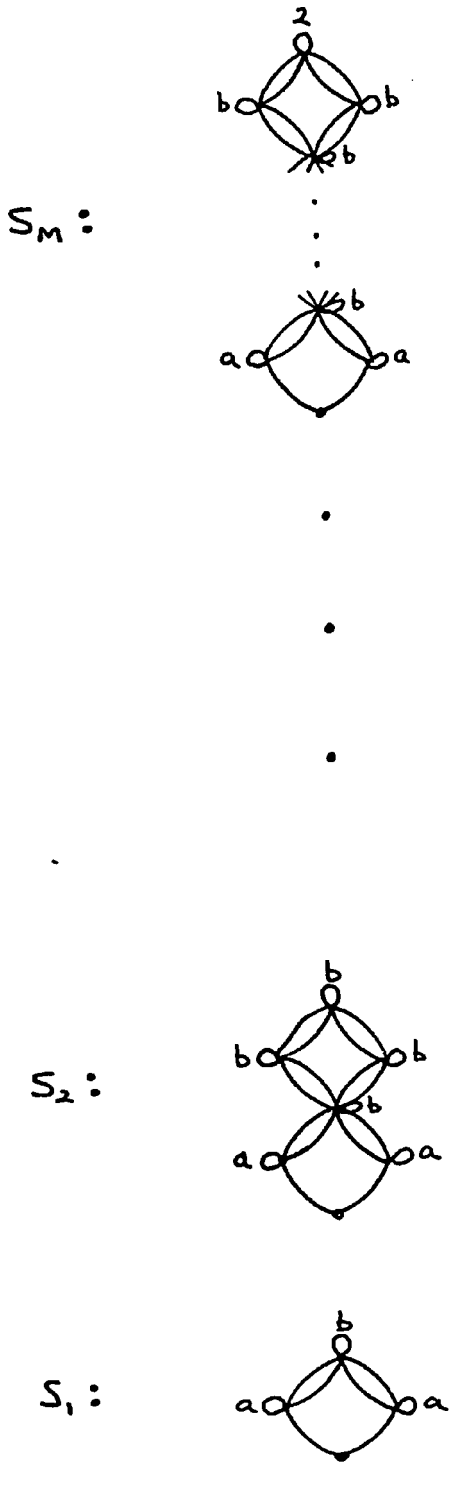


Fig. 8-6. The graphs S_i.

Theorem 8.2. Using the notation of theorem 6.6,

$$c^*_{S_i}(i,i) = (2\rho)^{2i}\, ab^{i-1}, \quad \text{for } i=1,2,\ldots M$$

Proof: Direct calculation gives

$$c^*_{S_1}(1,1) = 4\rho^2 a$$

so the theorem is true for i=1. Now assume it is true for i-1. By theorem 6.7,

$$\begin{aligned} c^*_{S_i}(i,i) &= (2\rho^2 b + 2\rho^2 b)\, c^*_{S_{i-1}}(i-1,i-1) \\ &= (4\rho^2 b)(2\rho)^{2i-2}\, ab^{i-2} \\ &= (2\rho)^{2i}\, ab^{i-1} \end{aligned}$$

The proof, by induction, is complete.

The next step is to determine the relative coefficient of a point just above a cutpoint, regarding the cutpoint as the zero point.

Theorem 8.3. Using the notation of theorem 6.7,

$$c^*_{A_{M-i}}(i+1,i) = c^*_{A_{M-i}}(i,i+1) = 2\rho b(4b)^{M-i-1} \quad \text{for } i>0$$

$$c^*_{A_M}(1,0) = c^*_{A_M}(0,1) = 2\rho a(4b)^{M-1}$$

Proof: By theorem 6.6 (and because (i+1,i+1) is a cutpoint)

$$c^*_{A_{M-i}}(i+1,i) = c^*_D(i+1,i)\, B_{M-i-1} + c^*(i+1,i)\{A_{M-i-1} - bB_{M-i-1}\}$$

where, for i>0, D is the graph shown in figure 8-7, and E is the graph shown in figure 8-8.

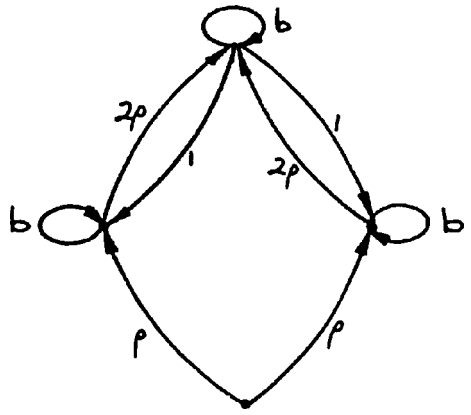


Fig. 8-7. The graph D when i>0.

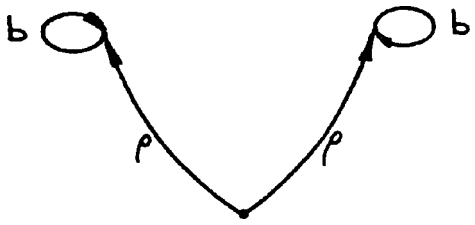


Fig. 8-8. The graph E when i>0.

By direct inspection, $c^*_E(i+1,i) = \rho b$. With only a little more calculation,

$$c^*_D(i+1,i) = \rho(b^2-2\rho) + 2\rho^2$$
$$= \rho b^2$$

Putting the results together,

$$c^*_{A_{M-i}}(i+1,i) = \rho b^2(4b^{M-i-1}) + \rho b\{2(4b)^{M-i-1} - b(4b)^{M-i-1}\}$$
$$= 2\rho b(4b)^{M-i-1}$$

In the case i = 0, the graph D is as pictured in figure 8-9 and E is as pictured in figure 8-10.

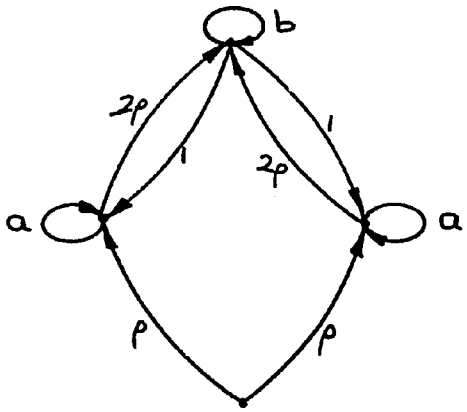


Fig. 8-9. The graph D when i=0.

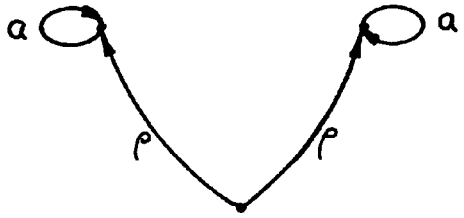


Fig. 8-10. The graph E when i=0.

The result for this case is obtained in the same manner as before:

$$c^*_E(1,0) = \rho a$$

$$c^*_C(1,0) = \rho a^2$$

$$c^*_{A_M}(1,0) = \rho ab(4b)^{M-1} + \rho a\{2(4b)^{M-1} - b(4b)^{M-1}\}$$

$$= 2\rho a\,(4b)^{M-1}$$

In every case, $c^*_{A_{M-i}}(i+1,i) = c^*_{A_{M-i}}(i,i+1)$ because of symmetry.

Now we have only to put these formulas together to solve the entire system. There are essentially three classes of states.

Case 1. States of the form (i,i). By corollary 6.6a, $c^*_G(i,i) = c^*_{S_i}(i,i)\, B_{M-i}$. We have, then, for i=1,2,...M,

$$c^*(i,i) = (2\rho)^{2i}\, ab^{i-1}\, (4b)^{M-i}$$

$$= (4b)^M \left(\frac{a}{b}\right) (\rho)^{2i}$$

for the case i=0,

$$c^*(0,0) = B_M = 2a\, (4b)^{M-1} = (4b)^M \left(\frac{a}{b}\right) \left(\frac{1}{2}\right)$$

Case 2. States of the form (i+1,i). By theorem 6.7, $c^*(i+1,i) = c^*_{S_i}(i,i)\, c^*_{A_{M-i}}(i+1,i)$. We have, then, for i=1,2,...M-1,

$$c^*(i+1,i) = (2\rho)^{2i}\, ab^{i-1}\, \rho b (4b)^{M-i-1}$$

$$= (4b)^M \left(\frac{a}{b}\right)\left(\frac{1}{2}\right)(\rho)^{2i+1}$$

For the case i=0,

$$c^*(1,0) = c^*_{A_M}(1,0) = \rho a (4b)^{M-1} = (4b)^M \left(\frac{a}{b}\right)\left(\frac{1}{2}\right)(\rho)$$

Thus the general formula works even in the case i=0.

