1996

Schedule network node time distributions and arrow criticalities

Scott Kenneth Singleton
Iowa State University

Schedule network node time distributions and arrow criticalities

by

Scott Kenneth Singleton

A Dissertation Submitted to the

Graduate Faculty in Partial Fulfillment of the

Requirements for the Degree of

DOCTOR OF PHILOSOPHY

Department: Industrial and Manufacturing Systems Engineering
Major: Industrial Engineering

Approved:

Signature was redacted for privacy.

In Charge of Major Work

Signature was redacted for privacy.

For the Major Department

Signature was redacted for privacy.

For the Graduate College

Members of the Committee:

Signature was redacted for privacy.

Signature was redacted for privacy.

Signature was redacted for privacy.

Signature was redacted for privacy.

Signature was redacted for privacy.

Iowa State University
Ames, Iowa

1996

UMI Number: 9626069

UMI
300 North Zeeb Road
Ann Arbor, MI 48103

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGMENTS

The author acknowledges the attentiveness and inquisitiveness of Dr. H. T. David and Dr. Mike McCann. Dr. David has not only been a mentor, but also a friend. Dr. McCann has gone out of his way to be a part of academic research and has provided many valuable insights into the topic of this dissertation.

A special thank-you goes to Dr. Geraldine Montag who helped the author on the logistics of Ph.D. work. I.S.U. will miss her dedication to academic excellence. The author wishes her an enjoyable retirement.

Thank-you also goes to the IMSE department in general. They have provided funding with the U.S. Department of Education to help finance this research. Thank-you also goes to my wife, Nancy, for her support.

ABSTRACT

This research develops exact methods to calculate project duration distributions and to calculate Van Slyke's (1963) criticality for arrows, the probability that an arrow is on a critical path, assuming nonnegative integer duration distributions. These calculations for project duration distributions correct estimates made by the Program Evaluation and Review Technique (PERT), and the Van Slyke criticality calculations extend the arrow criticality analysis by the Critical Path Method (CPM) into the probabilistic realm.

Exact methods for calculating project duration distributions and Van Slyke's criticality are demonstrated on series networks, parallel networks, parallel-series networks, and the Wheatstone network. The Van Slyke criticality equation for parallel networks is in a form that appears to improve upon one proposed by Dodin & Elmaghraby (1985). The present form is generalized to, in principle, include all networks.

The exact methods are enhanced by developing a procedure to limit the number of calculations needed to analyze large networks. The procedure identifies paths through a large network, calculates the minimum and maximum path durations, and ranks the paths by duration. A smaller skeletal network is constructed from the arrows of the longest paths and is analyzed by exact methods. The procedure emphasizes accuracy for the longer project durations, of greatest concern to project managers and schedulers, while limiting the number of necessary calculations.

The procedure for large networks is illustrated on the 40-arrow Kleindorfer (1971) network. Of the 51 Kleindorfer paths, the procedure selected 6 paths to construct a skeletal network. Analysis of the skeletal network yields a project duration distribution that is correct in its range and in the duration probabilities for the upper 5% of the distribution. Analysis results are compared with SLAM II and FORTRAN simulations. No arrow criticality appears to be seriously miscalculated. The project duration distribution is calculated to be bimodal, in keeping with the simulation.

Conditions under which the just mentioned bimodality can occur are determined for parallel, normally-distributed paths. The large-network procedure warns when these oddly shaped distributions are possible.

INTRODUCTION

Projects such as building highways, erecting dams, and constructing airports require a great deal of capital over extended time periods. Because of contractual, financial, and resource requirements, project planning and scheduling are important concerns for all involved. Project forecasting is complicated by the uniqueness of the project, the lack of repeat measurements, and the complexity of relationships between necessary activities.

Common practice models relationships among project activities in graphical form by arrows interconnecting nodes. The graph is commonly called a network. Arrows represent project activities, and nodes represent events (Kelley & Walker, 1959) such as "Foundation Pouring Completed," "Start Project," or "Project Completed.". Associated with each activity's arrow is a beginning node, an ending node, and a time duration. A duration is the time it takes to complete an activity (arrow) once it has been started. Arrows point from their starting nodes to their ending nodes. An activity may not start until its beginning event has been completed; or, in modeling terms, an arrow may not be processed until its starting node has been realized. All activities on the network must be performed. None are performed more than once. Networks have one definite starting node, called a source node, and one definite ending node, called a sink node. Events (nodes), themselves, take no time to complete. Their completion times are wholly dependent upon the activities (arrows) that precede them.

Project planning is aided by the enhanced communication that a schedule network represents. The visualization of relationships among project activities helps everybody understand how the project objectives are to be accomplished. Because network event times and durations are better estimated, plans are made to provide resources in a timely manner, thus helping to insure successful completion of the project (Granof, 1983). When problems are found with the project plan, the effectiveness of proposed changes are rapidly determined by a revised network forecast (Miller, 1963).

Two forecasting methods are the Critical Path Method (CPM) and the Program Evaluation and Review Technique (PERT). CPM assumes deterministic arrow durations. It was designed "to determine how best to reduce the time required to perform routine ... work" (Moder, Phillips, & Davis, 1983, p. 13). CPM identifies a critical path through the network where a path is a directed sequence of arrows leading through a network from start to finish, and a critical path has a time duration equal to the shortest possible project duration. Path durations are determined by the summing of arrow durations from the constituent activities. The difference between the critical path's duration and another path's duration is called the float. Analysis of the float yields an indication of the criticality of a path or any activity on the path.

PERT was designed to accommodate statistical estimates for its activity durations. "The time estimates are obtained from responsible technical persons and are subsequently expressed in probability terms" (Malcolm, Rosenbloom, Clark, & Fazar, 1959, p. 646). In other words, any given duration has a probability associated with it. The distributions are assumed to be unimodal, independent, and finite in range (Martin, 1965). From whatever the distributions are, PERT calculates activity means and variances. Using the means, one critical path is found through the PERT network. Summing the mean and variances of activities on this path tends to yield the mean and variance of a normal distribution because of the Central Limit Theorem. The problem is that a path's distribution of time is not necessarily the project's distribution of time.

The authors of PERT knew their "estimated expected time of events are always too small" (Malcolm, Rosenbloom, Clark, & Fazar, 1959, p. 654). They kept the estimates for simplicity's sake. Estimates are too small because the designated critical path is not, in fact, always critical. Other paths may become critical when larger values happen to manifest themselves. The result is a raising of the project duration mean.

For stochastic networks, like those attributed to PERT, Van Slyke (1963) estimated the probabilities that arrows are on a critical path by simulation. This definition of arrow criticality shall be referred to as the Van Slyke criticality. MacCrimmon and Ryavec (1964) suggested focusing on Van Slyke criticalities rather than focusing on critical paths because "the PERT-calculated critical path does not necessarily contain the most critical activities" (p. 36). The Van Slyke criticality should "focus management attention on the 10 to 20 per cent of the projects activities that are most constraining" (Moder, Phillips, & Davis, 1983, p. 19).

Project overruns occur when actual project completion times are higher than the scheduled time. The probability of a project running over its scheduled time is important to management because it represents a risk. Schedule overruns typically have penalties associated with them. For that reason, the right tail of the forecasted distributions is of utmost importance.

In short, PERT has three problems: because of its dependency on one path, its offered distribution is questionable so that it provides an incorrect distribution with which to estimate the distribution tails. Also, its suggested mean is recognized to be too low. Finally, PERT, unlike CPM, does not identify critical paths or activities. Attempts to remedy these shortcoming are addressed in the Literature Review. None, however, have been wholly successful.

The research presented here analyzes stochastic input, like PERT's, to better estimate project duration distributions and Van Slyke arrow criticalities. First, the research assumes, in addition to the Moder, Phillips, & Davis (1983) network definitions, that arrow inputs are in the form of integer-valued distributions. Second, methods for calculating duration distributions to network nodes, especially the sink node, are reviewed and developed. Third, calculation methods for Van Slyke arrow criticalities are discussed. Fourth, the duration distribution and Van Slyke criticality calculation methods are incorporated into an algorithm. Recognizing that many project networks become unwieldy computationally, the algorithm addresses a relevant subset of the input distributions. Fifth, the algorithm is tested on a published discrete network and its conclusions are compared to the results of a SLAM II simulation. Finally, the research explains why schedule network duration distributions might have bimodal rather than Normal form. The algorithm provides exact right tail probabilities, the exact duration range, good left-tail probabilities, the shape of the duration distribution, and a list and ranking of the critical arrows contributing to project overruns.

LITERATURE REVIEW

Literature relevant to project forecast planning and critical component analysis is classified into the categories of Statistical Studies, Algorithms, Bounding Studies, Simulation, and Criticalities. These should be compared with the outputs of PERT:

> Outputs include the expected time for the completion of each event, the identification of slack and critical areas in the programs, an expression of the probability of equaling or meeting the current schedule and the specification of the latest date by which every event must be completed in order to meet the end-objective deadline (Malcolm, Rosenbloom, Clark, & Fazar, 1959, p. 662).

Statistical Studies

Clark (1961), one of the original PERT authors, addressed the problem of estimating the distribution of normal, parallel paths. He does so by generating four moments for various combinations of means, variances, and correlations of one path to the first path's standard normal distribution. Clark observed that the maximum of two normals is not another normal distribution.

MacCrimmon and Ryavec (1964) attempted to do a comprehensive analysis of the PERT model. Among their analyses is an analysis of the network duration probabilities. The PERT was found to never exceed the true duration's expectation. The "parallelism in a network will tend to skew the [node duration] distribution to the left" (MacCrimmon and Ryavec, 1964, p. 35). Dominated paths should be dropped from the network.

Martin (1965) focused on reducing the time distributions of the activities into an equivalent time distribution for the project. He does so by fitting polynomials to the arrow distributions. He has recipes to combine polynomials into new ones more representative of the project duration. Although the polynomials were cumbersome, his concepts are statistically valid. For any two arrows in a pure series network, reduction is performed by accumulating the probabilities from the two distributions whose values sum to equal the new time. For a pure parallel network, reduction is performed by multiplying the cumulative distribution functions (*cdf*'s) to derive the reduced network's *cdf*. The duration distribution of some networks may be exactly found by appropriately using parallel reductions and series reductions.

Reducing networks with complex dependencies may require conditioning. Once arrows common to two paths have their times fixed, the remaining parallel activities may be treated as independent, making the product of the two *cdfs* equal to the *cdf* of the conditioned distribution of the dual path duration distribution. To complete the reduction, each independent *cdf* is weighted by the conditional probabilities and then summed. He also used conditioning to estimate the probability that a given arrow is on a critical path. He finds this "Criticality Index" (Martin, 1965, p. 62) by first calculating the probability that a path is critical, then summing the criticalities of all paths that go through particular arrows.

Dodin (1985a) points out that the dependencies between paths makes it hard to identify critical paths and activities. "The difficulty caused by the dependency between the paths has led to the publication of more than 30 papers dealing with various issues of the stochastic network" (Dodin, 1985a, p. 223). He offers, as a theorem, that a network is not completely reducible if it contains what he calls an interdictive graph, more commonly known as the Wheatstone. By duplicating arrows in the network, Dodin resolves his irreducibility problem, but alters the final distribution.

Dodin and Sirvanci (1990) explored the impact of the Extreme Value Distribution on stochastic networks. They depict a mechanism for parallel activities that is conceptually analogous to the Central Limit Theorem for activities in series. The theory says that as the number of parallel, independent, identical distributions become large, their maximums will approach the The Extreme Value Distribution. "The extreme value theory, which is based on the maximum of independent and identically distributed random variables, is used to develop more accurate approximations and still be practical" (Dodin & Sirvanci, 1990, p. 398). The distribution has an expected value that is greater than the constituent arrows and is right skewed.

> The PERT method approximates the network by only one of its longest paths, in the extreme value case, it is approximated by a network consisting of all the dominating paths of the original network. As a result, when there is only one dominating path in the network, and the probability of the second dominating path is to be realized as the longest path is quite small, the PERT method will tend to give accurate results. However, when there are more than one dominating path, it is suggested that the extreme value approach be used to determine the mean and the variance (Dodin & Sirvanci, 1990, p. 408).

Bounding Distributions

Because node time distributions from networks with complex dependencies are difficult to generate analytically, distribution bounds have been studied. PERT provides a lower-bound for the distribution mean. Fulkerson (1962) provides a tighter lower bound. He handles the complexities by working with activity means and ignoring path dependencies at every node. Kleindorfer (1971) provides an upper bound while Shogan (1977) constructed tighter bounds than either Kleindorfer or Fulkerson. Shogan's analysis on the large 22-Node, 40-Arrow Kleindorfer (1971) network yielded a bounding cumulative distribution function. Robillard and Trahan (1977) generated bounds for the distribution moments.

Simulations

Monte Carlo simulations have been used to estimate both completion time distributions and arrow criticalities. Random samples from each arrow's distribution are assigned to the arrows to build a network realization. Observations from each realization are recorded. Through many repeated realizations, probability distributions of the network node durations and the probability that each activity will be critical are estimated. Van Slyke (1963) did the first published analysis of PERT using FORTRAN, and Pritsker and Kiviat (1969) addressed network simulation using GASP which is a forerunner of SLAM II (Pritsker, 1995).

Van Slyke (1963) analyzed with Monte Carlo simulations a variety of PERT networks. He used 10,000 network realizations to generate his numbers. The time for simulation was linear with the number of random samples. He states two methods to reduce random variables by taking activities out of the network. The first disposes of all activities whose maximum durations do not impact the network length when all others are set to the minimum value. The second analyzes a smaller number of Monte Carlo simulations, and discards any activities which were not on a sample critical path. He showed graphically, the underestimating of PERT in parallel configurations. He also generated distributions that modeled the extreme value distribution but were not identified as such. One of his distributions (Van Slyke, 1963, p. 857) was skewed right with what appears to be a lump in the right tail.

Burt and Garman (1971) reduced the number of random numbers needed by separating arrows by whether they were on just one path or not. Sampling from the one-path activities was performed much less frequently because they were used to condition the rest of the network. Thus, the procedure was named "Conditional Monte Carlo." "The usefulness of conditional Monte Carlo depends upon the number of 'unique' activities in a given network" (Burt & Garman, 1971, p. 211).

Sigal, Pritsker, and Solberg (1979) introduced the Uniformly Directed Cutset (UDC) for stochastic network analysis. The uniformly directed cutset is "a set of arcs [arrows] which connect a set of nodes, W, which contains the network source, with its complement, $\overline{W}$ in the set of network nodes, which contains the network sink" (Sigal, Pritsker, & Solberg, 1979, p. 378). The UDC replaces the conditioned activities identified by Burt and Garman. After identifying the UDC, their algorithm calls for setting the finish time to a particular duration called a base point. The cumulative probability of the base point is then based upon the sampling of all activities not on the cutset. To generate a complete distribution, the algorithm must iterate through all distribution times.

Algorithms

Methods for estimating network completion times using discrete distributions have been performed by Fulkerson (1962), Martin (1965), Kleindorfer (1971), and Shogan (1977), all of whom have been discussed. Others have been Keefer & Bodily (1983), Dodin (1985a & 1985b), Hagstrom (1988), and Bonett & Deckro (1993).

Anklesaria & Drezner (1986) assume a multivariate normal distribution for path durations in their analysis. They recommend trimming nodes from the network that have no chance of being on a critical path. They claim "Robillard and Trahan's lower bound is quite effective for relatively high probabilities but is very far from the actual probabilities for low probabilities" (Anklesaria & Drezner, 1986, p. 813).

Bonett & Deckro (1993) argue multinomial representation of activity durations can lead to an exact discrete distribution for the project duration. They calculate duration probabilities by summing across the independent arrows the probabilities of their combinations. The method yields exact results for parallel-series networks although they included a distribution with a total probability of 0.9 as part of their example . For more complicated networks, they duplicate nodes and arrows ruining their exactness efforts.

Sculli (1983) presents a method to approximate the completion times of PERT networks by assuming that all arrows are normally distributed. He assumes path independence and adds dummy nodes to limit to two the number of arrows terminating at any node.