Case 3. States of the form (i,i+1). By symmetry,

$$c^*(i,i+1) = (4b)^M \left(\frac{a}{b}\right)\left(\frac{1}{2}\right)(\rho)^{2i+1}$$

for i=0,1,...M-1.

We now have available all of the relative coefficients. Before proceeding to the normalization of these to obtain the probabilities, let us check the exact relative coefficients against the approximations to them.

According to theorem 7.4, $c^*(i,j)$ should be a polynomial in ρ of degree equal to the number of states, 3M, minus the number of levels, 2M, plus the level of the state, i+j. That is, the degree of $c^*(i,i)$ should be M+2i, and the degree of $c^*(i+1,i)$ or $c^*(i,i+1)$ should be M+2i+1. Recalling that $a=(2\rho+1)$ and $b=(2\rho+2)$, we verify that this is indeed the case.

Because the instantaneous jockey system is one such that a path consisting only of upward transitions from (0,0) to any state exists, theorem 7.3 says that the lowest power of ρ in $c^*(i,j)$ should be equal to the level, i+j. In other words, the smallest power of ρ in $c^*(i,i)$ should be 2i, and in $c^*(i+1,i)$ or $c^*(i,i+1)$ should be 2i+1. We verify that this is also the case.

Although there arise no difficulties in normalizing the relative coefficients directly, it will perhaps be more instructive to obtain first the <u>true</u> coefficients of $q_{0,0}$ (to compare them to those obtained by Koenigsberg), and to normalize the latter. The true coefficients are related to the relative coefficients by equation 5.8,

$$c(i,j) = \frac{c^*(i,j)}{c^*(0,0)}$$

Substituting the known relative coefficients,

$$c(0,0) = 1$$

$$c(i,i) = 2(2\rho)^{2i} \qquad i=1,2,\ldots M$$

$$c(i+1,i) = c(i,i+1) = (2\rho)^{2i+1} \qquad i=0,1,\ldots M-1$$

Allowing for the difference in notation, this result agrees with that of Koenigsberg.

We have only to normalize these true coefficients to obtain the final solution.

$$1 = c(0,0)q_{0,0} + \sum_{i=1}^{M} c(i,i)q_{0,0} + 2\sum_{i=0}^{M-1} c(i+1,i)q_{0,0}$$

Omitting the details of the calculation,

$$q_{0,0} = \frac{1-\rho}{1+\rho-2\rho^{2M+1}}$$

The solution, then, is given by

$$q_{i,i} = 2(\rho)^{2i}\, q_{0,0}$$

$$q_{i+1,i} = q_{i,i+1} = (\rho)^{2i+1}\, q_{0,0}$$

The approximate solution given by theorem 7.5 would be, for $\rho \ll 1$,

$$q_{00} = 1$$

$$q_{i,i} = (\rho)^{2i}$$

$$q_{i+1,i} = q_{i,i+1} = (\rho)^{2i+1}$$

and for $\rho \gg 1$,

$$q_{00} = \frac{1}{\rho^{2M}}$$

$$q_{i,i} = \frac{(\rho)^{2i}}{(\rho)^{2M}}$$

$$q_{i+1,i} = q_{i,i+1} = \frac{(\rho)^{2i+1}}{(\rho)^{2M}}$$

By direct comparison, one can see that these approximations are quite good.

CHAPTER 9

THE OVERFLOW SYSTEM

In this chapter, the two-server overflow system will be solved. Because the effort required actually to obtain the exact general solution is considerable—as will become apparent within this chapter—the solution will be carried out only to an extent sufficient to indicate the manner in which the general solution could be obtained. The exact solution will be obtained for two special cases, and an approximate general solution will be derived from the theorems of chapter 7.

The overflow problem, as considered here, was first formulated by Disney in [9]. Later, in [10], he provided some numerical results for the case in which no storage is permitted for the first server. For the general case, a two-dimensional generating function has been provided by Gupta, in [13]. Unfortunately, the generating function is so complicated that it has been inverted only in the case where no storage is permitted for the second server. A closed-form solution for the case in which the first server has no storage capacity was supplied by Disney and Solberg in [12], and certain measures of efficiency for this case were numerically compared to the same measures for

two other network designs. A closed form solution for the general case is still unknown.

The System Description

The system itself is represented in figure 9-1. An arriving customer will join the queue of the first server if and only if that queue is not filled to capacity. If the first queue is filled, the customer will "overflow", or join the queue of the second server—provided that queue is not full—and will ultimately be served by the latter. If both queues are full, the customer is lost.

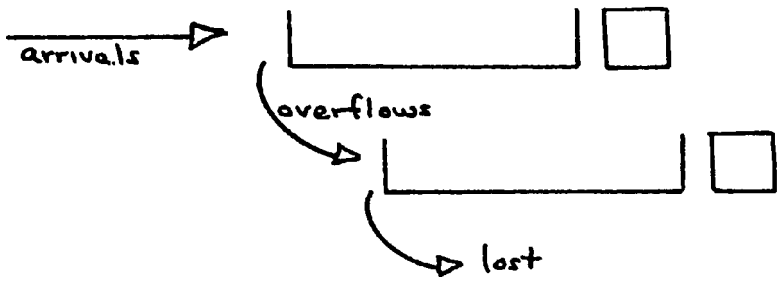


Fig. 9-1. The overflow system.

It will be assumed that the service rates are the same for the two servers. The arrival rate will be given by ρ, expressed in units of arrivals per mean service time.

States will be designated by the ordered pair (i,j), where i denoted the number in the first server's queue, including the one in service, and j denotes the number in the second server's queue. M will denote the capacity of

the first queue, and N the capacity of the second. The unique blocking state, then, will be the state (M,N).

The transition diagram for the M=3, N=4 case was illustrated in figure 3-3 of chapter 3. The modified transition diagram for the general case is shown in figure 9-2. All weights have been recorded as positive, and the substitutions $a=\rho+1$ and $b=\rho+2$ have been used for brevity.

The solution strategy will be roughly as follows: we will first identify certain subgraphs of the modified transition diagram which will appear repeatedly in the process of solution; we will then find expressions for the graph determinants of these subgraphs; we will next proceed to the determination of the relative coefficients for the graph determinants whose values are known; finally, we will determine the relative coefficients for the N=2 case.

Graph Determinants

Referring to figure 9-2, one can see that the states have been arranged in the diagram so that they might be referenced in a cartesian coordinate manner. Although the coordinate system is tilted to conform with the level convention established in chapter 3, we may still speak in terms of rows and columns of the graph. The m-th row consists of the set of states (m,j) having m as the first element; the n-th column consists of the set of states