> The Normality assumption for individual activity duration has often been made in the literature. And this can be justified by the fact that most large networks can be reduced to a guide network, where a completely independent path becomes one activity. The central limit theorem justifies the Normality assumption for the duration of activities in the guide network (Sculli, 1983, p. 157).

The Sculli and Shum (1991) technique is based on multivariate normal distributions and yield means and variances very close to those of simulation. They observe that there is a "perennial discussion as to why projects are always late.... The problem stems from what appears to be an inability to accurately estimate the completion time distribution of individual activities.... This indicates that there is still considerable research potential in ... the overall time analysis of networks" (Sculli & Shum, 1991, p. 7).

Dodin (1984) picks the most critical paths in the network by ranking cumulative distributions for each node's completion time. The cumulative distributions are only approximate because all activities ending at the node are assumed to be independent. Recognizing that ranking of the top paths do not necessarily correspond to rankings of probabilities that paths will be critical, Dodin maintains that the two top groups will be the same or nearly the same. He argues that his result "is very close to the exact distribution" (Dodin, 1985b, p. 262) because simulations converge toward it.

Criticality

Few have addressed how to quantify criticality by other than simulation methods. Dodin & Elmaghraby (1985) present an approximation, and Williams (1992) suggests some possible alternatives such as using the correlation coefficient between an activity's duration and that of the project. Correlation is unreliable because of nonlinearity of the constituent covariance variable.

Like the Dodin procedure described before, Dodin & Elmaghraby (1985) calculate the cumulative distributions of node values assuming independence of paths. With their procedure, they hope to avoid the time consuming tasks of "enumeration of all the paths in the activity network, the approximation of the corresponding critical paths, and the identification of the paths passing through each activity" (Dodin & Elmaghraby, 1985, p. 209). Arrow criticalities are approximated by assuming paths which merge at nodes are independent. The criticality of one of the incoming paths is calculated from the duration distributions of the paths from the source node. Although their equation for criticality is correct for independent paths, in general the paths are not independent. The results is to corrupt criticality from a probability of being on the critical path to merely an index. One of their criticality examples had a value of "1.0556" (Dodin & Elmaghraby, 1985, p. 214) which clearly can not be interpreted as a probability.

CALCULATION OF NETWORK DISTRIBUTIONS

With over thirty-five years of research, no one has been able to generate, in general, the duration distribution for networks with many activity arrows without extensive enumeration. Complete enumeration for a network quickly becomes impractical. For example, the Kleindorfer (1971) network with 40 activity arrows would require over $2X10^{24}$ combinations to analyze completely, and it is small compared to projects that have had hundreds of activities.

Stochastically discrete networks with one source and one sink node and no cycles are analyzable for project distributions according to the methods presented in this chapter. The effects of discrete distributions are examined before looking at reducing network arrow distributions into one distribution representing the project network's total duration. Network reductions techniques will be discussed for discrete networks that are in series, in parallel, in a combination of series and parallel, and in more complex arrangements as introduced by Martin (1965) but without assuming polynomial distributions. Martin used conditional probabilities for more complexly arranged networks. Bonett & Deckro (1993) attempted to find project duration distributions by modeling discrete arrow distributions by multinomials and analyzing the combinations of arrow durations. They were successful for parallel-series networks but not for more complex networks. This chapter will merge the statistically correct methods, originally for continuous distributions, from Martin (1965) with the multinomial concept from Bonett & Deckro (1993). These three modes of reductions are integrated to reduce distributions within a network to an equivalent distribution representing the completion time of the entire project. The three modes of series reduction, of parallel reduction, and of conditional probabilities are integrated to reduce network duration distributions into the distribution of the project's duration. All methods in this chapter will focus on statistically correct procedures.

The Discrete Distributed Network

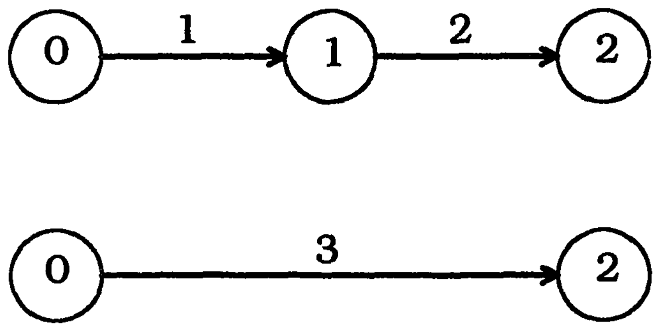
All activity distributions are assumed to be independent, discrete, and unimodal. The independent and unimodal assumptions have been standard throughout the literature and are generally accepted in industry as well. The discrete distribution is warranted on two counts. First, managers within the construction industry typically review their resource requirements on a periodic basis making resource allocations between reviews rare. First, because of periodic resource requirement reviews, union rules, and time lost in mid-day transportation of, resources are assigned by the entire day. The costs of allocating resources in a continuous nature can be no better than whole day assignments (Wagner & Whitin, 1958 and Hax & Candea, 1984). Therefore, any partial days in a schedule will be allocated the full day. Second, probability calculations are much easier for discrete values. Many continuous distributions are quantified through numerical integration which is a form of discrete approximation. With activities being estimated for 5 to 25 days, and projects commonly running over a year, there is enough resolution to build meaningful distributions.

One complicating factor of the discrete distributions is that it is possible to have ties; where with continuous distributions, there is no probability of ties. Industry has felt many times the pressure of simultaneous task completions. This drawback is, in reality, an important element of modeling the application. Some of the studies that have ignored the possibility of ties in completion times have been Fisher, Saisi, & Goldstein (1985), Bowman (1995), Clark (1961), and Kulkarni & Adlakha (1986).

The Series Reduction Operation

Martin (1965) reduced the duration distributions of activity arrows arranged in series by convoluting probabilities. The convolution operation forms a new time distribution by accumulating the probabilities from the distributions in the series into a single, representative distribution. An example will illustrate the procedure.

Consider a network where two activity arrows are in series. In reduction, Activities 1 and 2 (see Figure 1a) would be replaced by an equivalently distributed activity, Activity 3 (see Figure 1b). This reduces the size of the network by one activity arrow. If the distributions of Activities 1 and 2 were assigned according to Table 1, the minimum duration for Activity 3 is 5 duration units, and its maximum is 10. These are found by summing the minimum and maximum values, respectfully, of Activities 1 and 2. Because they are independent, the probability of Activity 3 taking 5 time units is found by multiplying the probabilities of Activity 1 and Activity 2 taking on their lowest values: $\Pr[Act_3 = 5] = \Pr[Act_1 = 3, Act_2 = 2] = \Pr[Act_1 = 3] \bullet \Pr[Act_2 = 2] = (0.1) \bullet (0.3) = 0.03$.



(a)

(b)

Figure 1: Reduction of Arrows in Series Example.
(a) Original Network. (b) Equivalently Distributed Reduced Network.

Calculations for intermediary values are slightly more complex. The probability of a duration of Activity 3 is found by summing the product of probabilities that correspond to the pairs of durations from Activities 1 and 2. For a duration of 8 for Activity 3, combination of Activity 1 and Activity 2 pairs are (3,5), (4,4), and (5,3). The resulting probability is $\Pr[Act_3 = 8] = (.2)(.3) + (.4)(.4) + (.3)(.3) = 0.31$. The other distribution values for Activity 3 are also listed in Table 1.

The convolution operation is generalized by the following equation:

$$f_{A3_t} = \sum_{j=Min(A1)+Min(A2)}^{Max(A1)+Max(A2)} \left\{ \sum_{i=Max(Min(A1),\ j-Max(A2))}^{Min(Max(A1),\ j-Min(A2))} \left[f_{A1_{t-i}} \cdot f_{A2_{t-j-i}} \right] \right\}. \quad (1)$$

where A3 is the distribution of the representative arc,
A1 is the distribution of one arc in series,
A2 is the distribution of the other arc in series, and
f_{A3_t} is the probability that distribution A3 takes on time t.

Repeating the convolution operation on activities in a pure series network, will methodically reduce the network's arrows by one until there is only one activity arrow left. When convoluting a large number of discrete distributions, the resulting distribution will be a discrete version of the normal distribution by the Central Limit Theorem. It is discrete instead of continuous because the constituent distributions started discrete, and the sum of integer variables is an integer variable (Bishop, Fienberg, & Holland, 1975).

Table 1: Probability Distributions for Series Reduction Example.

	Duration									
Activity	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Totals**
#1		0.10	0.20	0.40	0.30					1.00
#2	0.30	0.40	0.30							1.00
#3				0.03	0.10	0.23	0.31	0.24	0.09	1.00

The Parallel Reduction Operation

For a pure parallel network (see Figure 2), reduction of Arrow 1 and Arrow 2 into the equivalent distribution of Arrow 3 is performed by considering each time duration. The probability of achieving a given time period t on the equivalent distribution is the sum of three possibilities. First, both Arrow 1 and Arrow 2 have duration t. The other two possibilities are that Arrow 1 has value t and Arrow 2 has something less or vice versa. The probability of Activity 3 taking on duration t is expressible by the formula

$$\Pr[A3 = t] = \Pr[(A1 = t) \cap (A2 = t)] + \Pr[(A1 = t) \cap (A2 < t)] + \Pr[(A1 < t) \cap (A2 = t)] \ .$$

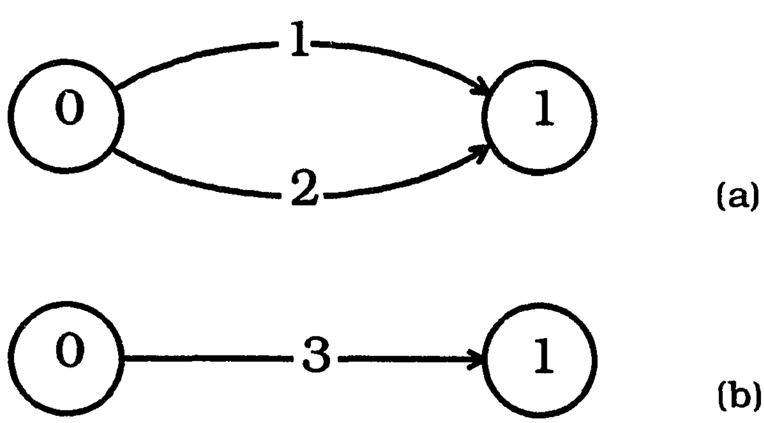


Figure 2: Parallel Reduction Example.
(a) Original Parallel Network. (b) Equivalent Reduced Network.

Recognizing that the probability of all values being less than t defines the cumulative distribution (Walpole & Myers, 1978, p. 34), indicated by F, and indicating the probability density function by f, the equation may also be expressed by

$$f3_t = f1_t f2_t + f1_t F2_{t-1} + F1_{t-1} f2_t \,. \qquad (2)$$

With some manipulation, a more straightforward method for calculating the probability is obtained.

$$f3_t = f1_t f2_t + f1_t F2_{t-1} + F1_{t-1} f2_t$$

$$f3_t = f1_t\left(f2_t + F2_{t-1}\right) + F1_{t-1} f2_t + \left[F1_{t-1} F2_{t-1} - F1_{t-1} F2_{t-1}\right]$$

$$f3_t = f1_t\left(f2_t + F2_{t-1}\right) + F1_{t-1}\left(f2_t + F2_{t-1}\right) - F1_{t-1} F2_{t-1}$$

$$f3_t = \left(f1_t + F1_{t-1}\right)\left(f2_t + F2_{t-1}\right) - F1_{t-1} F2_{t-1}$$

$$f3_t = F1_t F2_t - F1_{t-1} F2_{t-1}$$

$$F3_t = F1_t F2_t \qquad (3)$$

$$f3_t = F3_t - F3_{t-1} \,. \qquad (4)$$

In words, the distribution of the equivalent activity is calculated through its cumulative distribution that is the product of the cumulative distributions of Arrow 1 and Arrow 2. An example is given in Table 2. The probability density functions of Arrow 1 and 2's distributions are converted into cumulative distribution functions (cdf). The cdf product is calculated to generate the cdf of activity 3. Finally, the pdf of activity 3, the network-reducing equivalent activity, is derived by sequentially subtracting from its cdf probabilities the next lowest cdf probability.

Parallel - Series Reduction

When networks contain activities in series and parallel, the number of its duration distributions may be reduced, appropriately, by parallel and series operations. Most networks can be dramatically reduced, and many are reducible to a single arrow

Table 2: Parallel Network Reduction Example.

	Duration Units					
	1	**2**	**3**	**4**	**5**	**6**
Activity #1 p.d.f.	0.20	0.30	0.30	0.20	0.00	0.00
Activity #2 p.d.f.	0.00	0.20	0.40	0.20	0.10	0.10
Activity #1 c.d.f.	0.20	0.50	0.80	1.00	1.00	1.00
Activity #2 c.d.f.	0.00	0.20	0.60	0.80	0.90	1.00
Activity #3 c.d.f.	0.00	0.10	0.48	0.80	0.90	1.00
Activity #3 p.d.f.	**0.00**	**0.10**	**0.38**	**0.32**	**0.10**	**0.10**

distribution of duration. For example, consider the network depicted in Figure 3a. The initial network looks ungainly, but arrows 1 and 2 are readily observable to be in series. Reducing the distributions of arrow 1 and 2 into a distribution represented by arrow 8 yields the network shown in Figure 3b. Further series reduction is possible on arrows 5, 6, and 7. Reduction is accomplished by first reducing via the series operation two consecutive arrows, say 5 and 6, into arrow 9, then combining it with the third activity, arrow 7, to derive activity 10's distribution. Figure 3d illustrates the resulting network. The distributions of arrows 4 and 10 are then reducible by the parallel operation. The result is Figure 3e where Arrow 11 replaces the previous pair. A series reduction is performed on arrows 3 and 11 in Figure 3e, and a parallel operation is performed on the network of Figure 3f to yield the single distribution of Arrow 13 shown in Figure 3g.