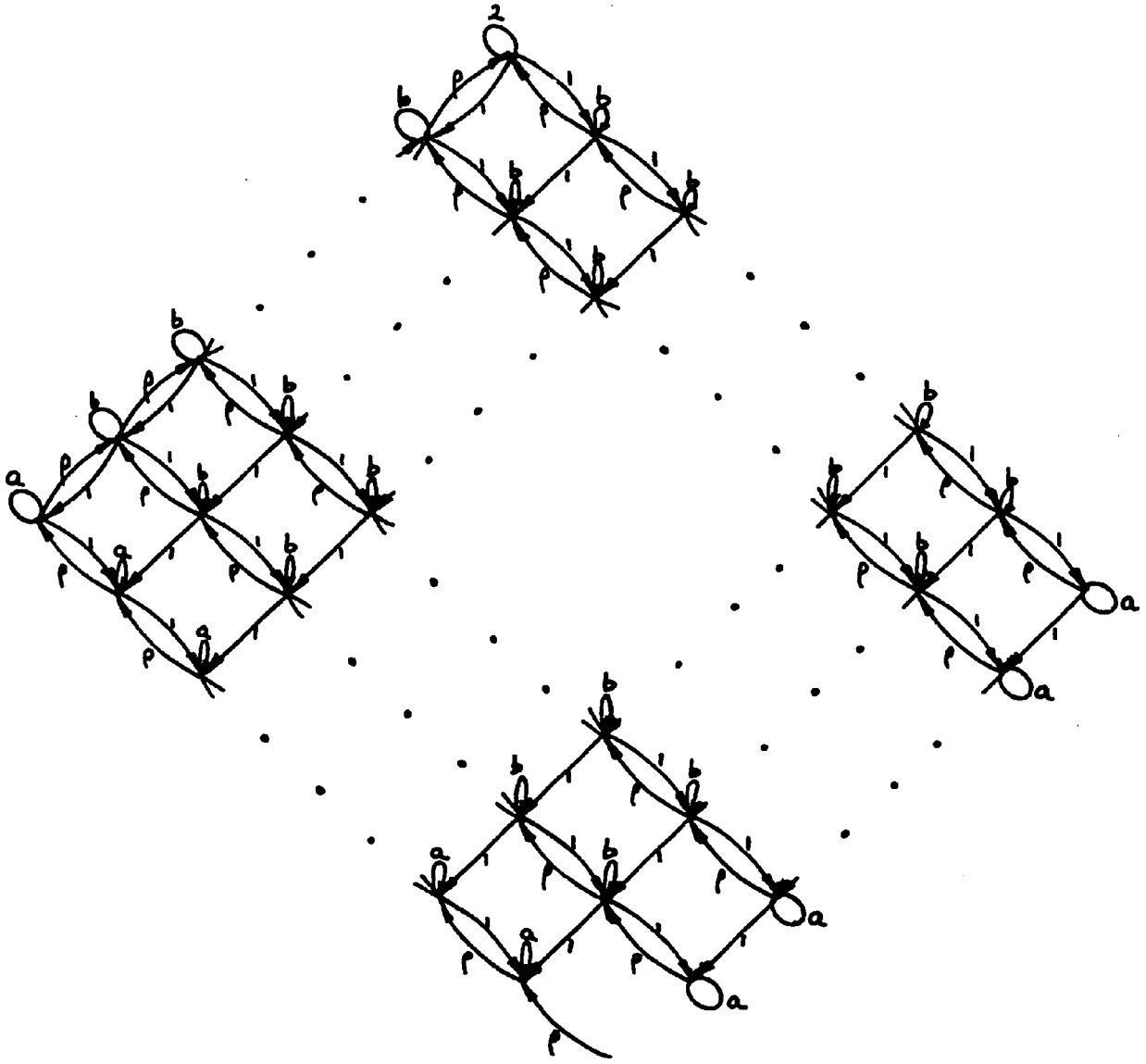


Fig. 9-2. The overflow transition diagram.

(i,n) having n as the second element. When speaking of a column as a subgraph, we mean the subgraph consisting of the points in a column, their loops, and all lines between points in that column.

Again referring to figure 9-2, notice that, with the exception of the 0-th and N-th column, all columns (as subgraphs) are identical. Notice also that, with the exception of the bottommost point—and the topmost point in the N-th column—every point within a column has the same loop weight. Let us define, then, four column types, as shown in figure 9-3. A_m is a graph which contains m points and looks like a portion of column zero in the transition diagram; B_m looks like a portion of any column except the 0-th; C_m looks like a portion of any column except the 0-th, with the bottom point included; and D_m looks like the entire N-th column.

Using recursive arguments, we will determine expressions for the values of the graph determinants of these column subgraphs. To keep the notation simple, let $A_m = \Delta_{A_m}$, $B_m = \Delta_{B_m}$, and so forth. That is, we will use the same notation for both the subgraph and its graph determinant, relying upon the context to distinguish the two.

By direct inspection, $A_1 = a = \rho+1$, and $A_2 = a^2 - \rho = \rho^2+\rho+1$. To obtain the value of A_m, use the balancing algorithm. There are two points in A_m which are not in balance:

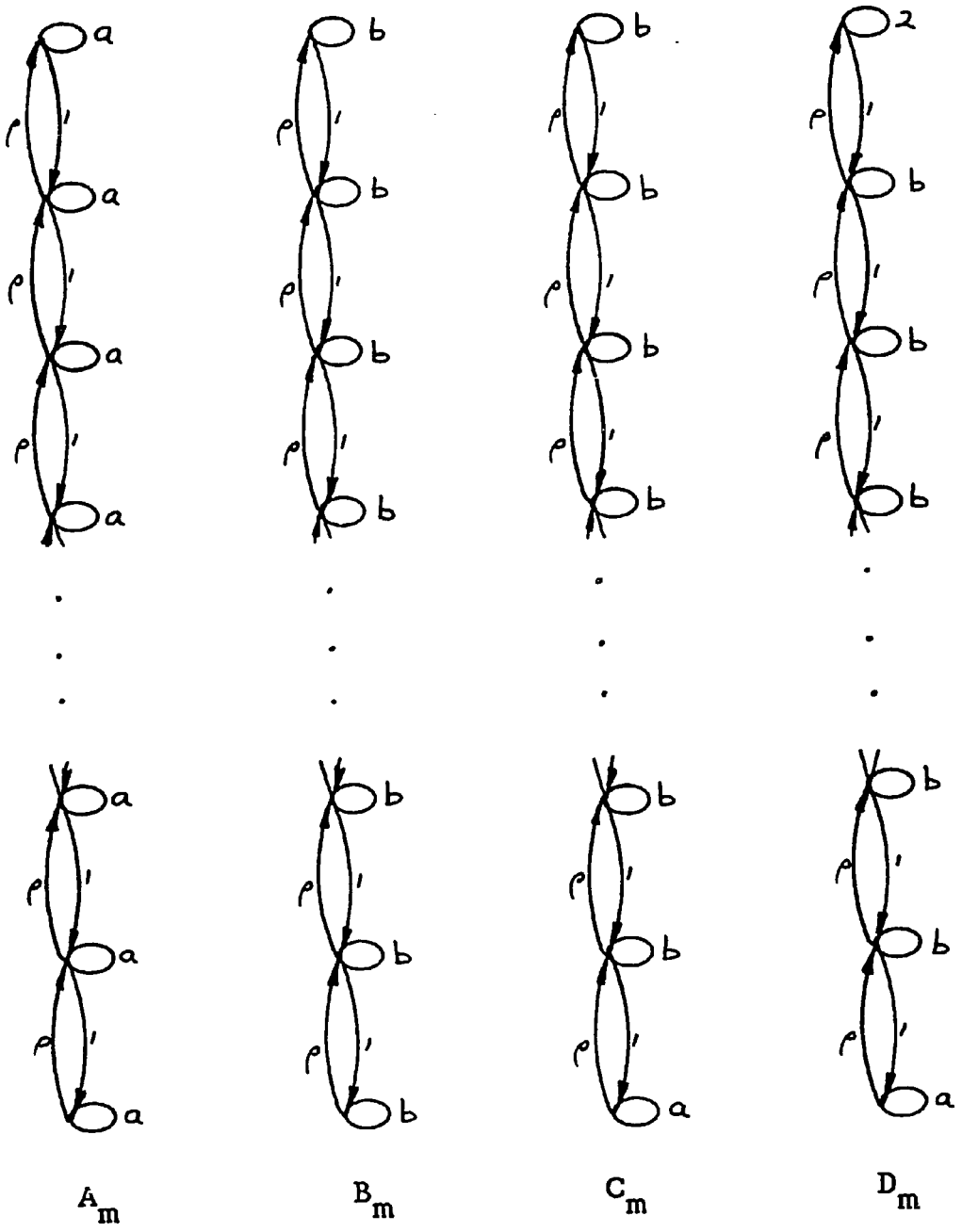


Fig. 9-3. The four column types.

the topmost and the bottommost. Successively balancing each of these, and dropping the balanced graph, as in figure 9-4,

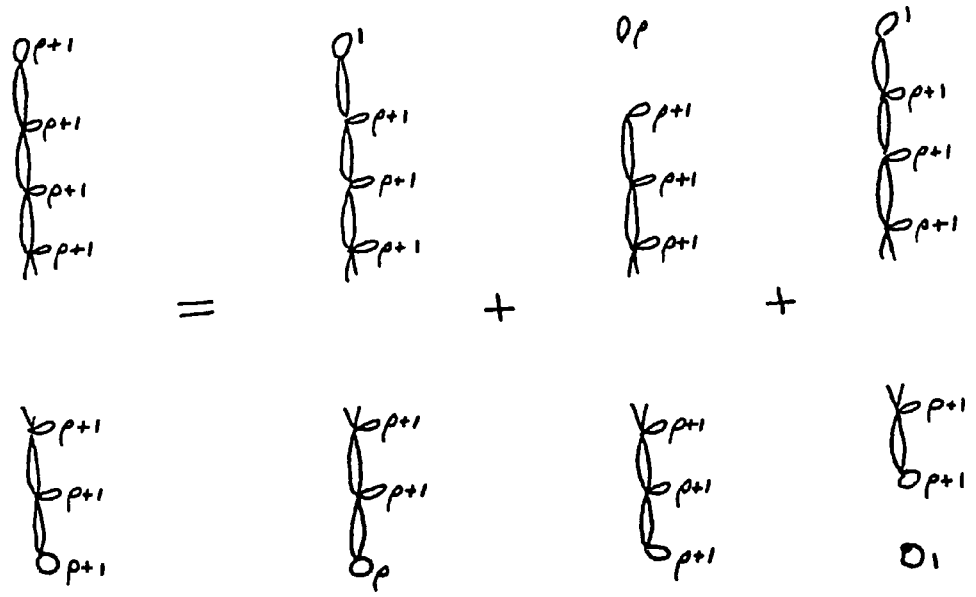


Fig. 9-4. Balancing A_m.