In general, network reductions are best performed by reducing all activities in series before identifying and performing parallel reductions. For one reason, projects are rarely estimated with parallel activities -- they would be combined into only one activity initially and estimated on that basis. Often, however, there are parallel paths, with one or more paths consisting of activities in series. For another reason, series are easier to detect. They are detectable by finding nodes that have one entering and one exiting arrow. A pair of arrows are identified as parallel when they both have the same starting

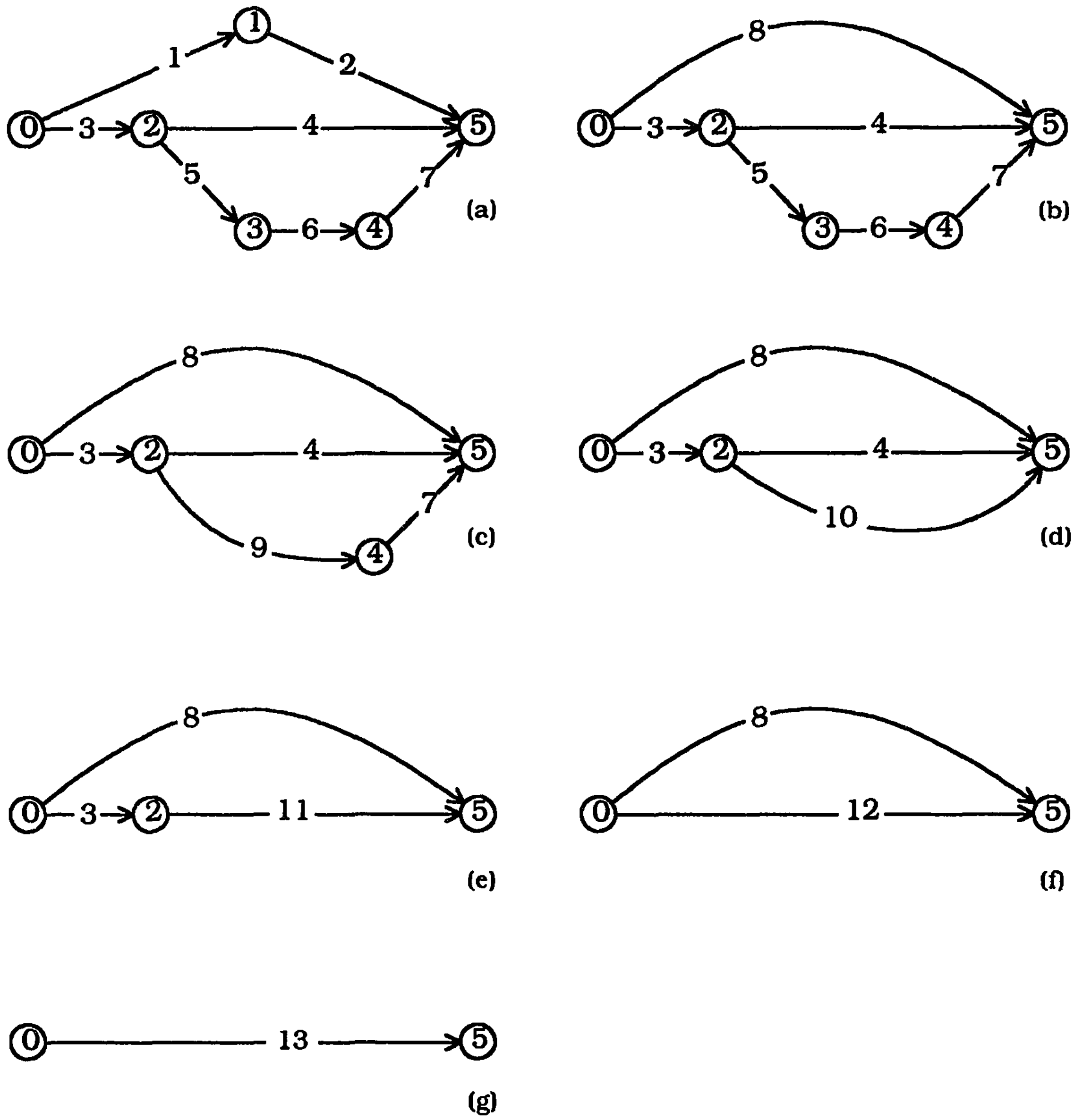


Figure 3: Parallel – Series Reduction Example.
(a) Unreduced Network. (b) Series Reduced Network. (c) Series Reduced Network.
(d) Series Reduced Network. (e) Parallel Reduced Network. (f) Series Reduced Network.
(g) Parallel Reduced Network to Project equivalent Distributed Arrow.

and ending nodes. This is a more intensive search procedure. An automated parallel-series reduction program should perform all series operations, search for a parallel arrow, if one is not found quit; otherwise, process one parallel reduction and begin again searching for series reductions. Each series operation reduces the number of nodes by one and the number of arrows by one. Each parallel operation reduces the number of arrows by one, but does not affect the number of nodes in the network.

In this manner, large networks are reducible to usually much smaller, but potentially complex ones. The simplest nontrivial network that is irreducible is the Wheatstone Network (see Figure 4a). The next section addresses how to reduce the wheatstone and other networks irreducible by series-parallel operations.

Network Reduction By Conditioning

Calculation of distributions in networks that can not be reduced by either parallel or series reductions requires a more robust technique: conditioning. Short of looking at all combinations of activity durations, conditioning is the only statistical method that will completely enumerate a network, and the number of duration combinations of a network can quickly become unmanageable. To condition a network, some arrows are identified for <u>conditioning</u> and others are <u>conditioned</u>. The conditioning arrows are all assigned durations from their distributions until all combinations of durations for the conditioning arrows have been assigned. A probability is associated with each combination of conditioning arrow durations. The conditioned arrow distributions are then, hopefully, easier to reduce. For every conditioning combination of durations, the reduced distribution durations from the conditioned network is multiplied by the combination probability and summed into the project duration distribution. In the limit, conditioning all arrows will result in analyzing all combinations of arrow durations. There are also methods that feature either sampling from the network distributions or setting limits on them, but these are only approximations. (See the Literature Review chapter for a discussion.)

The simplest network that requires conditioning is a network in the Wheatstone configuration (see Figure 4a). The wheatstone has 5 arrows, 4 nodes, and 3 paths. Examination of the three paths (see Figure 4b) reveals that Arrow 1 is common to Paths 1 and 2, and Arrow 5 is common to Paths 2 and 3. This makes both Path 1 and Path 3

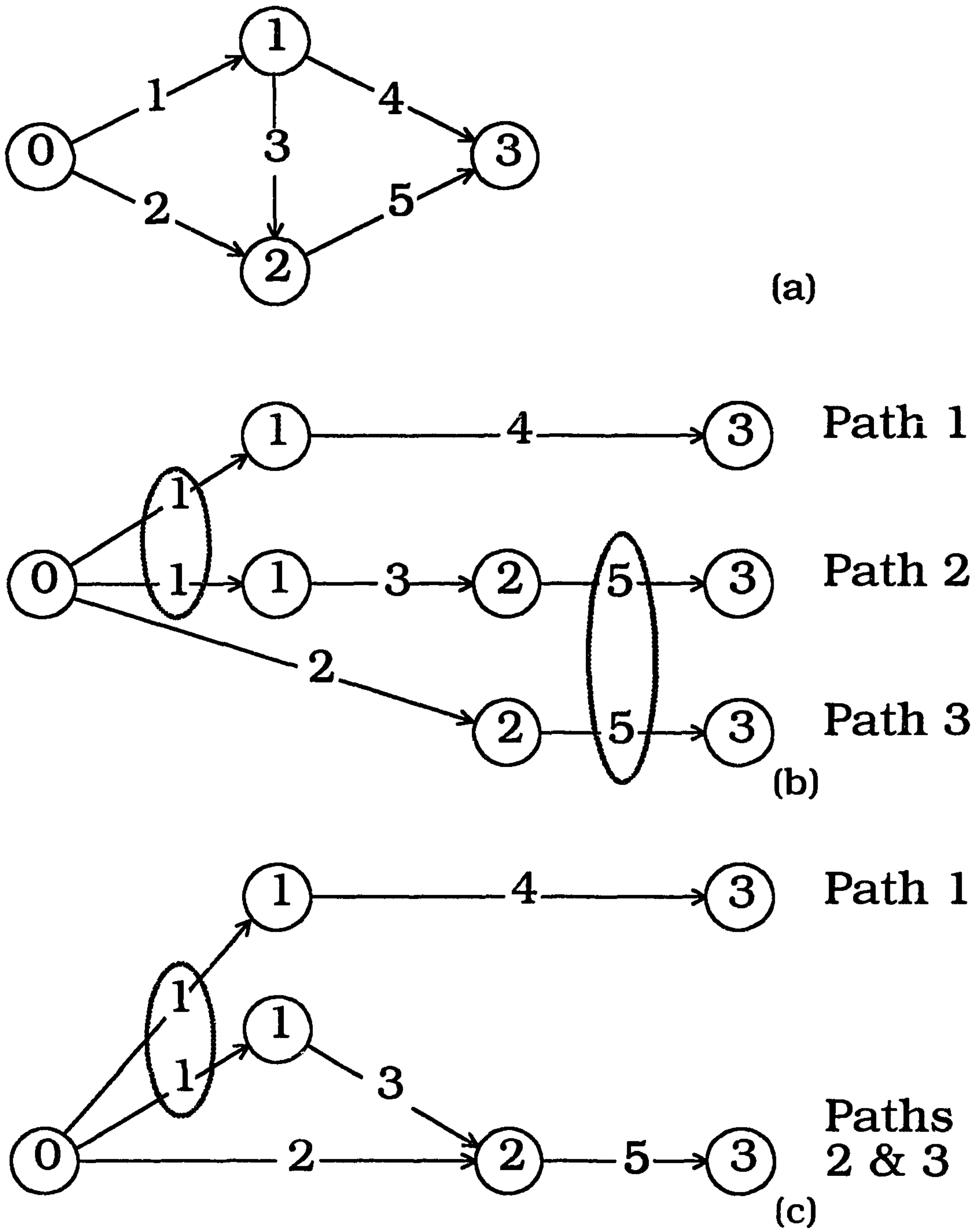


Figure 4: Wheatstone Network Configuration.
(a) The Wheatstone Configuration. (b) The 3 Paths.
(c) Path Configuration for Conditioned, Series, & Parallel Reduction.

correlated with Path 2 but not with each other.

Dodin (1985b) devised a method to calculate the network duration distribution, but the Wheatstone network exposes a flaw. The Dodin (1985b) method starts from node 0 of Figure 4. The duration distribution at Node 1 is the duration distribution of Arrow 1. Node 2, however, is the end of 2 independent paths. Therefore, its distribution is calculable by reducing the distribution of Arrows 1 and 3 by a series operation, then reducing the resultant distribution with that of Arrow 2's distribution by a parallel reduction operation. Under the Dodin method, the Arrow 4's distribution is reduced with Node 1's distribution by a series operation, and Arrow 5's distribution is combined with Node 2's distribution by another series reduction. The resulting distributions are combined by a parallel operation to estimate the distribution of Node 3. The problem is that the independence assumption of the parallel operation is violated -- the two incoming distributions at Node 3 are correlated by the variance of Arrow 1, unless Arrow 2's distribution dominates the distribution of series reduction of Arrows 1 and 3.

The problem is resolvable by conditioning on Arrow 1's duration distribution. Reorganizing the arrows into the configuration as shown in Figure 4c facilitates analysis. Specifically, Arrow 1 starts each of 2 pathsets because it is the path correlating arrow. Path 1 appears as before. The second pathset combines Paths 2 and 3 into a network skeleton which is in a parallel-series arrangement. Martin (1965) generated path trees as appears in Figure 4b, but until now the trees have not been recombined before the sink node. To analyze networks, conditioning fixes the durations of all arrows preceding network splits.

Once paths or pathsets are conditioned by fixing the values of conditioned arrows, distributions of the affected paths become independent and are calculable by using only parallel and series reduction operations. To complete the reduction, each conditioned distribution is weighted by the probability of the fixed values occurring and then summed. This is in accordance with the Law of Total Probability (Taylor & Karlin, 1984). The conditioning operation is theoretically capable of reducing any network. At the conceptual limit, conditioning would have to be performed on every arrow and every arrow's duration resulting in total enumeration. For even moderately large networks, this would be prohibitive. Luckily, conditioning on every arrow is rarely, if ever, necessary.

CRITICALITY CALCULATIONS OF NETWORKS

Project management has adapted CPM because it identifies through critical paths, the arrows of a network that represent critical activities. The previous chapter addressed PERT's problem of estimating the duration of a project when arrows have probabilistic durations. This chapter will address the CPM problem of identifying which arrows contribute to the project duration; but unlike CPM, the network arrows will be assumed to have a distribution of durations.

CPM takes deterministic input and determines not only how long a project will take, but also critical arrows, via paths, to meeting the schedule. A problem arises with CPM when the activities modeled by its arrows do not exhibit predictable durations even though the activity methods are standardized. The deterministic inputs of CPM become uncertain. Naturally, the degree of uncertainty is expressed by a probability distribution. Because the input parameters of CPM are uncertain as to their duration, the results of CPM also becomes uncertain. A critical arrow identified by CPM at given inputs is not necessarily critical for other inputs. Therefore, there is uncertainty of whether arrows will be critical to the project. Other arrows may become critical. To express this uncertainty, probabilities are assigned to the arrows to indicate how often they will be critical to the project. This is the definition of Criticality:

$$\text{Criticality} \equiv \Pr \text{ (an arrow will be on one or more critical paths).} \qquad (5)$$

As Van Slyke states it, "this index is simply the probability that the activity [arrow] will be on the [a] critical path" (1963, p. 839).

In developing methodologies for analyzing arrow criticalities, a number of assumptions are made. One is that the project and project methods will have been designed and estimated. Fixed methods on the activities keeps the arrow duration distributions stationary (Taylor & Karlin, 1984) through time. Every arrow identified must be performed with probability 1.00 and no additional arrows are performed. All arrows must be performed exactly once and then only after their precedent arrows are completed. The network structure does not change. Also, the distributions are assumed to be estimated in discrete probabilistic terms. Continuous distributions may have been used to approximate the distribution of times, but the results will have been converted into discrete units.

Before delving into arrow criticality calculations, 2 other network analysis methods will be presented. One will be an easy way to count the number of paths through complex networks, and the other will be how to identify all of the paths in a network. Both will be illustrated on the 40-arrow Kleindorfer (1971) network. Path criticality calculations are presented in four categories progressing in complexity. The result is a pair of equations that provide probabilistically correct Van Slyke criticality calculations for parallel-series networks like Martin (1965) did for reductions of parallel-series network duration distributions. Finally, a methodology for calculating arrow criticalities is developed for complex systems requiring conditioning.

Path Identification

Fundamental to the analysis of criticalities is predicting how many paths there are through a network and then finding all of the paths predicted. Quantifying how many paths that are in large networks can be very confusing. Dodin (1985b, p. 252) claims "the identification of all paths ... can be a burdensome task." The following section will present both a method for counting the number of paths in a network and a method for identifying the paths.

<u>Counting the Number of Network Paths</u> Since every scheduling network has one source node and one sink node, there is only one path entering a network, and only one path exiting a network. A pure series network, such as shown in Figure 1a, has only one path. The parallel network shown in Figure 2a, on the other hand has 2 paths through it. The path starting at the source node is first split into the two network paths, and then are recombined or merged at the sink node. Because every arrow has both a source node and an ending node, the total number of arrows leaving all nodes must equal the total number of arrows ending at all nodes.

Paths can be counted by tracking splits and mergers. A <u>split</u> occurs when a node starts more than one arrow; whereas a <u>merge</u> occurs when a node ends more than one arrow. For every node in a network, there is one or more paths that can reach it. So long as the network does not change its configuration, this number will always be the same. Splits from a node have the effect of duplicating the path counts to that node. Each arrow in the split has the same number of path combinations to reach the end of the arrow as it took to reach its starting node. Merges, on the other hand, add the combinations of path counts from the starting nodes of all merging arrows. This suggests an algorithm.

The algorithm is illustrated on the Kleindorfer (1971) 40–arrow network shown in Figure 5. The algorithm procedure is presented in Table 3. The first column of Table 3 lists all of the nodes in the network. The second column lists all of the nodes that immediately precede the first column's nodes. The third column lists the path counts to the node listed in column 1. Node 0 is assigned a path count of "1" because it represents the single entry into the network. Next, a search is conducted to find an eligible node to assign a path count. An eligible node does not already have a path count, but all of its predecessor nodes do have path counts. If all nodes have assigned path counts, the number of paths through the network is read from the path count assigned to the sink node, and the algorithm ends. A path count for an eligible node is calculated by summing the path counts from all of the node's immediate precedent nodes. Column 3 in Table 3 also gives the summed path counts.

The algorithmic path count of network paths can also start from the sink node rather than the source node. The node path counts are sums of path counts from the Target nodes rather than preceding nodes, and the network path count is read from the source node rather than the sink node. The backward analysis of the Kleindorfer network is shown in the last 3 columns of Table 3. The network path counts are the same for both the forward and backward analysis. The similarity is only applicable for the source and sink node path counts; the intermediate node path counts do not generally agree.

Identifying Network Paths To examine which arrows are on critical paths, it is imperative to identify the paths and to list the arrows on each path. As developed in the last subsection, the number(s) of splits and mergers at nodes determine the number of network paths. Martin (1965) modeled network paths as a tree. Each path on the tree started with the source node and ended with the sink node. Other than at the source node, the paths did not intersect, duplicating nodes and arrows on each path as needed. As a result, the number of sink nodes on the tree match the number of paths. The duplicating of nodes for each path is similar to the duplication of nodal path counts for

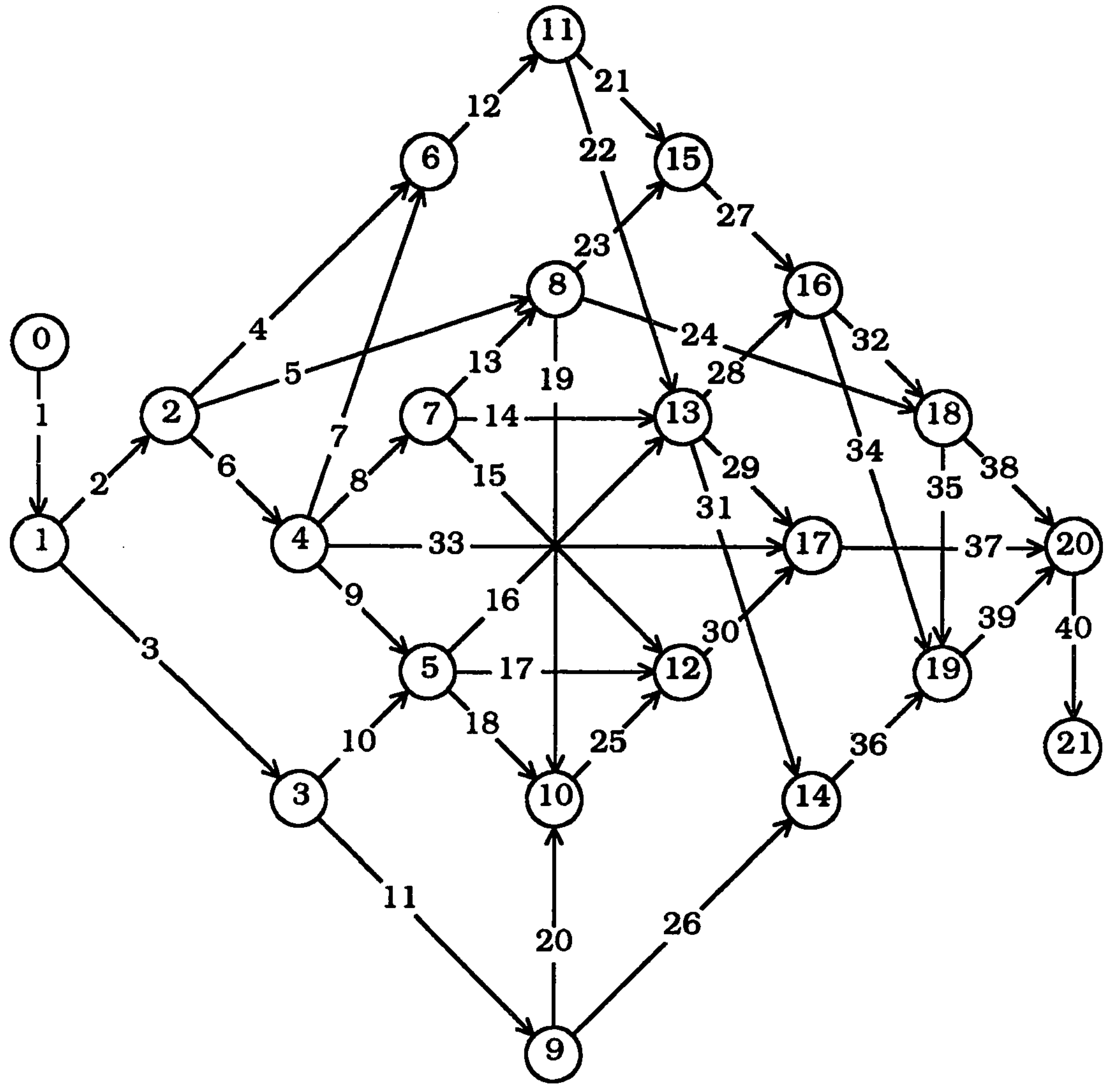


Figure 5: The 40 Arrow, 22 Node Kleindorfer (1971) Network.

Table 3: Path Counts of Kleindorfer (1971) 40-Arrow Network.

Forward Path Counting			Backward Path Counting		
Node	Precedent Nodes	Path Count	Node	Target Nodes	Path Count
0	None	1	21	None	1
1	0	1	20	21	1
2	1	1	19	20	1
3	1	1	18	19,20	1+1=2
4	2	1	17	20	1
5	3,4	1+1=2	16	18,19	2+1=3
6	2,4	1+1=2	15	16	3
7	4	1	14	19	1
8	2,7	1+1=2	13	14,16,17	1+3+1=5
9	3	1	12	17	1
10	5,8,9	2+2+1=5	11	13,15	5+3=8
11	6	2	10	12	1
12	5,7,10	2+1+5=8	9	10,14	1+1=2
13	5,7,11	2+1+2=5	8	10,15,18	1+3+2=6
14	9,13	1+5=6	7	8,12,13	6+1+5=12
15	8,11	2+2=4	6	11	8
16	13,15	5+4=9	5	10,12,13	1+1+5=7
17	4,12,13	1+8+5=14	4	5,6,7,17	7+8+12+1=28
18	8,16	2+9=11	3	5,9	7+2=9
19	14,16,18	6+9+11=26	2	4,6,8	28+8+6=42
20	17,18,19	14+11+26=51	1	2,3	42+9=51
21	20	**51**	0	1	**51**

each arrow leaving a split. Constructing a table of network paths requires a method to generate all paths in a network from the network splits. A path will be defined by its sequence of arrows.

Tabulation of network paths begins, of course, with the source node. To track splits in the network, an initial path needs to be recorded from which splits can occur. An easy way to generate this initial path is to take the lowest numbered arrow leaving every node. For example, the initial path for the 40-arrow Kleindorfer (1971) network shown in Figure 5 consists of Arrows 1, 2, 4, 12, 21, 27, 32, 35, 39, and 40 and includes Nodes 0, 1, 2, 6, 11, 15, 16, 18, 19, 20, and 21. This is the first path shown in Table 4.

There are 6 splits from this initial path. For each split, a new path is started in Table 4. Each new path duplicates the partial path that led up to its creating split, then adds the arrow numbers of all the splits from the path whose number is Index #1. Index #1 indicates the path number of the path currently being examined for splits. To account for how many paths have been started, Index #2 is maintained to show the path number of the next path to be started. After a path has been examined, and it has reached the sink node, Index #1 is incremented by one. Then the row indicated by Index #1 starts at the ending node of the last recorded arrow to examine it for further splits. The default path through the network is always the sequence of arrows with the lowest numbers, starting with the last node reached for the beginning of a path. After the initial path is examined Index #1 is "2" to start examining Path 2, and Index #2 is 8 to show that there have been 7 paths started and that Path 8 would be the next one to be started. In Table 4, the arrow numbers appearing in italics are the last recorded arrows before the path is reached to complete its examination for splits. The path number that generated the new path when it was examined is given in Column 2, entitled "Row Source." The arrow numbers after the one in italics is the sequence of arrows that have the lowest arrow number leaving each subsequent node.

For the network in Figure 5, Node 1 is the first splitting node with two arrows starting with it. The lowest numbered arrow is Arrow 2, and the other is Arrow 3. Arrow 2 has been listed in the initial path. Arrow 3 starts a new path. To record the start of this new path, the path leading up to Node 1 is recorded, and then Arrow 3 is added to the end of the sequence. The next node along the initial path is Node 2. It starts 3 arrows, but one of those, Arrow 4, is on the original path. The other two, Arrows 5 and 6, start new paths. The 2 new path starts should be recorded as Arrow sequences 1-2-5 and 1-2-6.

Table 4: Paths of the 40-Arrow Kleindorfer (1971) Network.

Path	Row	Ordered Path Arrow Numbers											Index	
Number	Source	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th	11th	#1	#2
1	Step 2	*1*	*2*	*4*	*12*	*21*	*27*	*32*	*35*	*39*	*40*		2	8
2	1	1	*3*	10	16	28	32	35	39	40			3	15
3	1	1	2	*5*	19	25	30	37	40				4	17
4	1	1	2	*6*	7	12	21	27	32	35	39	40	5	23
5	1	1	2	4	12	*22*	28	32	35	39	40		6	27
6	1	1	2	4	12	21	27	*34*	39	40			7	27
7	1	1	2	4	12	21	27	32	*38*	40			8	27
8	2	1	3	*11*	20	25	30	37	40				9	28
9	2	1	3	10	*17*	30	37	40					10	28
10	2	1	3	10	*18*	5	30	37	40				11	28
11	2	1	3	10	16	*29*	37	40					12	28
12	2	1	3	10	16	*31*	36	39	40				13	28
13	2	1	3	10	16	28	*34*	39	40				14	28
14	2	1	3	10	16	28	32	*38*	40				15	28
15	3	1	2	5	*23*	27	32	35	39	40			16	30
16	3	1	2	5	*24*	35	39	40					17	31
17	4	1	2	6	*8*	13	19	25	30	37	40		18	35
18	4	1	2	6	*9*	16	28	32	35	39	40		19	41
19	4	1	2	6	*33*	37	40						20	41
20	4	1	2	6	7	12	*22*	28	32	35	39	40	21	45
21	4	1	2	6	7	12	21	27	*34*	39	40		22	45
22	4	1	2	6	7	12	21	27	32	*38*	40		23	45
23	5	1	2	4	12	22	*29*	37	40				24	45
24	5	1	2	4	12	22	*31*	36	39	40			25	45
25	5	1	2	4	12	22	28	*34*	39	40			26	45
26	5	1	2	4	12	22	28	32	*38*	40			27	45
27	8	1	3	11	*26*	36	39	40					28	45
28	15	1	2	5	23	27	*34*	39	40				29	45
29	15	1	2	5	23	27	32	*38*	40				30	45
30	16	1	2	5	24	*38*	40						31	45
31	17	1	2	6	8	*14*	28	32	35	39	40		32	49
32	17	1	2	6	8	*15*	30	37	40				33	49
33	17	1	2	6	8	13	*23*	27	32	35	39	40	34	51
34	17	1	2	6	8	13	*24*	35	39	40			35	52
35	18	1	2	6	9	*17*	30	37	40				36	52
36	18	1	2	6	9	*18*	25	30	37	40			37	52
37	18	1	2	6	9	16	*29*	37	40				38	52
38	18	1	2	6	9	16	*31*	36	39	40			39	52
39	18	1	2	6	9	16	28	*34*	39	40			40	52
40	18	1	2	6	9	16	28	32	*38*	40			41	52
41	20	1	2	6	7	12	22	*29*	37	40			42	52
42	20	1	2	6	7	12	22	*31*	36	39	40		43	52
43	20	1	2	6	7	12	22	28	*34*	39	40		44	52
44	20	1	2	6	7	12	22	28	32	*38*	40		45	52
45	31	1	2	6	8	14	*29*	37	40				46	52
46	31	1	2	6	8	14	*31*	36	39	40			47	52
47	31	1	2	6	8	14	28	*34*	39	40			48	52
48	31	1	2	6	8	14	28	32	*38*	40			49	52
49	33	1	2	6	8	13	23	27	*34*	39	40		50	52
50	33	1	2	6	8	13	23	27	32	*38*	40		51	52
51	34	1	2	6	8	13	24	*38*	40				52	52

The new path recordings should proceed for the splits occurring at Node 11, at Node 16, and at Node 18.

By completely examining the splits off all path examined, all paths in the network are generated. After examining a path, if Index #1 (after incrementing) and Index #2 are equal, all paths in the network will have been examined and tabulated.

Criticality Definitions

The next step after counting and identifying paths through a network is to establish how each of the path arrows might contribute to the overall network completion time. Van Slyke's criticality (1963) is defined for arrows as the probability that an arrow is on a critical path. A critical path is a path whose duration equals the duration of the total network. If the path's duration is increased, the network's duration is increased by the same amount. Because of the nature of project durations, it is not possible for the path's duration to be greater than the project's duration. It can only be less than or equal to the path's duration.

Derivations of equations to calculate Van Slyke's Criticality proceed through the increasingly complex networks of series networks, parallel networks, parallel-series networks, and complex networks (networks that require conditioning to reduce its duration distribution).

Van Slyke's Criticality In A Series Network

When there is only one path through a network (see Figure 1a), it must always be critical. The duration of the network is determined exclusively by its only path. The path's duration is the sum of all of the durations from the series of arrows which constitute the path. Since all arrows contribute to the project duration, Van Slyke's criticality for arrows must be equal to the probability that the path is critical. The probability that the path is critical is 1.00, and the Van Slyke's criticality for the arrows must be 1.00 also.

The Van Slyke criticality for a single path also applies to a pure series of arrows. A pure series of arrows contain nodes (not the beginning or ending node) that do not have splits or mergers. The series may make an entire path from source node to sink node, or the series may be only a portion of a path.

Van Slyke's Criticality In A Parallel Network

The 2 paths of the parallel network shown in Figure 2a are assumed to be independent. Dodin & Elmaghraby (1985) used this fact to calculate the criticality for one of the paths, say Arrow 1. They calculated the criticality for each duration of Arrow 1 by multiplying the probability of that duration by the cumulative distribution of Arrow 2 at the same duration. $Crit_{A1_i} = f_{A1_i}F_{A2_i}$. Ross (1980) showed that this equation calculates the probability that one distribution variable is greater than or equal to another independent variable. The equation explains why longer paths dominate shorter paths. For any duration, the Van Slyke criticality of an arrow (or path) depends on whether its duration is greater than or equal to the durations of the competing arrows (or paths), and the probability that Arrow 1 assumes a duration i is f_{A1_i}, while the probabilities that Arrow 2 does not have a longer duration is F_{A2_i}. Finally, the total Van Slyke criticality of Arrow 1 is the sum of all of the terms $f_{A1_i}F_{A2_i}$:

$$Crit_{A1} = \sum_{i \in A1} f_{A1_i}F_{A2_i}. \qquad (6)$$

For networks with many independently distributed parallel paths, the Dodin & Elmaghraby (1985) calculation can be expanded into

$$Crit_{A1} = \sum_{i \in A1} f_{A1_i}F_{A2_i} \bullet\bullet\bullet F_{An_i}. \qquad (7)$$

A better method than using all of the independent distributions in the calculation is to only use the distribution of the arrow of interest and the distribution of the overall network. Recall that the product of cumulative distributions was used in the calculation of the network duration distribution. By using this fact, the cumulative probabilities do not need to be repeatedly multiplied to find each arrow's criticality. By using this fact, the Van Slyke arrow criticalities can be calculated from each arrow's cumulative duration distribution and the network's cumulative duration distribution, instead of from all the cumulative duration distributions from all parallel paths.

Expanding on this idea, consider Figure 2, and the arrow numbers assigned in Figure 2 to designate distributions. The cumulative network duration distribution is $F_{A3_i} = F_{A1_i} F_{A2_i}$. Here, A2 may be taken to represent a collection of paths parallel to A1. Now let F_{A2_i} represent the cumulative probability distribution of A2, evaluated at Arrow 1 duration i. Then $F_{A2_i} = \frac{F_{A3_i}}{F_{A1_i}}$. Substituting this equation into the Dodin & Elmaghraby (1985) equation reduces the number of distributions needed to calculate the criticality of one path amongst many independent paths to only 2 -- the duration distributions of the arrow and of the total network:

$$\mathrm{Crit}_{A1} = \sum_{i \in A1} f_{A1_i} \frac{F_{A3_i}}{F_{A1_i}} = \sum_{i \in A1} \left(F_{A1_i} - F_{A1_{i-1}} \right) \frac{F_{A3_i}}{F_{A1_i}} . \tag{8}$$

This equation will henceforth be called the Parallel Criticality Equation. The distribution of Arrow 3, the network's duration distribution, may be calculated by any means and the equation will still be valid so long as the path modeled by Arrow 1 is independent of (and of course parallel with) the rest of the network.

Van Slyke's Criticalities In A Parallel-Series Network

Methods to calculate Van Slyke's criticalities in parallel-series networks (see, for example, Figure 3) must encompass not only arrows in series, and arrows in parallel, but also parallel paths with splits in them. In parallel-series networks, the series (see Figure 1) and parallel (see Figure 2) configuration of arrows comprise the entirety of the networks. As such, the duration distributions for the networks are calculable by repeating the series reduction operations and the parallel reduction operations as necessary. Calculating the Van Slyke criticality, on the other hand, is more complicated. For the simplest of parallel-series networks (see Figure 6) the Parallel Criticality Equation is all that is required to calculate the Van Slyke criticality, but for the only slightly more complex network shown in Figure 7a, arrows 2 and 3 require an additional Van Slyke criticality calculation, as discussed below.