it is clear that $A_m = \rho A_{m-1} + 1$ (because the last graph can be repeatedly balanced until it is just a set of loops, each of weight 1). Using the initial condition $A_1 = \rho+1$,

$$A_2 = \rho(\rho+1) + 1 = \rho^2+\rho+1$$

$$A_3 = \rho(\rho^2+\rho+1) + 1 = \rho^3+\rho^2+\rho+1$$

$$\vdots$$

$$A_m = \rho^m+\rho^{m-1}+ \ldots +\rho+1$$

In general, the,

$$A_m = \frac{1-\rho^{m+1}}{1-\rho}$$

Proceeding to the second type of column subgraph, $B_1 = \rho+2$ and $B_2 = \rho^2+3\rho+4$, by direct calculation. Because every point in B_m is out of balance, the balancing algorithm will not produce immediate results as it did for A_m. By using the point separation theorem, however, as in figure 9-5,

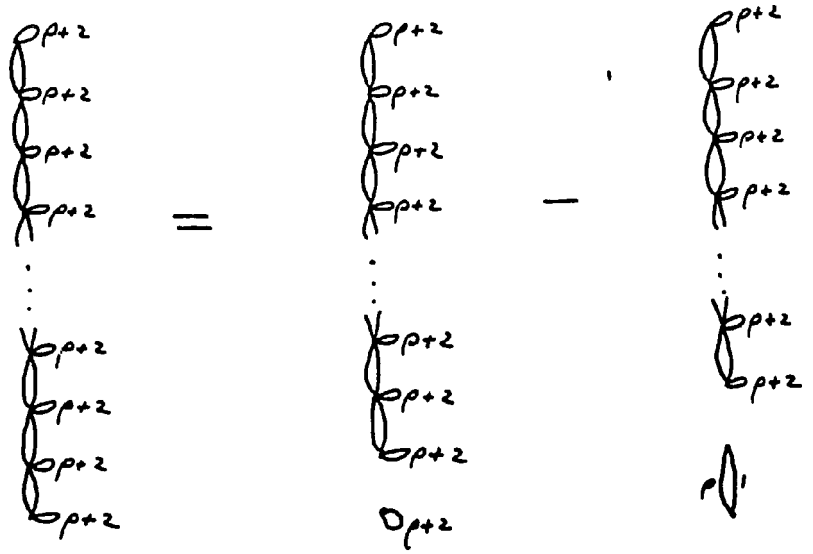


Fig. 9-5. Separating the bottom point.

we do obtain the difference equation $B_m = (\rho+2)B_{m-1} - \rho B_{m-2}$. Defining $B_0 = 1$, we verify that the equation will work for m as low as 2. We wish, then, to solve the difference equation for B_m using the two initial conditions $B_0 = 1$ and $B_1 = \rho+2$.

Using conventional techniques for solving difference equations, first assume that B_m can be written as $B_m = kr^m$, where k and r are constants yet to be determined. If so, then the difference equation is

$$kr^m = (\rho+2)kr^{m-1} - \rho kr^{m-2}$$

or

$$r^2 - (\rho+2)r + \rho = 0.$$

There are two roots to this equation:

$$r_1 = \frac{(\rho+2)+\sqrt{(\rho+2)^2-4\rho}}{2}$$

$$r_2 = \frac{(\rho+2)-\sqrt{(\rho+2)^2-4\rho}}{2}$$

An adequate solution to the difference equation is therefore given by

$$B_m = k_1 r_1^m + k_2 r_2^m$$

where r_1 and r_2 are the roots given above, and k_1 and k_2 are constants to be determined from the initial conditions. A more convenient solution is obtained if the trigonometric substitution

$$\cosh \theta = \frac{\rho+2}{2\sqrt{\rho}}$$

is made. The hyperbolic function is required, as opposed to the ordinary cosine, because $(\rho+2)/2\sqrt{\rho}$ exceeds 1. Performing the substitution and using a number of trigonometric identities, the general solution is given by

$$B_m = \rho^{m/2}\{k_1 \cosh m\theta + k_2 \sinh m\theta\}$$

where k_1 and k_2 are to be determined from the initial conditions. Substituting m=0 in the general form,

$$1 = B_0 = k_1$$

$$k_1 = 1$$

Substituting m=1,

$$\rho+2 = B_1 = \rho^{1/2}\{\cosh\theta + k_2 \sinh\theta\}$$

$$(\sinh\theta)\, k_2 = \frac{\rho+2}{\sqrt{\rho}} - \cosh\theta = \cosh\theta$$

$$k_2 = \frac{\cosh\theta}{\sinh\theta} = \coth\theta$$

Summarizing,

$$B_m = \rho^{m/2}\{\cosh m\theta + \coth\theta \sinh m\theta\}$$

where $\theta = \operatorname{arccosh} \frac{\rho+2}{2\sqrt{\rho}}$

Observe that columns of the third type differ from those of the second type only in the value of the bottom loop. Using the point separation theorem on the bottom point,

$$C_m = (\rho+1)B_{m-1} - \rho B_{m-2}$$

But we have already established that

$$B_m = (\rho+2)B_{m-1} - \rho B_{m-2}$$

Consequently,

$$C_m = B_m - B_{m-1}$$

Because B_m is already known, C_m is adequately represented by this expression.

Columns of the fourth type differ from those of the third type only in the value of the top loop. Using an argument similar to that just given,

$$D_m = C_m - \rho C_{m-1}$$

Substituting for C_m and C_{m-1},

$$D_m = B_m - (\rho+1)B_{m-1} + \rho B_{m-2}$$

$$D_m = B_{m-1}$$

The Case N = 1

To see how the column determinants can be used as "building blocks" of a solution, consider first the N = 1 system. The modified transition diagram of this system, shown in figure 9-6, has only two columns: column 0 and column 1.

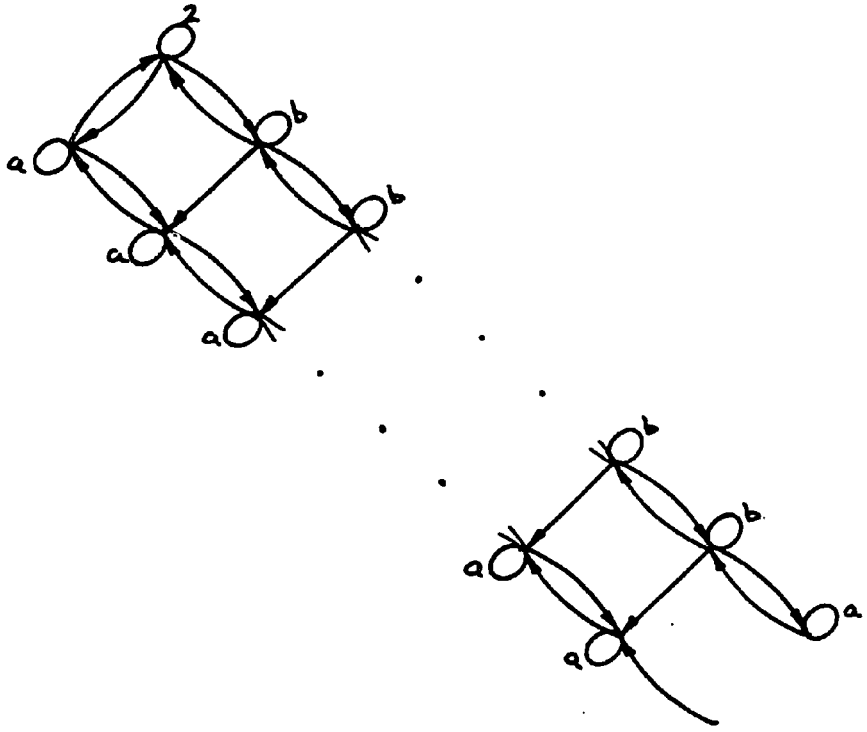


Fig. 9-6. Transition diagram when N=1.

Referring to figure 9-6, notice that column 1 can be reached from column 0 only through the topmost point. Thus

any path from the point (0,0) to a point (i,1) in column 1 must contain the line from point (M,0) to point (i,1). Similarly, any cycle containing points in both columns must contain the same line. With these observations, we may proceed to the determination of the relative coefficients.

For states in column 1, the relative coefficients are determined as follows:

a) State (0,1). There is only one path to this state, and it touches all of the points. Hence $c^*(0,1) = \rho^{M+1}(\rho+1)$.

b) State (1,1). Again, there is only one path; it touches every point but the bottom one of column 1. Hence $c^*(1,1) = \rho^{M+1}(\rho+1)$.

c) State (k,1). In general, there is a single path to any point in column 1, which leaves only the points below that point to be covered by cycles. Every path has weight ρ^{M+1}; the value of the determinant of the remaining subgraph is given by C_i, since this subgraph is just a column of the third type containing i points. Hence

$$c^*(i,1) = \rho^{M+1}C_i$$

For states in column 0, the relative coefficients are determined as follows:

a) State (M,o). A single path touches all of the points of column zero and leaves all of column 1. Since column one is of the fourth type and has M+1 points,

$$c^*(M,0) = \rho^{M}D_{M+1}$$

b) State (M-1,0). The path touches all except the top point of column 0 and all of the points of column 1. The path has the value ρ^{M-1}. The remaining subgraph would have the value $(\rho+1)D_{M+1}$ if there were no possibility of cycles between the columns. But in this case, there is exactly one cycle between the columns, namely the one involving the top points of the two columns. Hence the graph determinant actually has the value

$$(\rho+1)D_{M+1} - \rho C_M$$

(formally, this follows from theorem 6.2, the point separation theorem), and therefore,

$$c^*(M-1,0) = \rho^{M-1}\{(\rho+1)D_{M+1} - \rho C_M\}$$

c) State (M-2,0). The path has the value ρ^{M-2}. The remaining subgraph consists of the top two points of column 0 and all of column 1. The determinant is the product of the two column determinants, A_2D_{M+1}, corrected to take account of cycles between the columns. (Formally, this is theorem 6.3.) There are two such cycles: the first consisting of the top points of each column and the second consisting of the top two points of each column. See figure 9-7.

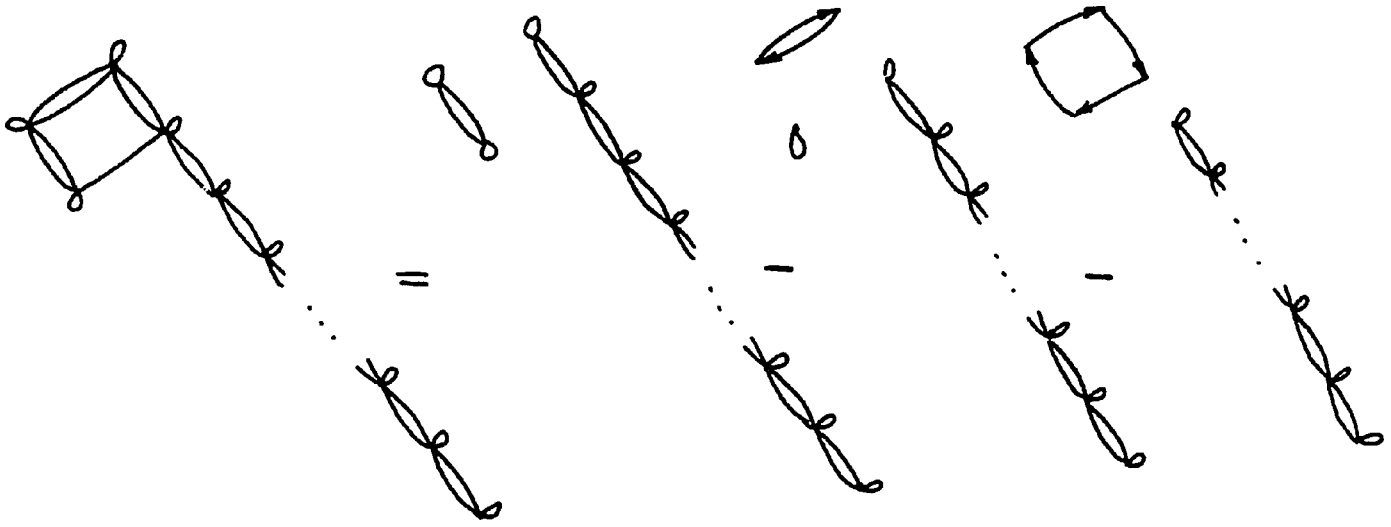


Fig. 9-7. The remaining subgraph decomposed.

so the relative coefficient is

$$c^*(M-2,0) = \rho^{M-2}\{A_2 D_{M+1} - \rho A_1 C_M - \rho^2 A_0 C_{M-1}\}$$

where A_0 is defined to be 1.

d) State (i,0). The relative coefficient is the product of the path value, ρ^i, and the determinant of the remaining subgraph which consists of the top (M-i) points of column 0 and all of column 1. The determinant of this graph is the product of the column determinants, $A_{M-i} D_{M+1}$, corrected for cycles between columns. For each of the M-i top points of column 0, there is an additional between-columns cycle. Hence

$$c^*(i,0) = \rho^i\{A_{M-i} D_{M+1} - \sum_{k=1}^{M-i} \rho^k A_{M-i-k} C_{M+1-k}\}$$

This general formula applies for all i = 0,1,...M-1. It applies to the case i = M as well if it is accepted that the sum from k=1 to k=0 is zero.

This completes the representation of the c*(i,j) in computable form. It is quite feasible to program a computer to produce values for the c*(i,j), for all i,j, any value of ρ, any value of M. It is then a small task to retrieve the state probabilities which were originally desired, since

$$q(i,j) = c^*(i,j)q^*(0,0)$$

where q*(0,0) is obtained by normalizing. That is, after the relative coefficients are calculated, they are summed, and each is finally modified by dividing each by the sum. The results are the state probabilities.

The Case N = 2

The solution just presented is for the overflow system in which the first server's queue may be of any specified capacity, but the second server's queue has a maximum capacity of one. Let us consider now the overflow system in which the second queue has a maximum capacity of two. The modified transition diagram has three columns, designated zero, one, and two. See figure 9-8.

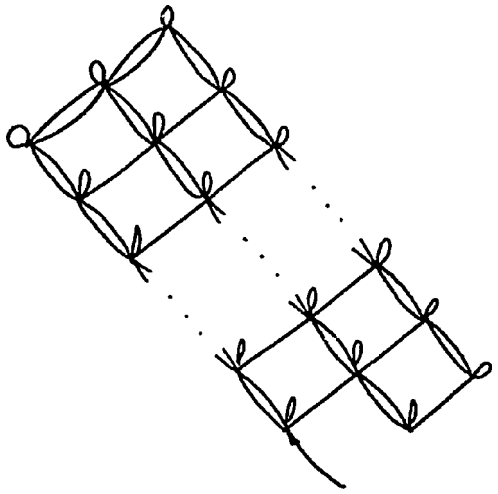


Fig. 9-8. Transition diagram when N=2.

Consider first the relative coefficients for states in column two. From experience with the N=1 case, it is clear that there is only one path to each state in this column and that it touches all of the points of column zero. Notice now that it touches column one at the top point only, leaving the other M points of this column isolated. Since there is only one line passing from column one to column two and this is used in the path, there can be no cycles between columns one and two involved in the relative coefficients of states in column two. Convinced that there are no complications, the solution for states of this column can be written directly.

$$c^*(i,2) = \rho^{M+2} C_M C_i$$

Consider next the relative coefficients for states in column one. The situation is exactly as it was for states in column one in the N=1 case, except for the possibility of paths through points of column two. Hence the values

are as before, viz.

$$c^*(i,1) = \rho^{M+1}C_i$$

multiplied by the graph determinant of column two, D_{M+1}, and corrected for paths going through points of column two.

Let us consider a specific case, $c^*(0,1)$. We take account of all terms involving the single path which does not include points of column two with the product $\rho^{M+1}C_0D_{M+1}$. Consider the paths to the point (0,1) which do pass through column two. They look like the paths in figure 9-9.

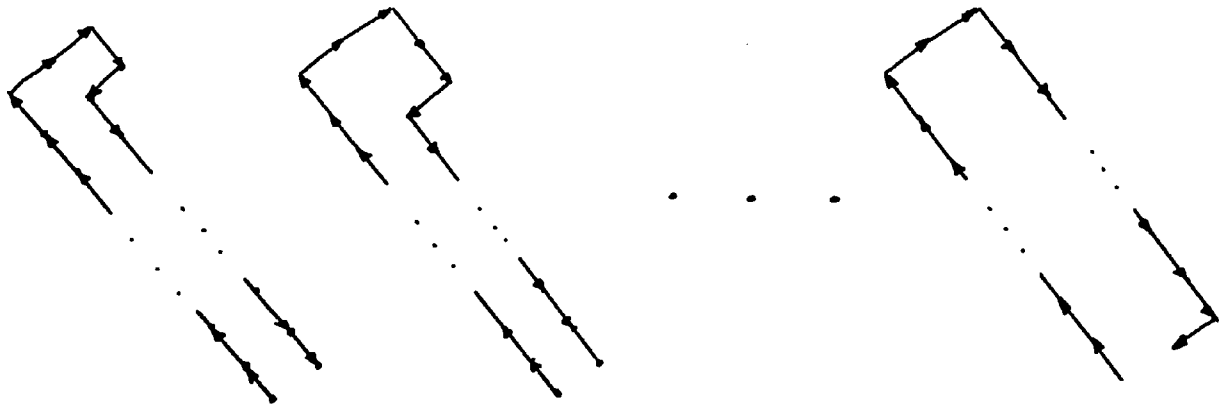


Fig. 9-9. The set of paths.