The simplest of parallel-series networks (see Figure 6) have one set of parallel paths and an arrow in series either within one of the parallel paths (see Figure 6a) or in sequence to the parallel paths (see Figure 6b). For Figure 6a, the Van Slyke criticality for Arrow 3 is calculated from the Parallel Criticality Equation. The Van Slyke criticality for Arrows 1 and 2 in Figure 6a is also calculated using the Parallel Criticality Equation, but the arrow duration distributions must first be reduced by the series operation. Arrow 1 and 2's Van Slyke criticality is the same as that for the path they form since there are no splits within the path.

In Figure 6b, Arrow 1 has a criticality of 1.00 because no matter what combination of Arrow 2's and 3's durations determine the duration from Node 1 to Node 2, the duration of Arrow 1 will always be added to it to determine the duration of the project. The Van Slyke criticalities of Arrows 2 and 3 in Figure 6b are calculated by the Parallel Criticality Equation with the arrow's duration distribution and the distribution from the parallel reducing operation between Arrows 2 and 3. It is important to note that the criticality of Arrow 2 may not be calculated by the Parallel Criticality Equation using the distribution of the path consisting of Arrow 1 and Arrow 2 against the distribution of the network because the path has a split at node 1 (Node 1 would have a merge if the arrow directions were reversed).

The Parallel Criticality Equation assumes an independent path, and the path made-up of Arrows 1 and 2 in Figure 6b is not independent because Arrow 1 is also a member of the path made-up of Arrows 1 and 3. The shapes of the distributions from Arrow 2 and 3 will remain constant relative to each other. Only the absolute value of their durations will change. Criticality depends on the shape and relative distance between distributions, not the absolute value of their durations.

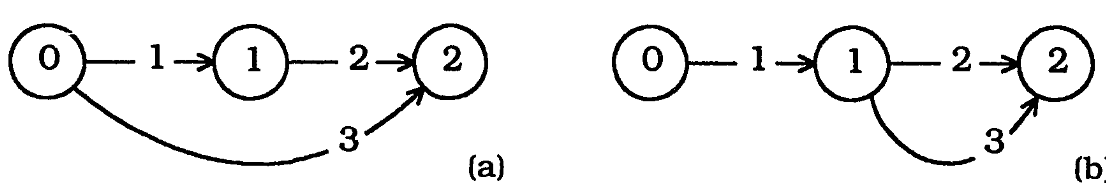


Figure 6: The Simplest of Parallel-Series Networks.
(a) Two activities in series paralled by a third.
(b) One activity followed in series by two in parallel.

To calculate the Van Slyke criticality of the arrows in Figure 7, the duration distributions of the arrows in the network must first be reduced to the project's duration distribution via parallel and series reduction operations. The parallel-series reduction of the network in Figure 7 has been illustrated in Figure 7a, 7b, 7c, and 7d, and the distributions identified in Figure 7 are given in Table 5. The network in Figure 7a is identical in form to the network in Figure 3d.

The Van Slyke criticality of Arrows 1, 4, 5, 6, and 7 are easily calculated. Arrow 7's criticality is 1.0000. Arrows 4 and 6 are calculated from the Parallel Criticality Equation. As listed in Table 5, the criticality of Arrow 4 is 0.8760, and Arrow 6's criticality is 0.5680. Arrows 1 and 5 must have the same criticality as Arrow 6 because they are the series of arrows that make up arrow 6, and there are no splits from Node 1 in Figure 7b.

The Parallel Criticality Equation does not work on Arrows 2 and 3 in Figure 7a. It is tempting to use the Parallel Criticality Equation to calculate the probabilities that Arrow 2's duration (or Arrow 3's) meets or beats Arrow 3's Duration (Arrow 2's) then multiply the probability by Arrow 1's Van Slyke criticality to calculate Van Slyke criticality of Arrow 2 for the whole network, but this does not work. The reason is that the distribution at Node 2 is not wholly dependent upon the varying durations of Arrow 1. Node 2 also depends upon the varying durations of Arrow 4. The result is that the criticalities represented by Arrow 2 and 3 depends upon the value of Arrow 1.

The derivation of an equation to calculate the Van Slyke criticalities for Arrows 2 and 3 in Figure 7 will start with the 2 arrows and proceed by including more and more of the network until the whole network is encompassed. The equation to calculate the Van Slyke criticality will be developed for Arrow 2, the development being the same for Arrow 3.

The duration distributions of Arrows 2 and 3 from Figure 7a are reduced into the duration distribution of Arrow 5 in Figure 7b by the parallel operation: $F_{A5_i} = F_{A2_i} F_{A3_i}$. If Arrow 5 represented the entire network, the Van Slyke criticality of Arrow 2 would be $Crit_{A2} = \sum_{i \in A2} f_{A2_i} \left[\frac{F_{A5_i}}{F_{A2_i}} \right]$ by the Parallel Criticality Equation. Substituting for the duration of Arrow 5, the Van Slyke criticality becomes $Crit_{A2} = \sum_{i \in A2} f_{A2_i} \left[\frac{F_{A2_i} F_{A3_i}}{F_{A2_i}} \right] = \sum_{i \in A2} f_{A2_i} F_{A3_i}$ which is the form used by Dodin & Elmaghraby (1985) and is equation (6) applied to Arrow 2 and Arrow 3. Within the network of Figure 7, the Van Slyke criticality of Arrow 5 is equal to the Van Slyke criticality of Arrow 1 and is equal to the Van Slyke criticality of

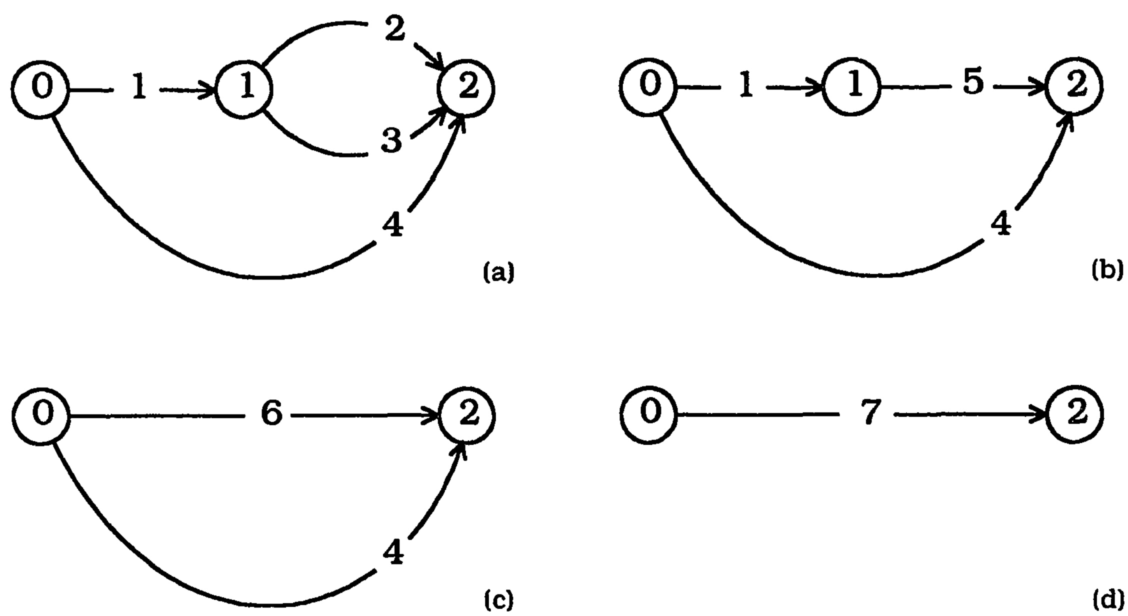


Figure 7: Parallel-Series Criticality Example.
(a) A parallel-series network. (b) Same network reduced by a parallel reduction.
(c) After a series reduction. (d) After another parallel reduction.

Table 5: Probabilities for Parallel-Series Van Slyke Criticality Example.

	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Criticality
Arrow 1 p.d.f.	0.800	0.200					0.5680
Arrow 2 p.d.f.			0.600	0.400			0.5680
Arrow 3 p.d.f.		0.300	0.700				0.1176
Arrow 4 p.d.f.				0.100	0.900		0.8760
Arrow 5 p.d.f.			0.600	0.400			0.5680
Arrow 6 p.d.f.				0.480	0.440	0.080	0.5680
Arrow 7 p.d.f.				0.048	0.872	0.080	1.0000
Arrow 6 c.d.f.				0.480	0.920	1.000	
Arrow 7 c.d.f.				0.048	0.920	1.000	

Arrow 6 because Arrows 1 and 5 form the path of Arrow 6: $Crit_{A5} = Crit_{A1} = Crit_{A6}$.

Again, from the Parallel Criticality Equation, $Crit_{A6} = \sum_{k \in A6} f_{A6_k} \left[\frac{F_{A7_k}}{F_{A6_k}} \right]$. The duration distribution of Arrow 6 is defined by the series reduction operation on Arrows 1 and 5. Substituting "i+j" for the "k" in the Van Slyke criticality equation for Arrow 6, seen as a series network composed of Arrows 1 and 5, a new equation is derived:

$Crit_{A6} = \sum_{i \in A1} \sum_{j \in A5} \left\{ f_{A1_i} f_{A5_j} \left[\frac{F_{A7_{i+j}}}{F_{A6_{i+j}}} \right] \right\}$. The Van Slyke criticality of Arrow 2 is found by substituting its Van Slyke criticality for each of Arrow 5's durations for the probability of that duration which appears in the Van Slyke criticality equation for Arrow 5.

$$Crit_{A2} = \sum_{i \in A1} \sum_{j \in A2} \left\{ f_{A1_i} f_{A2_j} F_{A3_j} \left[\frac{F_{A7_{i+j}}}{F_{A6_{i+j}}} \right] \right\}. \quad (9)$$

By rearranging the equation for the Van Slyke criticality of Arrow 2 in Figure 7, it is more interpretable.

$$Crit_{A2} = \sum_{j \in A2} f_{A2j} F_{A3_j} \left\{ \sum_{i \in A1} f_{A1_i} \left[\frac{F_{A7_{i+j}}}{F_{A6_{i+j}}} \right] \right\}. \quad (10)$$

For Arrow 2 to be on a critical path, its duration must be at least as great as the duration from Arrow 3. The first two factors of the first summation states the probability that a duration of Arrow 2 occurs and reduces it by the probability that it does not beat Arrow 3. The inside summation finds the criticality of the rest of the path after the path duration is found by adding the duration Arrow 1 to the duration of Arrow 2. In effect, this term untangles the series reduction operation. This equation will be referred to as the Dependent-Parallel Criticality Equation.

Consider for example the distributions given in Table 5 for the parallel-series network in Figure 7. Equation (10), written out explicitly for this example would be:

$$Crit_2 = f_{2_{j=3}} F_{3_{j=3}} \left[f_{1_{i=1}} \left(\frac{F_{7_{i+j=4}}}{F_{6_{i+j=4}}} \right) + f_{1_{i=2}} \left(\frac{F_{7_{i+j=5}}}{F_{6_{i+j=5}}} \right) \right] + f_{2_{j=4}} F_{3_{j=4}} \left[f_{1_{i=1}} \left(\frac{F_{7_{i+j=5}}}{F_{6_{i+j=5}}} \right) + f_{1_{i=2}} \left(\frac{F_{7_{i+j=6}}}{F_{6_{i+j=6}}} \right) \right].$$

The numeric calculations are

$$Crit_2 = 0.600 \bullet 1.000 \left[.800 \left(\frac{.048}{.480} \right) + .200 \left(\frac{.920}{.920} \right) \right] + 0.400 \bullet 1.000 \left[.800 \left(\frac{.920}{.920} \right) + .200 \left(\frac{1.000}{1.000} \right) \right]$$

$$Crit_2 = 0.600[.080 + .200] + 0.400[.800 + .200]$$

$Crit_2 = .0.600[0.280] + .0400[1.000] = .0.168 + .0.400 = .568$. The criticality of Arrow 2 is the same as Arrow 1's criticality because the durations Arrow 2 is always at least as big as the durations of Arrow 3 as listed in Table 5. Therefore, Arrow 2 is always critical so long as Arrow 1 is critical. The criticality calculation for Arrow 3 is more interesting.

$$Crit_3 = f3_{j=2}F3_{j=2}\left[f1_{i=1}\left(\frac{F7_{i+j=3}}{F6_{i+j=3}}\right) + f1_{i=2}\left(\frac{F7_{i+j=4}}{F6_{i+j=4}}\right)\right] + f3_{j=3}F2_{j=3}\left[f1_{i=1}\left(\frac{F7_{i+j=4}}{F6_{i+j=4}}\right) + f1_{i=2}\left(\frac{F7_{i+j=5}}{F6_{i+j=5}}\right)\right]$$

$$Crit_3 = 0.300 \bullet 0.000\left[0.800\left(\tfrac{.000}{.000}\right) + 0.200\left(\tfrac{.048}{.480}\right)\right] + .700 \bullet .600\left[.800\left(\tfrac{.048}{.480}\right) + .200\left(\tfrac{.920}{.920}\right)\right]$$

$$Crit_3 = 0.000[0.000 + .020] + .420[.080 + .200] = 0.000[0.020] + .420[.280]$$

$$Crit_3 = 0.0000 + 0.1176 = 0.1176.$$

The first term of Arrow 3's criticality is zero because Arrow 3's lowest duration is always dominated by Arrow 2. The first term in the square bracket is zero by definition. Because the purpose of the calculation is to find the probability that an arrow will be on a critical path, and both values in the ratio are cumulative probabilities at the zero level, that means that the combination of Arrow 3's and Arrow 1's durations have not contributed to the path or project duration distribution. Hence the ratio of $\frac{0.000}{0.000} = 0.000$ to be consistent with intentions for the equation.

If Arrow 2 and 3 constituted the whole network, their criticalities would be 1.000 and 0.420 respectively. Notice that 1.000 multiplied by criticality of Arrow 1 yields the network value of Arrow 2's criticality. If, on the other hand, 0.420 were multiplied by the .568 of Arrow 1, the result is not equal to Arrow 1's network criticality ($0.420 * 0.568 = 0.23856 \neq .1176 = Crit_3$). This gives credence to the Dependent-Parallel Criticality Equation for parallel-series networks. The results of the equation are also verifiable by taking all combinations of the arrows in the sample. Since all 4 arrows in Figure 7 only have one decimal place for the probabilities assigned to them in Table 5, the probabilities for the combinations of the 4 arrows must have 4 decimals. The before mentioned equation yields probabilities with only 4 decimal places, but the pure parallel criticality equation multiplied by the path's criticality does not.

The parallel-series Van Slyke criticality equations, the Parallel Criticality Equation and the Dependent-Parallel Criticality Equation, are applicable in more places than would seem obvious. Each arrow depicted in Figure 7a may represent a number of arrows or subnetworks. If a network can be reduced into the form of Figure 7a (see, for example, the reduction of Figure 3a into Figure 3d.) the criticality of Arrow 2 (or 3) is calculated from the parallel-series criticality equation. Arrow 3 may have originated as another series-parallel network, a Wheatstone network, or some other more complex network. Criticality analysis of Arrow 2 requires only the distribution of Arrow 3 which can be a reduced distribution of a much larger network. Arrow 1 can represent the distribution of all arrows that come before and after the parallel Arrows 2 and 3. It too can be a reduced set of many arrows. The only requirement for Arrow 1 is for its distribution be known and that it constitutes the rest of the path with Arrows 2 and 3 to run parallel with Arrow 4. The distributions of Arrows 6 and 7 must also be known. They can be found by parallel and series reduction. Arrow 4's distribution is not required so long as the network distribution at node 2 is known. The equation calculates Arrow 4's distribution impact. The Dependent-Parallel Criticality Equation reduces into the Parallel Criticality Equation when the distributions of Arrow 1 and Arrow 4 in Figure 7 are both set to durations of zero with probabilities of 1.000.

The Dependent-Parallel Criticality Equation for the calculation of Van Slyke criticalities in parallel-series networks is the perfect complement to duration distribution reduction techniques introduced by Martin (1965) and used by Dodin (1985a, 1985b, & 1985c), Dodin & Elmaghraby (1985), and Burt & Garman (1971). There has not existed before now a procedure that could handle criticality calculations of networks with many arrows in parallel and series. The equation not only covers arrows in parallel and series but also explains criticalities when there is a mixture of the two.