That is, each path must contain the only line from column one to column two, and there is a distinct path for each distinct line passing from column two to column one (except for the one at the top). The remaining subgraph in each case consists of two components; the remainder of column one and the remainder of column two. There can be no connection since the sole line from column one to column two is used in the path. The paths are of the same length and all have the value ρ^{M+2}. Putting all this together,

$$c^*(0,1) = \rho^{M+1}C_0D_{M+1} - \rho^{M+2}[B_0C_{M-1}+B_1C_{M-2}+\ldots+B_{M-1}C_0]$$

$$= \rho^{M+1}[C_0D_{M+1} - \rho\sum_{k=0}^{M-1} B_kC_{M-1-k}]$$

A slightly different situation arises for other states in column one. It is not always the case that all paths have the same weight. Consider a point somewhere in the middle of column one. Some of the paths, those pictured in figure 9-10, have the value ρ^{M+2},



Fig. 9-10. Paths having value ρ^{M+2}.

but the remainder, shown in figure 9-11, have higher values.

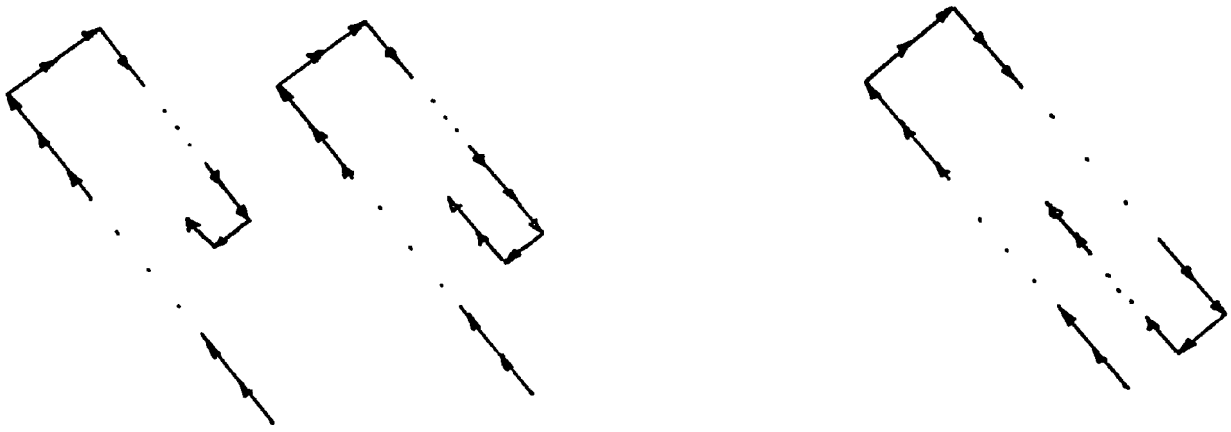


Fig. 9-11. Paths having higher values.

It is still true that the remaining subgraph is not connected, but this time there are two components from column one and one from column two. Thus, in the general case, the correction terms for states in column one are:

$$\rho^{M+2}B_0C_iC_{M-1} + \rho^{M+2}B_1C_iC_{M-2} + \ldots + \rho^{M+2}B_{M-1-i}C_iC_i$$
$$+ \rho^{M+3}B_{M-1-i}C_{i-1}C_{i-1} + \rho^{M+4}B_{M-1-i}C_{i-2}C_{i-2}$$
$$+ \ldots \rho^{M+2+i}B_{M-1-i}C_0C_0$$

Therefore, the general case is really

$$c^*(i,1) = \rho^{M+1}C_iD_{M+1} - \rho^{M+2}C_i\sum_{k=0}^{M-1-i} B_kC_{M-1-k}$$
$$-\rho^{M+2}B_{M-1-i}\sum_{k=1}^{i} \rho^kC_{i-k}C_{i-k}$$

It is easily verified that $c^*(0,1)$ is just a special case of this equation.

Finally, consider the relative coefficients for states in column zero. This time the path is again unique, but we must consider cycles between columns one and two, and possibly even column zero. Consider first the state (M,0). The path occupies all of column zero, and is of weight ρ^M. The remaining subgraph consists of columns one and two. See figure 9-12.

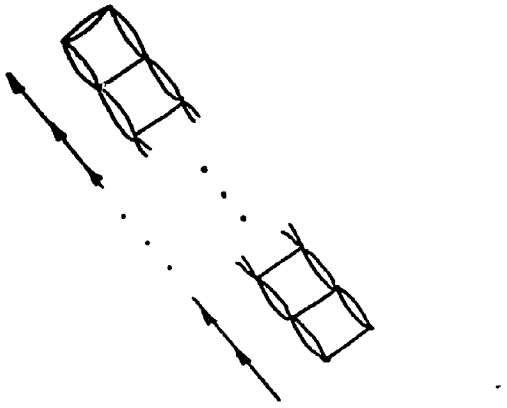


Fig. 9-12. The path and remaining subgraph.

Ignoring cycles between columns, the determinant value is the product of the column determinants, $C_{M+1}D_{M+1}$. The cycles between columns, shown in figure 9-13,

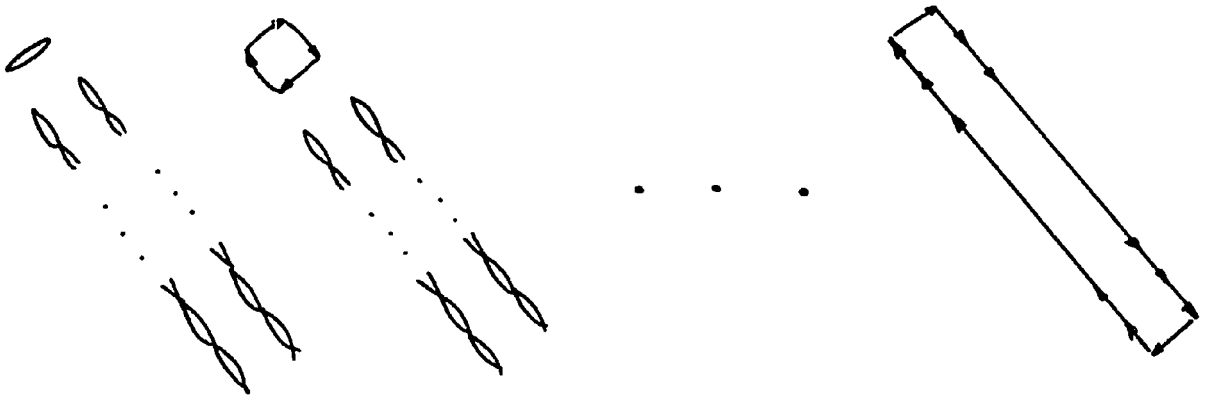


Fig. 9-13. Cycles between columns.

must necessarily contain the single line from column one to column two. There is a distinct cycle for each line returning to column one, including the one at the top. There can be no others, because there is only one line from column one to column two. So the corrections are

$$\rho C_M C_M + \rho^2 C_{M-1} C_{M-1} + \ldots + \rho^{M+1} C_0 C_0$$

The relative coefficient is therefore

$$c^*(M,0) = \rho^M\left[C_{M+1}D_{M+1} - \sum_{k=1}^{M+1} \rho^k C_{M+1-k}C_{M+1-k}\right]$$

Proceeding step-by-step in order to introduce complications gradually, consider now the state (M-1,0). The single path has the value ρ^{M-1}. The remaining graph consists of the top point of column zero and all of columns one and two. Ignoring cycles between columns zero and one, the situation is like that just considered. See figure 9-14.

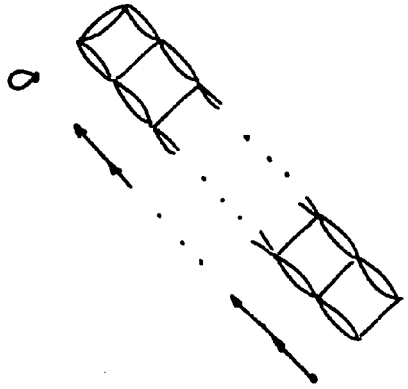


Fig. 9-14. The path and remaining subgraph.

In addition, however, there is one cycle between columns zero and one which has not yet been accounted for, as shown in figure 9-15.

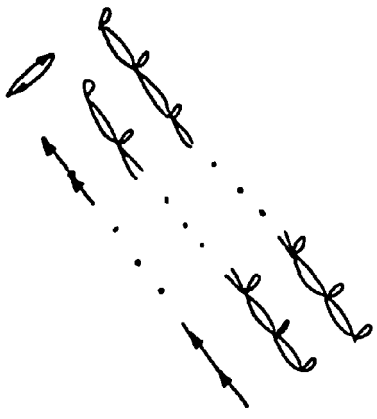


Fig. 9-15. The other cycle.

Putting these together,

$$c^*(M-1,0) = \rho^{M-1}\left[A_1\left\{C_{M+1}D_{M+1} - \sum_{k=1}^{M+1} \rho^k C_{M+1-k}C_{M+1-k}\right\} - \rho C_M D_{M+1}\right]$$

Continuing to the most general case, consider an arbitrary state in the zero column, (i,0). The unique path has the value ρ^i. Ignoring cycles between column zero and column one, the determinant of the remaining subgraph has the value

$$A_{M-i}\left\{C_{M+1}D_{M+1} - \sum_{k=1}^{M+1} \rho^k C_{M+1-k}C_{M+1-k}\right\}$$

See figure 9-16.

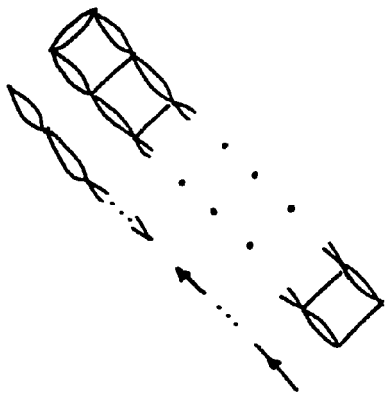


Fig. 9-16. The path and remaining subgraph.

The remaining terms, which correct the above to take account of the cycles it ignores, can be classified into two groups: those containing cycles between columns zero

and one but not between one and two, and those containing cycles involving all three columns. The former of these two classes, shown in figure 9-17, is easily handled.

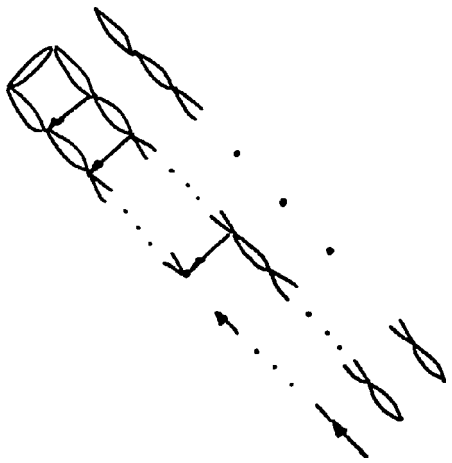


Fig. 9-17. Some of the correction terms.

Comparing figure 9-17 to the corresponding graph for the N=1 case, this graph is seen to have the determinant value

$$D_{M+1}\left[A_{M-i}C_{M+1} - \sum_{k=1}^{M-i} \rho^{k} A_{M-i-k}C_{M+1-k}\right]$$

There remain the correction terms corresponding to cycles involving all three columns. These can be systematically arranged as in figure 9-18.

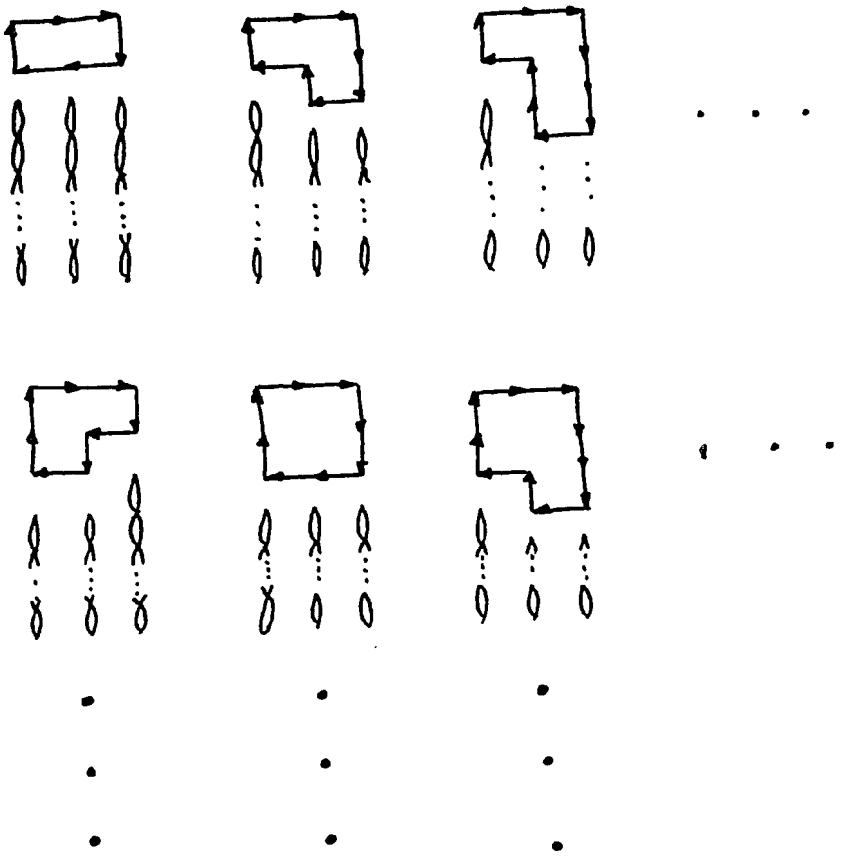


Fig. 9-18. The rest of the correction terms.

Notice that the terms are arranged in a rectangle. There are as many columns to this rectangle as there are lines from column two to column one (excluding the top one), namely, M. There are as many rows as there are lines from column one to column zero <u>above</u> the point (i,0), or M-i-1. Designating a term by its location in this rectangle, term k,j has the value

$$p^{j+2}A_{M-i-k-1}B_{r-1}C_{M+1-j-k}C_{m-j}$$

where r is the minimum of k and j. The sum of all of them, then, is

$$\sum_{j=1}^{M} \rho^{j+2} \sum_{k=1}^{M-i-1} A_{M-i-k-1}\, B_{r-1}\, C_{M+1-j-k}\, C_{m-j}$$

In summary, then,

$$c^*(i,0) = \rho^i \Big[A_{M-i} \Big\{ C_{M+1} D_{M+1} - \sum_{k=1}^{M+1} \rho^k C_{M+1-k} C_{M+1-k} \Big\} -$$

$$D_{M+1} \Big\{ A_{M-i} C_{M+1} - \sum_{k=1}^{M-i} \rho^k A_{M-i-k} C_{M+1-k} \Big\} -$$

$$\sum_{j=1}^{M} \rho^{j+2} \sum_{k=1}^{M-i-1} A_{M-i-1-k} B_{r-1} C_{M+1-j-k} C_{M-j} \Big]$$

This completes the solution of the overflow system in the case where N=2. Normalization is, of course, necessary to obtain the probabilities.

Approximate Solution For The General Case

By now it should be apparent that it is possible, but extremely tedious, to obtain the relative coefficients in the general case. The solution obtained would contribute little to our understanding of the overflow system, however, because of its complexity. In contrast to this situation, the approximations to the general solution implied by the theorems of chapter 7 are very easy to obtain and have considerable intuitive value.

Notice first that the overflow system as described at the beginning of this chapter is a parallel system having

a single blocking state. In order to apply theorems 7.3, 7.4 and 7.5, we must know the number of states and the number of levels in the transition diagram for the general case.

Lemma 9.1. The total number of states, excluding the empty state, is MN+M+N.

Proof: Since any number of customers from o through M may be present in the first server's queue, and any number from 0 through N in the second, the total number of states including the empty state is (M+1)(N+1). Subtracting 1 for the empty state, lemma 9.1 follows.

Lemma 9.2. L, the number of levels, is M+N.

Proof. L is the level of the topmost point (M,N). Since the level of any state is the number of customers present in the system when the system is in that state, the level of state (M,N) is M+N.

The above two lemmas provide the specific numbers required by the approximation theorems. The results obtained by applying these theorems to the overflow problem are given in the following three theorems.

Theorem 9.1. For the overflow system, (i,j) is any state, the smallest r for which c*(i,j) does not equal zero is i, when j=0, and M+j, when j>0.

Proof: When j=0, the state (i,j) is in the zero column. The shortest path to any such point is of length i. When j>0, the shortest path to any point consists of the M

upward lines in column zero, exactly j upward lines from the top point of column zero to the op point of column j, and a number of downward lines within column j. The theorem follows directly from theorem 7.3.

Theorem 9.2. For the overflow system, if (i,j) is any state, the largest r for which $c^*(i,j)$ does not equal zero is MN+i+j.