Van Slyke's Criticality In Complex Networks

Like the duration distribution reduction operations, Van Slyke criticality calculations for networks with precedent structures more complex than arrows in parallel and series require conditioning. By conditioning on the arrows that precede splits in the network, the remaining arrows of the network are reduced into a merging parallel-series tree. A merging parallel-series tree is defined as a network that starts with independent arrows, ends in a single node, and is completely reducible using only the Parallel and Series Reduction Operations. The network is thus reduced into two categories -- the group of arrows whose durations are fixed by conditioning and the group of arrows that form a merging parallel-series tree. The arrows whose durations are fixed by conditioning will be called Conditioning Arrows, and the arrows formed into a merging parallel-series tree will also be referred to as Conditioned Arrows.

For example, refer to the conditioned Wheatstone network depicted in Figure 4c. Splits in the network of Figure 8a (identical with Figure 4a) are preceded by only one arrow, Arrow 1. Arrow 1 is designated as a conditioning arrow and shown in the "Conditioning Arrows" category in Figure 8b. By fixing the duration of Arrow 1 before adding it to the next arrows, the accumulated distributions of Arrow 3 and Arrow 4 are independent; whereas before conditioning they were not. Arrows 3 and 4 are shown in the "Conditioned Arrows" category of Figure 8b. Arrow 2 was already independent of Arrows 3 and 4. Adding Arrow 1's fixed duration to the duration distributions of Arrow 2 and Arrow 4 does not change the shape of their duration distributions.

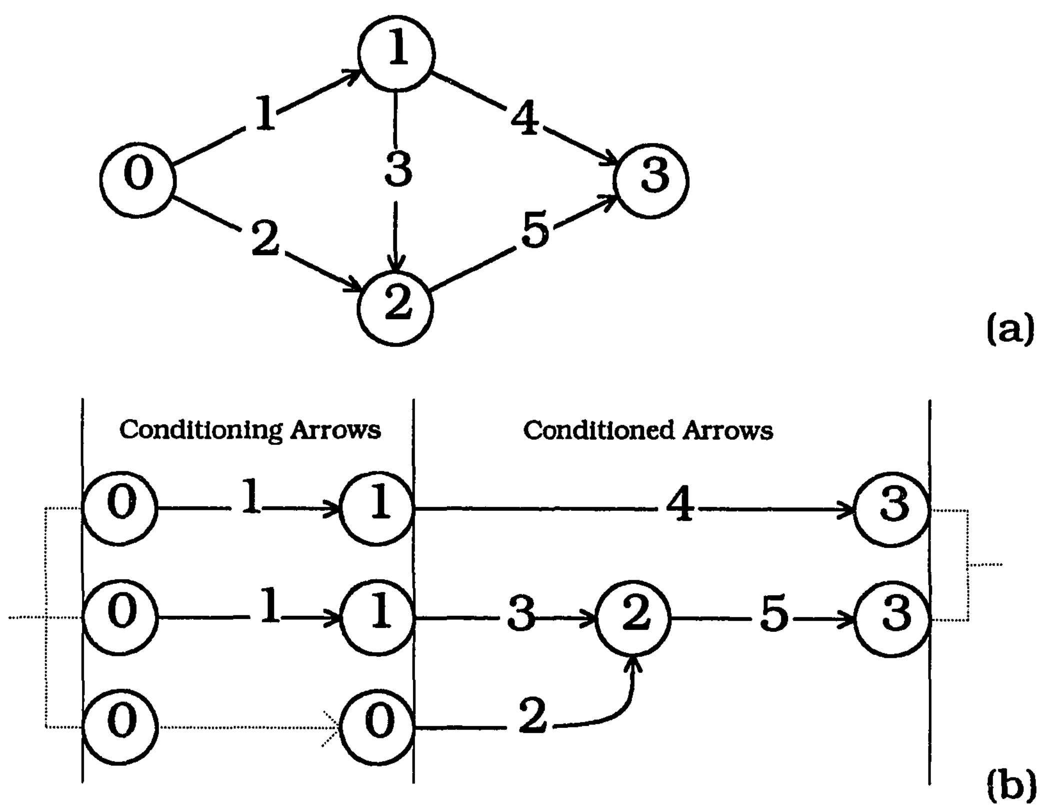


Figure 8: The Wheatstone Divided into Conditioning and Conditioned Arrows.
(a) The Wheatstone Configured Network.
(b) The Conditioning Arrows & the Merging Parallel-Series Tree.

Once the network of Figure 8 is conditioned by Arrow 1, the Van Slyke criticalities of Arrow 4 and Arrow 5 are easily calculated by the Parallel Criticality Equation. Arrows 2 and 3 require the Dependent Parallel Criticality Equation because they merge at node 2 which is not the sink node. Each Van Slyke criticality calculation for conditioned arrows is weighted by the probability of the conditioning arrow's duration. To find the total Van Slyke criticalities of Arrows 2, 3, 4, and 5, the weighted Van Slyke criticalities are summed over all conditioning duration combinations by the Law of Total Probability.

The Van Slyke criticality of Arrow 1 is calculated differently. For the Wheatstone network in Figure 8, there are 2 approaches to calculate the Van Slyke criticality. One method examines when Arrow 1 is not on a critical path $(1 - \mathrm{Crit}_{A1})$, and the other method calculates the Van Slyke criticality from probabilities in the merging parallel-series tree.

The Wheatstone, as shown in Figure 4b, has 3 paths. Arrow 1 starts 2 out of the 3 paths. If either of the 2 paths determine the network duration, Arrow 1 will be critical. If on the other hand, the third path, the one without Arrow 1 on it, has a larger duration than the paths with Arrow 1, then the Van Slyke criticality of Arrow 1 is 0.00. The Van Slyke criticality of the third path, the one with Arrows 2 and 5, is the same as the criticality of Arrow 2 since that is Arrow 2's only path. The Van Slyke criticality of the path with Arrow 2 includes probabilities that the path ties with one or more paths. By taking the instances when the third path is longer than the other 2, a probability without ties is calculated. The probability that Path 3 in Figure 4b is the longest of the 3 paths is the probability that Arrow 2's duration is longer than the sum of Arrow 1's duration and Arrow 3's duration and that the sum of durations from Arrows 2 and 5 is longer than the sum of durations from Arrows 1 and 4. The Van Slyke criticality of Arrow 1 using this method is

$$\mathrm{Crit}_{A1} = \sum_{i \in A1} f_{A1_i} \left\{ 1 - \left[\sum_{j \in A2} f_{A2_j} F_{A3z_{i+j-1}} \left(\sum_{k \in A5} f_{A5_k} F_{A4z_{i+j+k-1}} \right) \right] \right\} \tag{11}$$

where "z" in the subscripts designate a distribution calculated through the end of the designated Arrow. The designation "A3z" designates the distribution resulting from adding the conditioned duration of Arrow 1 to the durations in Arrow 3's distribution, and "A4z" designates the distribution resulting from adding the conditioned duration of

Arrow 1 to the durations in Arrow 4's distribution. Because the Wheatstone network only has 1 path on which Conditioning Arrow 1 is not a member, the calculation equation is easily expressed. For more complex networks, the equation will only be more complicated. Remember, that the Wheatstone network is the simplest of networks that require conditioning. This complementary method to calculate the Van Slyke criticality can become untenable as the numbers of alternative paths to the one the conditioning arrows are on become numerous.

The other method observes that the influence of Arrow 1 on the project duration is manifested through Arrows 3 and 4. It should, therefore, be possible to calculate the Van Slyke criticality of Arrow 1 from the Van Slyke criticalities of Arrows 3 and 4. The Van Slyke criticality of Arrow 1 can not be the sum of the Van Slyke criticalities of Arrows 3 and 4 because sometimes both are on a critical path simultaneously. The probability of both Arrows 3 and 4 simultaneously being on a critical path is found by the intersection of their Van Slyke criticalities. Criticalities are probabilities after all. If the intersection of the criticalities is found, then the Van Slyke criticality for Arrow 1 would be

$$\boxed{Crit_{A1} = Crit_{A3} + Crit_{A4} - Crit_{A3 \cap A4}\,.} \tag{12}$$

However, the Van Slyke criticalities of Arrows 3 and 4 are not independent because the duration distribution of the network is used in both calculations. Therefore, additional information is needed to calculate the Van Slyke criticality intersection of Arrows 3 and 4.

The Van Slyke criticality intersection of Arrows 3 and 4 occurs when the durations of their paths are tied with each other and is at least as long as the third path. At the sink node, there is 1 merger: Arrows 4 and 5 end at the sink node. Arrow 1 precedes Arrows 3 and 4. Arrow 3 precedes Arrow 5. Take the distributions of durations at the end of Arrows 4 and 5 just before they are parallel reduced to calculate the network duration distribution. The probability of the accumulated distributions of Arrows 4 and 5 tying as they enter the sink node, Node 3 in Figure 8, is the summation of the probability products that each will take on the same of all durations:

$$\Pr(A4z = A5z) = \sum_{i=0}^{\infty} f_{A4z_i} f_{A5z_i} \, . \tag{13}$$

The distribution of A5z, however, represents the reduced distributions of Arrows 2, 3, and 5. Arrow 2's contribution to the network duration should not be considered when calculating the Van Slyke criticality of Arrow 1. The path criticalities based upon any duration of Arrow 3 should be reduced by the probability that Arrow 2 exceeds the duration. Arrow 5's accumulated duration distribution is the series reduction of Arrow 5's duration distribution with the parallel-reduced duration distribution from Arrow 3's accumulated distribution and Arrow 2's duration distribution.

$$f_{A5z_i} = \sum_{j \in A5} f_{A5_j} \left(F_{A3z_{i-j}} F_{A2_{i-j}} - F_{A3z_{i-j-1}} F_{A2_{i-j-1}} \right) . \tag{14}$$

To find the probability that the 2 paths starting with Arrow 1 are both critical, 2 events must occur. The durations of the accumulated durations of Arrow 5 and Arrow 4 must be equal, and Arrow 2 must not be longer than the Accumulated duration of Arrow 3. Combining all of these factors into an equation,

$$\Pr(\mathrm{Crit}_{A3} \cap \mathrm{Crit}_{A4}) = \sum_{i \in A1} f_{A1_i} \left\{ \sum_{j \in A4z} \left[f_{A4z_{i+j}} \left(\sum_{k \in A5} f_{A5_k} f_{A3z_{i+j-k}} F_{A2_{i+j-k}} \right) \right] \right\} . \tag{15}$$

For the distributions given in Table 6, the following calculations will be performed with Arrow 1 of Figure 8 set at its conditioning value of 1 duration. Substitutions into the equation derived above for simultaneous Van Slyke criticalities of conditioned arrows, $\Pr(\mathrm{Crit}_{A3} \cap \mathrm{Crit}_{A4}) = .1[(.8)(.6)(1.0)+(.2)(.0)(.0)]+.9[(.8)(.4)(1.0)+(.2)(.6)(.3)] = .3348$. When Arrow 1 is set at duration 1, the Van Slyke criticality for Arrow 3 is .4504, and the Van Slyke criticality for Arrow 4 is 0.7668. The criticality of Arrow 1 when it is set to 1 is $\mathrm{Crit}_{A1_1} = .4504 + .7668 - .3348 = .8824$. The probability of Arrow 1 taking on the duration 1, in the example of Table 6, is 0.5. To find the overall Van Slyke criticality of Arrow 1 for the entire network, the 0.8824 is multiplied by the weight of 0.5 and added to the weighted Van Slyke criticalities of the other durations of Arrow 1. Each conditioning

Table 6: Wheatstone Example Probabilities to Calculate Van Slyke Criticalities.

	Duration Days							
	0	1	2	3	4	5	6	Criticality
Arrow 1 p.d.f.	0.1	0.5	0.4					0.86616
Arrow 2 p.d.f.			0.3	0.7				0.52224
Arrow 3 p.d.f.		0.6	0.4					0.46440
Arrow 4 p.d.f.			0.1	0.9				0.75540
Arrow 5 p.d.f.		0.8	0.2					0.76240
Project p.d.f.				0.0312	0.4920	0.4448	0.0320	

duration of Arrow 1 follows the same calculations. The numeric results for all Arrow 1 durations and their Van Slyke criticalities are given in Table 7.

Conceptually, conditioning can reduce any network into either a parallel-series network or into a network where the conditioning encompasses every combination of durations from every arrow. A method to condition a complex network to form a parallel-series merging tree is to condition upon all of the arrows that precede splits in the network. This could be done from either a forward pass or a backward paths. Depending upon the network, it is possible that almost all arrows will have to be conditioned. If that happens, complete enumeration of all combinations is almost duplicated but in a more confusing manner. The number of combinations in the

Table 7: Wheatstone Van Slyke Criticality Calculations for Arrow 1.

Arrow 1		Criticalities				Duration
Duration	Probability	Arrow 3	Arrow 4	3&4	3+4-(3&4)	Criticality
0	0.1	0.1200	0.2160	0.0864	0.2496	0.02496
1	0.5	0.4504	0.7668	0.3348	0.8824	0.44120
2	0.4	0.5680	0.8760	0.4440	1.0000	0.40000
		Total Arrow 1 Van Slyke Criticality:				0.86616

conditions can also get prohibitively large. The next chapter introduces an algorithm to minimize the prohibitive growth of duration combinations when there are many conditioning arrows.

A NETWORK ANALYSIS ALGORITHM

The algorithm presented in this chapter has the objectives of generating the duration distribution of a network and calculating the Van Slyke (1963) criticalities of the arrows in the network. Secondary goals are to perform the analysis efficiently and to identify features of the network that cause longer network durations.

The technical details on how to calculate the duration distributions and the Van Slyke criticalities in networks were discussed in previous chapters. Theoretically, those techniques are applicable to any network. Practically, however, calculations for large networks may become prohibitive. Computations may take a prohibitively long time to complete because of the large number of arrows in actual projects, of combinations of durations to condition the network, and of the number of summations required for series distribution reduction operations and for Van Slyke criticality calculations, even with modern computers. Hagstrom (1988) claims "that computing even a single point of the cumulative distribution function ... is NP-Hard" (p. 139). She was referring to the project duration distribution.

The algorithm must manage to limit the number of calculations to be efficient. The algorithm must also control the number of calculations without giving up much accuracy in order to be useful. To accomplish these seemingly contradictory goals, the algorithm will identify the important parts of the network before doing very many calculations.

Dr. Juran (1951) observed that in many, many cases, "a small percentage of the ... characteristics always contributes a high percentage" (p. 39) of effects. That although there may be many characteristics of a system, their influences are "*never uniformly distributed* over the ... characteristics. Rather, the losses are *always maldistributed*" (p. 39). Identifying the significant few network characteristics from amongst the trivial many is called a "Pareto Analysis" (Aubrey & Gryna, 1991, p. 12). The trivial many characteristics, while contributing something, will probably not contribute much. By focusing on the significant few for calculations the efficiency of the algorithm would be enhanced without losing much accuracy.

A basic characteristic of scheduling networks is that maximum path durations leading to event nodes determine the duration from the project beginning to the event's completion. This is because of the precedent relationships. As Miller (1963) puts it "no activity may start until its predecessor event is completed; in turn, no event may be considered complete until all activities leading into it have been completed. This is the key topological ground rule" (p. 34). In an analysis of the network starting with the source node, "all activities are assumed to start as soon as possible, that is, as soon as all of their predecessor activities are completed" (Moder, Phillips, & Davis, 1983, p. 74). Because event completions must wait for the longest duration of paths that lead up to the event, paths with shorter durations have no effect on when the event is completed.

By identifying the paths with the longest duration, the significant few characteristics of the network are also found. If Juran's teachings (1951) hold true for the scheduling network case, these significant few paths should be able to explain most of the network duration distributions and the Van Slyke criticalities. By discarding the paths that do not very often influence completion durations, the trivial many are ignored, thus reducing the number of computations necessary for analysis. An algorithm developed on this basis could generate probabilities without complete enumeration and without extensive simulations.

Analysis of the network may be simplified by analyzing just the significant few paths and their interactions. The interactions amongst the significant few paths could be modeled by making a skeletal network from the arrows within the few paths. With few paths chosen, the skeletal network duration distributions and Van Slyke criticalities can be calculated exactly by the methods described in previous chapters. The number of calculations necessary to perform an exact analysis would be drastically reduced. In effect, the algorithm would pick from the network a sample biased by the duration of paths, build the paths into a skeletal network, and then completely analyze the smaller network.

Before developing the details of this algorithm, the algorithm assumptions will be stated. In the project duration distribution section, a procedure is presented to select the significant few paths of a network to best represent the completion duration distribution. Analysis of the selected paths include forming a new network and calculation duration probabilities. The Criticality section will discuss how to recognize when the different exact calculation methods for Van Slyke criticalities, presented in the previous chapter, are applicable. Finally, the last section of this chapter will summarize the algorithm. Emphasis in the summary will be on the steps and rules needed to apply the algorithm to any network. In the next chapter, the algorithm will be tested on a moderately large, complex, published network.

Algorithm Assumptions

The algorithm assumes a scheduling network that models activities as arrows and events as nodes. The network is assumed to have one source node and one sink node. All arrows in the network have a starting node, an ending node, and a finite, discrete distribution of possible durations, and an assigned index number. The duration distributions for all arrows are assumed to be independent of all other arrow duration distributions. All nodes in the network are also assumed to have index numbers, and they are assumed to have recorded the number of arrows that start and end at the node. The precedence relationships within the network are defined by arrows indicating the node index numbers of its starting and ending nodes. Before an arrow can start its duration, its starting node must be realized. A node is realized if all of the arrows that end at that node have completed their durations.

The precedence relationships within the network are assumed to have no cycles. That is, once a path reaches a node, it cannot reach the node again. In addition, there are no decision points in the network. All arrows must be performed once and only once, and are assumed to start as soon as they are eligible to do so.

It is further assumed that, for analysis purposes, it makes no difference whether the network is analyzed forward in time from the source node or backwards in time from the sink node. Time is a strictly nonnegative entity. The analysis will focus on the distributions and precedents of the arrows in the network not what has to physically happen in the project to make the precedents necessary. Since the algorithm will generate probabilities associated with lengths of time (duration distributions and the probability of the longest durations), this assumption will result in no loss in generality. Whether one adds up a list of numbers in ascending order or in descending order makes no difference to the final sum.

Estimating The Project Duration Distribution

CPM and PERT find critical paths in a network from fixed arrow durations. CPM inputs call for fixed durations. PERT accepts duration distributions for arrows as input, but calculates the critical paths using the means from the arrow duration distributions, and only uses the variances to break ties. Both CPM and PERT examine the influence of the critical path on other parts of the network once the critical path is found.

The durations of paths are directly comparable to project durations. Critical paths are easily seen to be critical if their durations match the duration of the project. The path duration is the sum of its arrow durations. An arrow is critical only if it is on a critical path. Arrows are often on more than one path.

Paths have distributions of durations when the arrow durations it sums come from distributions. Because the paths have distributions for their durations, their durations are uncertain. When all paths in a network have duration distributions, the duration of the network is uncertain. It has long been recognized that the single critical path through PERT networks underestimate the network's duration (For example, Malcolm, Rosenbloom, Clark, & Fazar, 1959, and MacCrimmon & Ryavec, 1964). Paths other than the one designated as critical by PERT increase the probability of achieving longer durations. When PERT's critical path assumes a shorter duration from its distribution, the project duration might not be shorter because a competing path may have assumed a larger duration from its distribution. The path with the longest duration determines the project duration. Because of the path duration distributions and the uncertainty as to which paths will be critical for any given actual project, the network duration has a distribution.

Two things are clear about the paths in a network with a distribution of durations. First, the longest paths dominate the distribution. The author likes to compare this to basketball players. The tall people tend to get the rebounds. Tall people don't always get the rebounds, but they get them more often. It is the same way with network durations. Paths with durations longer than many of the competing paths will determine the project duration more often. The second thing is that as well as dominating paths there are paths that are dominated. Dominated paths have maximums that are shorter than some other path's minimum. The algorithm finds the dominating paths.

<u>Path Selection</u> The network minimum and maximum durations are quantified with the first 2 paths selected. A path determining the minimum network duration is identified by taking the path with the longest duration when all network arrows are assigned their minimum durations. A path determining the maximum network duration is identified by taking the path with the longest duration when all network arrows are assigned their maximum durations. The Path Identification method initially defines all paths as discussed in the previous chapter. Path durations are the sum of durations

from the path's arrows. This is true whether the durations are assigned or whether they are sampled from the arrow duration distributions. The longest paths are found by sorting, in descending order, the path durations generated by summing arrow durations. The 2 paths dominate portions of other paths but in different manners.

The longest path with minimum arrow durations determines the lower bound of the network distribution. Other paths are dominated when their durations are less than the network's minimum duration. If the maximum durations of other paths are greater than the network's minimum duration, the other paths are totally dominated, and can be disregarded altogether. Van Slyke (1963) recommended this technique for eliminating paths that could never be critical. The network's minimum duration is this path's minimum duration. The path's duration can only increase becoming more dominant as it does so. This is an excellent example of the significant few paths have large impacts -- the path not only dominates other paths at its minimum duration, but it also dominates other paths throughout its entire duration distribution.

The longest path with maximum arrow durations determines the upper limit of the network distribution. This path is always critical when it is at its maximum duration. No other paths are capable of being longer. Unless there is a tie for the longest possible path, the path has a range of durations which will guarantee that it will be critical. The range is the difference between the path's maximum duration and the next longest duration from another path. The probability of being a critical path is diminished when its duration drops below the secondary path's maximum duration. The probability that the network will reach any duration in this range is exactly the probability that the path will achieve a duration in the range from its distribution. In other words, the right tail of the path's duration distribution is the right tail of the network's duration distribution past the secondary path's maximum duration. The path defines the upper limit of the network's duration distribution.

It is possible for there to be ties for the longest path. Paths are picked from ties by maximizing the probabilities of having large durations. Ties for the longest path with maximum duration arrows are broken by selecting the path with the longest duration when the arrows are set at their minimums. Ties for the longest path with minimum duration arrows are broken by selecting the path with the longest duration when the arrows are set at their maximums. If there are still ties, pick the path with the lowest number. (The number is the order in which the Path Identification procedure generates paths.)

In practice, the significant few paths involve the right tail of the network duration distribution. Project schedulers set project completion time targets to include most of the project duration distribution. The schedule target dates "are usually in the high 90s percentile" (Elmaghraby, 1977, p. 30) of the project duration distribution. The network scheduled times of completion "can be used as vehicles for producing useful information about project ... planned costs" (Moder, Phillips, & Davis, 1983, p. 135). The costs are important in making budgets for the project. "The budgeting process enables the firm to set forth specifically how it intends to realize its objectives and to coordinate the various activities that it will be required to carry out" (Granof, 1983, p. 563). Indeed, incentives to meet schedule are built into some contracts (Miller, 1963).

To select the paths that dominate the right tail of the network duration distribution, the paths are sorted in descending order after all arrows have been assigned their maximum duration. The first path from the sort has already been picked; it is the one which determines the far right-tail of the network duration distribution. The second path competes for criticality when the first path's duration drops below its guaranteed criticality range. Between the first and second paths, the probabilities of the durations above the third path's maximum duration are exactly calculable by the methods defined in the Calculation of Network Distributions chapter. Adding additional paths increases the range of network distribution durations which have exact calculated probabilities. The paths with the longest maximum durations influence more than the network duration distribution's right tail. Lesser durations from these same paths still dominate; so long as they are greater the network's minimum duration.

Paths with diminishing maximum durations could be added until a path maximum duration equals the network minimum duration or until calculation times became unbearable. Theoretically, these paths, together with the exact network distribution methods already discussed, would yield an exact distribution for the network durations. Practically, computing times for networks with many paths will become unbearable. In addition, each new path's contribution to the network distribution becomes less and less. The last ones added truly are the trivial many.

The Skeletal Network The number of paths to include in a skeletal network can be somewhat flexible. Paths could be added to a skeletal network until either computing resources or human patience reaches a limit, or a maximum number of paths could be set before hand. Anklesaria & Drezner (1986) "predicted very accurately" (p. 813) the project duration distributions by using 5 paths in their PERT analysis using multivariable normal distributions. The algorithm will select 6 paths. The 6 paths include the path with the longest duration when all arrows are set to their minimum durations, and the 5 paths with the longest durations when the arrows are set to their maximums.

A skeletal network is built from the arrows of the 6 paths. Arrows from the 6 paths are given additional new numbers, but nodes from the original network are not. Arrow numbers are cross referenced. The nodes are reproduced for the new network. The new network will have one source node, one sink node, no cycling path of arrows. The new skeletal network may have more than the original 6 paths. When paths intersect, they create places where paths are partitioned, and the partitions can form combinations of paths not originally included. This is another example of how the significant few paths influence the trivial many. The included paths are included in subsequent analysis.

The skeletal network is designed to minimize the number of calculations needed to get accurate network probabilities. By identifying and reducing arrows in series, the number of necessary calculations are decreased further. Arrows in series are identified by nodes which end exactly 1 arrow and starts exactly 1 arrow. The arrows on either side of the node are reduced into an equivalently distributed arrow by the series reduction operation, and the node is eliminated from the skeletal network. The new arrow has a different number, but references to the original arrows are maintained. (Arrows in parallel arrangements are needed for Van Slyke criticality calculations.)

Network Duration Distribution Calculations The algorithm calculates network duration probabilities by the Law of Total Probability (Taylor & Karlin, 1984). Pritsker (1995) describes the law as "the probability of the outcome B is equal to the sum of the conditional probabilities associated with B given the occurrence of mutually exclusive and exhaustive outcomes A_i weighted by the probability of A_i, that is, $P(B) = \sum_i P(B|A_i)P(A_i)$" (p. 26). The combinations of arrow durations are mutually exclusive and exhaustive. The network duration distribution is calculated by the algorithm conditioning on the durations of a set of arrows. The probability of any duration in the network duration distribution is calculated by the Law of Total

Probability. For a given duration, the total network probability is the sum of the products of the conditioned network probabilities times the probability of the combination of arrow durations used for conditioning.

Conditioning the skeletal network is the first step in calculating network distribution probabilities. Conditioning reduces the skeletal network complexity to the complexity of a parallel-series network. Thus, conditioning is not required for parallel-series skeletal networks. Exact probability calculations are available for networks with arrows in parallel-series arrangements.

A parallel-series network exists when the arrow duration distributions are reducible to the network's duration distribution by using only the parallel reduction and the series reduction operations. A parallel reduction operation is indicated if 2 or more arrows have the same starting and the same ending nodes. A series reduction operation is indicated if a node starts 1 arrow and ends 1 arrow. The parallel-series check does not have to actually perform the duration distribution reduction operations. Only the reduction in arrow numbers and the elimination of nodes are tracked in the check for a parallel-series network. The algorithm assumes that the skeletal network of arrows is not initially in a parallel-series arrangement.

The algorithm conditions the skeletal network by conditioning arrows which come before splitting nodes or after merging nodes. A splitting node starts 2 or more arrows, and a merging node ends 2 or more arrows. The conditioning arrows can either be designated starting at the source node and working forwards or can be designated starting with the sink node and working backwards through the network. Depending upon the skeletal network, one set of designated conditioning arrows will be more efficient than the other. Network probability calculations must be repeated for every combination of durations from the conditioning arrows. The number of combinations for each set of conditioning arrows are calculated by taking the product of the number of each conditioning arrow's duration. The minimum number of combinations determine which set of conditioning arrows are used. If the duration combinations from the set of conditioning arrows coming before splits are less than or equal to the duration combinations coming after mergers, the set of conditioning arrows coming from the source node are selected.

The arrows in the selected set of conditioning arrows are designated Conditioning Arrows. All other arrows are designated Conditioned Arrows. In addition, Conditioned

Arrows which share nodes with Conditioning Arrows are also designated as Starting Arrows. The nodes common to Starting Arrows and Conditioning Arrows will be before the Starting Arrows if the Conditioning Arrows start from the source node and will come after the Starting Arrows if the Conditioning Arrows end at the sink node. The Conditional Probability is defined as the probability of a combination of Conditioning Arrow durations. The conditional probability is calculated by multiplying the probabilities of the Conditioning Arrow durations in the combination.

The durations from the Conditioning Arrows are translated into the Starting Arrow duration distributions. A combination of Conditioning Arrow durations fixes the conditional durations of the splitting nodes (or merging nodes if the Conditioning Arrows start at the sink node). The conditional split (or merge) node durations are translated into the Starting Arrows' duration distributions. The probabilities in the Starting Arrow duration distribution do not change, only what duration the probabilities represent. The durations of the Starting Arrow distributions are increased by the conditional splitting node duration. Instead of trying to enter the Conditioning Arrow durations into conditional calculations, the Conditioning Arrow durations are translated into Starting Arrows' duration distributions before conditional calculations are performed. The translated distributions of the Starting Arrows are still independent. If constants are added to 2 independent variables, they are still independent afterwards (Freund, 1992).

The conditional network duration distributions are calculated with parallel distribution reduction operations and series distribution reduction operations once the durations of the Conditioning Arrows are translated into the Starting Arrow duration distributions. The conditional network duration probabilities are multiplied by the Conditional Probability and accumulated into a total probability for each duration. The probability of each network duration is the accumulated total probability after all combinations of Conditioning Arrow durations have been used for conditioning.

Combinations of Conditioning Arrow durations are sequentially identified. The Conditioning Arrow numbers are listed in ascending order. All Conditioning Arrows are set to their minimum durations, the first conditional probability is calculated, and the network conditional calculations are performed. The Conditioning Arrow durations are incremented in the order they appear on the list. The first Conditioning Arrow always gets incremented if its duration is less than its maximum. If the first Conditioning Arrows on the list are already set to their maximum durations, the first Conditioning

Arrow not at its maximum is incremented, and the Conditioning Arrows higher on the list are reset to their minimum durations. All combinations have identified when all arrows on the list are set at the maximum durations. The conditional probability and network conditional calculations are performed for all combinations.

Van Slyke Arrow Criticality Calculations

The algorithm calculates Van Slyke criticalities for the arrows in the skeletal network. Arrows which are not part of the skeletal network are assumed to be unimportant. "Critical path methods facilitate ... focusing management attention on the 10 to 20 per cent of the project activities that are most constraining on the schedule" (Moder, Phillips, & Davis, 1983, p. 19). The longest paths selected for the skeletal network will necessarily identify the most critical arrows. Arrows which are not part of the 6 paths selected for the skeletal network must belong to the trivial many. The excluded arrows might occasionally be on a critical path, but only if the project duration is below the 95th percentile where schedules are set (Elmaghraby, 1977).

The key to calculating Van Slyke criticalities is the skeletal network topology. The topology is divided by designating Conditioned Arrows and Conditioning Arrows as described in the previous section. Conditioning the skeletal network allows the Dependent Parallel Criticality Equation to calculate Van Slyke criticalities directly for the Conditioned Arrows and indirectly for the Conditioning Arrows. Identifying the Parallel Merging Tree topology formed by the Conditioned Arrows is crucial to correctly calculating the Van Slyke criticalities.

Assume, without loss of generality, that the set of Conditioned Arrows come after the Splitting Nodes (nodes which start more than 1 arrow), and that all of the arrows which precede the Splitting Nodes form the set of Conditioning Arrows. The Merging Parallel Tree consists of the Conditioned Arrows. The Starting Arrows are the branches of the Merging Parallel tree because they start with Splitting Nodes. The sink node is the base of the Merging Parallel tree. All nodes between the Starting Arrows and the sink node are merging nodes. The initial series distribution reduction operations performed on the skeletal network would have precluded any nodes with 1 path, and all of the Splitting Nodes occur before the Starting Arrows.