Proof: The level of state (i,j) is i+j. From lemma 9.1, the total number of states is MN+M+N. From lemma 9.2, the number of levels is M+N. Using theorem 7.4, the required r is equal to

$$(MN+M+N) - (M+N) + (i+j) = MN+i+j$$

Putting theorems 9.1 and 9.2 together, $c^*(i,j)$ is a polynomial having powers of ρ running from M+j to MN+i+j, for j>0, and from i to MN+i, for j=0.

Theorem 9.3. For $\rho<<1$, q_{i0} is of the order of ρ^i, and q_{ij}, for j>0, is of the order of ρ^{M+j}. For $\rho>>1$, q_{ij} is of the order of $\rho^{i+j-M-N}$.

Proof: This theorem is a direct application of theorem 7.5, using what has already been said about the shortest path to (i,j) and the number of levels in the diagram.

Theorem 9.3, in addition to providing approximate values for individual probabilities, carries some interesting implications about the relationship of different states. For any specific case, M and N are fixed. Consequently, for $\rho<<1$, the (approximate) steady state probability of a state (i,j) depends only on j when i>0. In terms of the

transition diagram, every state in a particular column, other than the zero column, has approximately the same probability when $\rho \ll 1$. For $\rho \gg 1$, it is the sum i+j which determines the probability, so every state in a particular level has approximately the same probability.

Consider now what these statements mean in terms of the system itself. Under low traffic conditions, given that the second server is busy, the probability of finding i items in the first server's queue is independent of i. That is, for a given state of the second queue, other than empty, all states of the first queue are (approximately) equally likely. This result is not intuitively obvious. One might have guessed that if the second server is busy (indicating that overflow has occurred in the recent past), that the likelihood of finding the first queue full would be much greater than that of finding it empty. Under conditions of light traffic, this conjecture is false.

Under heavy traffic conditions, the probability of any state depends only on the number in the system. That is, it is as likely that one will find, say, none in the first queue and five in the second as it is that one will find five in the first and none in the second. Again, this result is not intuitively obvious.

PART IV

CONCLUSION

CHAPTER 10

SOME INFORMAL REMARKS

Having completed all of the theoretical development and having examined in depth two examples of its application, it is now appropriate to make some brief, informal judgments, and to indicate directions for further research.

Observations

The examples considered in chapters 8 and 9 happened to have very regular transition diagrams. If the transition diagram of a system cannot be neatly drawn in two dimensions, one probably should not try to use this method, because it relies upon "seeing" all paths and cycles. On the other hand, whenever the transition diagram is small or very regular, the technique is at least potentially suitable for hand computation.

The method could undoubtedly be programmed for use with a computer, but this is not particularly recommended. The whole advantage of the technique lies in the graphs. For purposes of digital representation, one would probably translate the graphs into variable adjacency matrices, but under such circumstances, it would not be surprising if the computer program reduced to the ordinary determinant method for solving equations.

Relying as it does upon visual recognition and intuition, the application fo the method has aspects of an art or skill. One becomes "better" at using the method (and hence it becomes more useful) as one acquires familiarity with the techniques. This characteristic does not, however, detract from the mathematical generality of the method (the same kinds of statements could be made in reference to other established mathematical methods, such as those for solving differential equations). On the contrary, the one great advantage that this graphic method enjoys over purely algebraic methods is the intuitive one.

Directions For Further Research

One of the most obvious limitations of the theory presented in this paper is the requirement that the systems be finite. Extension of the results to infinite systems should be possible, but will require additional considerations at several points. For one thing, graph theorists have so far had little to say about infinite graphs. For another, the steady-state solution may not even exist. Still, the fact that the correct solution to the infinite two-server instantaneous jockey system is obtained by letting M go to infinity in the solution we obtained for the finite case should serve as encouragement for the development of the necessary theory.

Although decomposition has been studied, the concept of altering a transition diagram to simplify calculation

has been left untouched. Two papers by Chen [2] and [3], may provide suggestions for ways in which this may be done. Because of the relation between flowgraphs and determinants, some of the "tricks" used in the evaluation of determinants may have convenient graphical interpretations.

In order to obtain the approximation theorems, a number of restrictions were placed upon the systems considered. It would be interesting to see what happens when the restrictions are weakened. The requirement that the system possess a unique blocking state, for example, should be dispensible. It would also be worthwhile to attempt to find approximation theorems for non-parallel systems.

BIBLIOGRAPHY

1. Benes, V. E., Mathematical Theory of Connecting Networks and Telephone Traffic. New York: Academic Press, 1965.

2. Chen, W. K., "On Modifications of Flow Graphs," SIAM J. vol. 13 (1965), 493-505.

3. Chen, W. K., "On Directed Graph Solutions of Linear Algebraic Equations," SIAM Review vol. 9 (1967), 692-707.

4. Chen, W. K., "On Flow Graph Solutions of Linear Algebraic Equations," SIAM J. App. Math. vol. 15 (1967), 136-142.

5. Coates, C. L., "Flow-graph Solutions of Linear Algebraic Equations," IRE Trans. Circuit Theory vol. CT-6 (1959), 170-187.

6. Cox, D. R., and H. D. Miller, The Theory of Stochastic Processes. New York: John Wiley & Sons, Inc., 1965.

7. Cox, D. R., and W. L. Smith, Queues. London: Methuen & Co., Ltd., 1961.

8. Desoer, C. A., "The Optimum Formula for the Gain of a Flow Graph or a Simple Derivation of Coates' Formula," Proc. IRE vol. 48 (1960), 883-889.

9. Disney, R. L., "Some Multichannel Queueing Problems with Ordered Entry," J. Indust. Engin. vol. XIII (1962), No. 1.

10. Disney, R. L., "Some Results of Multichannel Queueing Problems with Ordered Entry—An Application to Conveyor Theory," J. Indust. Engin. vol. XIV (1963), No. 2, 105-108.

11. Disney, R. L., "Analytic Studies of Stochastic Networks Using Methods of Network Decomposition," J. Indust. Engin. vol. XVIII (1967), No. 1, 140-145.

12. Disney, R. L., and J. J. Solberg, "The Effect of Three Switching Rules on Queueing Networks," J. Indust. Engin. vol. XIX (1968), No. 12, 584-590.

13. Gupta, S. K., "Analysis of a Two-Channel Queueing Problem with Ordered Entry," J. Indust. Engin. vol. XVII (1966), No. 1, 54-55.

14. Harary, F., "The Determinant of the Adjacency Matrix of a Graph," SIAM Review vol. 4 (1962), 202-210.

15. Harary, F., R. Z. Norman, and D. Cartwright, Structural Models: An Introduction to the Theory of Directed Graphs. New York: John Wiley & Sons, Inc., 1965.

16. Hoskins, R. F., "Flow-graph and Signal Flow-graph Analysis of Linear Systems," in Recent Developments in Network Theory, ed. by S. R. Deards. New York: Macmillan, 1963.

17. Howard, R. A., "Sequential Decision Processes," in Notes on Operations Research, ed. by R. A. Howard and G. E. Kimball, Operations Research Center, MIT. Cambridge: The Technology Press, 1959.

18. Huggins, W. H., "Signal Flow Graphs and Random Signals," Proc. IRE vol. 45 (1957), 74-86.

19. Karlin, S., A First Course in Stochastic Processes. New York: Academic Press, 1966.

20. Koenigsberg, E., "On Jockeying in Queues," Man. Sci. vol. 12 (1966), 412-436.

21. Lorens, C. S., Flowgraphs. New York: McGraw-Hill, 1964.

22. Mason, S. J., "Feedback Theory: Some Properties of Signal Flow Graphs," Proc. IRE vol. 41 (1953), 1144-1156.

23. Morris, R., and E. Wolman, "A Note on 'Statistical Equilibrium'," Operations Research vol. 9 (1961), 751-753.

24. Morse, P. M., Queues, Inventories, and Maintenance. New York: John Wiley & Sons, Inc., 1958.

25. Saaty, T. L., Elements of Queueing Theory. New York: McGraw-Hill, 1961.

26. Saaty, T. L., "Stochastic Network Flows: Advances in Networks of Queues," in Proceedings of the Symposium on Congestion Theory, ed. by W. L. Smith and W. E. Wilkinson. Chapel Hill, North Carolina: The University of North Carolina Press, 1965.

27. Sittler, R. W., "Systems Analysis of Discrete Markov Processes," IRE Trans. Circuit Theory vol. CT-3 (1956), 257-266.

28. Stoller, D. S., Operations Research: Process and Strategy. Berkeley and Los Angeles: The University of California Press, 1964.

29. Syski, R., "Congestion in Telephone Exchanges," in Information Theory, Fourth London Symposium, ed. by Colin Cherry. New York: Butterworths, 1961.