Van Slyke Criticality Calculations for Conditioned Arrows The Dependent Parallel Criticality Equation calculates the Van Slyke criticalities of the Conditioned Arrows after they are conditioned by the Conditioning Arrows. The algorithm calculates Van Slyke criticalities for Conditioned Arrows via an alternate form of the Dependent Parallel Criticality Equation. Referring to Figure 7, the Dependent Parallel Criticality Equation (Equation 10) is transformed into the following:

$$\boxed{Crit_{A2} = \sum_{j \in A2} \left[f_{A2j} F_{A3_j} \left(\sum_{i \in A1} f_{A1_i} F_{A4_{i+j}} \right) \right].} \tag{16}$$

The $F_{A4_{i+j}}$ is the probability that Arrow 2's path of Figure 7 beats or ties the parallel path of Arrow 4. The new form of the Dependent Parallel Criticality Equation must have a cumulative distribution for every path which merges with nodes that lay between Arrow 2 and the sink node. While potentially increasing calculations, the sparsity of the skeletal network guarantees manageability.

Most distributions input into the revised Dependent Parallel Criticality Equation (Equation 16) are accumulated arrow duration distributions. An accumulated arrow duration distribution is the duration distribution to the end of an arrow, including all skeletal paths which include the arrow. In the algorithm, the accumulated arrow duration distributions will be calculated after the durations of the Conditioning Arrows have been fixed. The accumulated arrow duration distribution of a Starting Arrow is the arrow's duration distribution translated by the conditional duration of its starting node, a Splitting Node. Accumulated duration distributions of arrows which come after the Starting Arrows are calculated by parallel reducing all accumulated arrow durations which enter the present arrow's starting node and series reducing the resultant duration distribution with the arrow's duration distribution. These operations are already performed to calculate the skeletal network duration distribution.

Configuring the Dependent Parallel Criticality Equation (Equation 16) for a Conditioned Arrow is dependent upon how many mergers and nodes there are from the arrow to the last merging node. A nested summation is required for every node, including the arrow's ending node, between the Conditioned Arrow and the last merging node. For the summation indicated by each subsequent node, the number of factors equals the number of arrows starting or ending at the node (except arrows starting at the last merging node are ignored). Each summation is over the duration range of the arrow whose Van Slyke criticality is being calculated or is on the path of the arrow whose Van Slyke criticality is being calculated. The first factor in each nodal summation is a duration distribution probability of the calculation arrow or an arrow on its path. If the arrow whose Van Slyke criticality is being calculated merges at the node, the probabilities are from its accumulated arrow duration distribution (p.d.f.); otherwise, the probabilities are from subsequent path arrow's duration distribution (p.d.f.). Other factors at each nodal summation are the cumulative probabilities (c.d.f.) from the accumulated duration distributions from the arrows merging at the node. Finally, the last factor is a summation from the next node, if any.

Van Slyke Criticality Calculations for Conditioning Arrows   Fixing the durations of the Conditioning Arrows prevents direct calculation of the Van Slyke criticalities via the Dependent Parallel Criticality Equation. The probabilities of one Conditioning Arrow being longer than another is either 0.000 or 1.000 for any combination of conditional values. The conditional probability is the only probability associated with a combination of Conditioning Arrow durations. The Conditioned Arrow duration distributions in the Parallel Merging Tree provided the Dependent Parallel Criticality Equation the probabilities for calculations. The same duration distributions are also used to calculate the Van Slyke criticalities for the Conditioning Arrows.

The first step in calculating the Van Slyke criticalities for Conditioning Arrows is to examine what happens at the Border Nodes. Border Nodes are the Splitting Nodes which simultaneously end Conditioning Arrows and start the Starting Arrows (Starting Arrows are also Conditioned Arrows). The conditional durations of the Conditioning Arrows are initially assigned to the Border Nodes before being translated into the Starting Arrow duration distributions. A Border Node conditional duration is the largest path duration from the paths leading from the source node to the Border Node. The Van Slyke criticalities of the Starting Arrows of a Border Node contribute to the Van Slyke criticalities of all Conditioning arrows which are on the critical paths leading up to the Border Node.

The algorithm identifies the Border Nodes and the paths leading up to it. For each path, a number is assigned, the ending Border Node is recorded, and the defining sequence of Conditioning Arrows is recorded. Then, for each combination of

Conditioning Arrow durations, the algorithm sums the conditional duration of all paths leading to Border Nodes, designates and records the critical paths, and records the numbers of the critical paths that are associated with each Conditioning Arrow.

Conditioning Arrow Van Slyke criticalities are a function of what Starting Arrows follow the conditioning arrows' critical paths leading to the Border Nodes. The Van Slyke criticality for Conditioning Arrows is 0.000 for the skeletal network when the Conditioning Arrow is not on a critical path. The Van Slyke criticality of a Conditioning Arrow that is on 1 critical path ending in a Border Node with has 1 Starting Node is the Van Slyke criticality of the Starting Node. If, on the other hand, the Conditioning Arrow leads to multiple Starting Arrows, the Van Slyke criticality calculation is a joint probability problem.

The Van Slyke criticality of any Conditioning Arrow is contained in the Van Slyke criticalities of the Starting Arrows which follow any critical path to Border Nodes that includes the Conditioning Arrow. The Van Slyke criticalities for a set of Starting Arrows, and thus a Conditioning Arrow, is the union of dependent probabilities. Concerning the probability of at least one of the events in the union are true, Ross (1980) states the following:

> The probability of the union of *n* events equals the sum of the probabilities of these events taken one at a time minus the sum of the probabilities of these events taken two at a time plus the sum of the probabilities of these events taken three at a time, and so on (p. 6).

The algorithm, therefore, calculates the probabilities that a set of Starting Arrows indicated by a Conditioning Arrow are simultaneously Van Slyke critical. To have simultaneous Van Slyke criticalities, there must be more than one critical path. Each critical path must have the same duration and be of greater duration than other paths or they wouldn't be critical paths. The probabilities that any two paths have identical durations is given by Equation (13). The probability that a path is critical over other path is given by Equation (7). The two equations can be combined to calculate the simultaneous Van Slyke criticalities of any set of Starting Arrows.

For any set of Starting Arrows, label the Starting Arrow set and all arrows following the Starting Arrow set to the last merging node as <u>Champion Arrows</u>. All other arrows are designated <u>Competing Arrows</u>. The labels come from the paths the arrows form: the

Champion Arrows form the paths of interest, and the other arrows form paths that compete for criticality.

Starting at the last merging node and proceeding backwards until the Starting Arrows are reached, the probability that Starting Arrows are simultaneously on critical paths is calculated by climbing the Parallel Merging Tree. To build a formula equivalent to Equation (15) for each set of Starting Arrow Van Slyke criticality ties, the rules are as follows. The last merging node sums over its duration range as many factors as there are merging arrows. The duration range of the last merging node is relevant to all Starting Arrows since the paths that the Starting Arrows begin must end with the last merging node. The merging arrows which are labeled Champion Arrows have a summation for a factor, and the arrows which are labeled Competing Arrow have a cumulation probability from an accumulated arrow duration distribution for its factor. The summations for Champion Arrows are over the range of the merging Champion Arrow, and includes as factors the p.d.f. of the Champion Arrow, and the probabilities from arrows which merge at the Champion Arrow's starting node, provided the node is not a Border Node. Again, the merging probabilities are in the form of p.d.f. distributions for Champion Arrow durations and in the form of c.d.f. distributions for the Competing Arrows. The subscripts for the merging arrows is the difference of the last merging node's duration and all other summation duration indexes.

Once the probabilities of Van Slyke criticality ties for all combinations of a set of Starting Arrows have been figured, the Van Slyke criticality of the Conditioning Arrow is calculated. The Van Slyke criticalities of the Starting Arrows are summed and adjusted by the probabilities of ties according to the description of Ross (1980). The procedure is repeated for all combinations Conditioning Arrow durations. The Van Slyke criticality for the Conditioning Arrows are then accumulated over each conditioning combination.

Algorithm Summary

A summary of the algorithm is presented in Table 8. The algorithm follows an encouragement by Bowman: "estimate project completion time simultaneously with ... criticalities" (Bowman, 1995, p. 66)." Algorithm calculations for duration distributions are also needed to calculate arrow Van Slyke criticalities. Criticality and duration probabilities are related in that critical paths determine durations.

The algorithm calculates a network duration distribution that is mathematically correct in its duration range and upper-tail duration probabilities via a skeletal network. The skeletal network is constructed from the arrows in the paths with the longest durations. The network range is determined by choosing the paths with the longest minimum and maximum durations. The right tail probabilities are determined by the paths with the longest maximum durations. Once constructed, the skeletal network is analyzed with exact calculations. The range of exact right-tail probabilities is from the maximum network duration to the maximum duration of the longest path not included in the skeletal network.

The algorithm efficiently estimates Van Slyke criticalities and duration probabilities not in the right duration distribution tail. Efficiency of the algorithm comes from picking the significant few arrows in the network and ignoring the trivial many. The Van Slyke arrow criticality estimates are biased toward the longer network durations because the algorithm picks the paths with the longest durations. This bias is actually beneficial to management who are most concerned with project overruns. The duration probability estimates between the upper-tail and the minimum network durations are dominated by the longest paths, but other paths contribute to duration probabilities in this region. The estimates could be improved if more paths were included in the skeletal network, but efficiency will suffer.

Table 8: Steps of the Algorithm.

Step	Step Description
1	Identify all paths in the network and record their arrows.
2	Set all arrows within the network to their minimum durations.
3	Calculate and record the minimum durations of all paths.
4	Designate the path with the longest minimum duration as skeletal path number 1.
5	Set all arrows within the network to their maximum durations.
6	Calculate and record the maximum durations of all paths.
7	Rank the paths in descending order by their maximum durations.
8	Designate skeletal paths 2-6 by picking 5 additional paths with the longest maximums.
9	Identify and record the arrows in the 6 skeletal paths.
10	Construct and record a skeletal network from the arrows in the 6 skeletal paths.
11	Condense the skeletal network by performing series reductions where possible.
12	Designate skeletal nodes which start more than 1 arrow as Splitting Nodes.
13	Identify all reduced arrows preceding the splitting nodes.
14	Calculate the combinations of condensed skeletal arrow durations which precede splits.
15	Designate skeletal nodes which end more than 1 arrow as merging nodes.
16	Identify all reduced arrows following the merging nodes.
17	Calculate the combinations of condensed skeletal arrow durations which follow mergers.
18	If fewer combinations, designate arrows following mergers as conditioning arrows, otherwise designate the arrows preceding splits.
19	Designate the other arrows as Conditioned Arrows.
20	Designate conditioned arrows sharing a node with conditioning arrows as Starting Arrows.
21	Assign to all conditioning arrows, their minimum durations.
22	Calculate and record the product of the assigned conditioning arrow duration probabilities.
23	Label the product of the conditioning arrow duration probabilities "Conditional Probability."
24	Calculate conditioning node durations from the conditioning arrow assigned durations.
25	Translate the distributions of the Starting Arrows by the conditioning node duration.
26	Calculate conditional network's distributions by parallel-series reduction operations.
27	Multiply the conditional network duration distribution by the conditional probability.
28	Accumulate the conditional network duration probabilities into a total distribution.
29	Calculate the conditional Van Slyke criticalities for the Starting and Conditioned Arrows..
30	Calculate the conditional Van Slyke criticalities for the Conditioning Arrows.
31	Multiply the conditional Van Slyke arrow criticalities by the Conditional Probability.
32	Accumulate the conditionally calculated Van Slyke criticalities into totals for all arrows.
33	If available, assign a new combination of durations to the Conditioning Arrows & go to Step 20.
34	If new combinations are not available, calculations are complete.
35	Record the total project duration distribution from the accumulated total probabilities.
36	Record the total arrow Van Slyke criticalities from the accumulated totals.

EVALUATION OF THE ALGORITHM

Performance of the algorithm is demonstrated on the 40-arrow, 22-node network published by Kleindorfer (1971). The network is depicted in Figure 5 and will be referred to in the following text simply as the Kleindorfer network although Kleindorfer published other ones. Various researchers have studied Kleindorfer networks, among them have been Robillard & Trahan (1977), Shogan (1977), Dodin (1985b & 1985c), and Dodin & Sirvanci (1990). The largest Kleindorfer network has, in fact, become the benchmark for complex scheduling networks. Although industry has had much larger networks, few, if any, exceed the complexity of dependency relationships within the Kleindorfer network. Another reason the Kleindorfer network is a benchmark is because of the proprietary data associated with many industry networks are not published.

This chapter will test the algorithm described in the previous chapter on the Kleindorfer network. First, characteristics of the Kleindorfer network are reviewed. Second, an evaluation of the Kleindorfer network duration distribution by the algorithm is performed and compared with a SLAM II simulation. Third, the Van Slyke criticalities are calculated by the algorithm for the Kleindorfer arrows and compared with simulation results. Finally, there is a brief discussion about the Kleindorfer network's duration distribution.

Kleindorfer Network

The Kleindorfer (1971) 40-arrow, 22-node network, shown in Figure 5, is "the largest stochastic network appearing in the open literature" (Dodin & Sirvanci, 1990, p. 402). It is "complex with respect to both its geometry and the probability distributions of the lengths of its arcs [arrows]" (Shogan, 1977, p. 377). The Kleindorfer arrow duration distributions are discrete and are assumed to be independent. The arrow duration distributions are one of two types: uniform or triangular. The uniform distributions vary in their range, and the triangular distributions vary in both their range and the locations of the peak the range. Kleindorfer calculated arrow duration probabilities by separately calculating the numerator and denominator of fractions. The Kleindorfer arrow duration distributions given in Table 9 present the fractional numerator under the arrow duration columns, and the arrow's fractional denominator is given in the second column, abbreviated "Denom." The probability for any arrow's duration is found by dividing the

Table 9: Arrow Distributions for the Kleindorfer (1971) Network.

Arrow	Denom.	Probability numerators for Arrow Durations																													
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	1	1																															
2	4										1	2	1																				
3	15		3	6	4	2																											
4	5			1	1	1	1	1																									
5	2													1	1																		
6	2		1	1																													
7	15		2	4	6	3																											
8	4				1	2	1																										
9	4		1	1	1	1																											
10	15																2	4	6	3													
11	23			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1							
12	3		1	1	1																												
13	3		1	1	1																												
14	15					3	6	4	2																								
15	15					3	6	4	2																								
16	49						1	2	3	4	5	6	7	6	5	4	3	2	1														
17	35		2	4	6	8	10	5																									
18	15		3	6	4	2																											
19	4			1	2	1																											
20	4		1	2	1																												
21	4					1	1	1	1																								
22	3				1	1	1																										
23	3			1	1	1																											
24	2															1	1																
25	15		2	4	6	3																											
26	169								1	2	3	4	5	6	7	8	9	10	11	12	13	12	11	10	9	8	7	6	5	4	3	2	1
27	15		3	6	4	2																											
28	3				1	1	1																										
29	64		1	2	3	4	5	6	7	8	7	6	5	4	3	2	1																
30	4			1	1	1	1																										
31	3		1	1	1																												
32	2		1	1																													
33	9									1	2	3	2	1																			
34	3				1	1	1																										
35	2		1	1																													
36	9				1	1	1	1	1	1	1	1	1																				
37	17						1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1										
38	2														1	1																	
39	19		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1												
40	1	1																															

arrow duration's numerator by the arrow's denominator. Blanks in Table 9 are interpreted as zeros.

Kleindorfer (1971) introduced the network to illustrate a procedure for bounding network duration cumulative distributions. Subsequent to Kleindorfer publishing upper and lower bounds on his distribution, Shogan (1977) and Dodin (1985c) studied the network and published their own bounds on the cumulative duration distribution. Of the three, Shogan has the tightest bounds on the Kleindorfer network's cumulative duration distribution. Shogan's bounds are illustrated in Figure 9. There has been nothing published about the Van Slyke criticalities of the Kleindorfer network arrows.

Algorithm Distribution Analysis On Kleindorfer Network

The Kleindorfer (1971) network shown in Figure 5 meets all of the algorithm assumptions. The arrow duration distributions are all discrete and independent, the network has 1 source node and 1 sink node, and there are no loops in the precedent relationships. Therefore, no special processing is needed to adapt the network for the algorithm.

The paths through the network shown in Figure 5 were numbered and defined in Table 4 by the path identification procedure described in a previous chapter. For each of

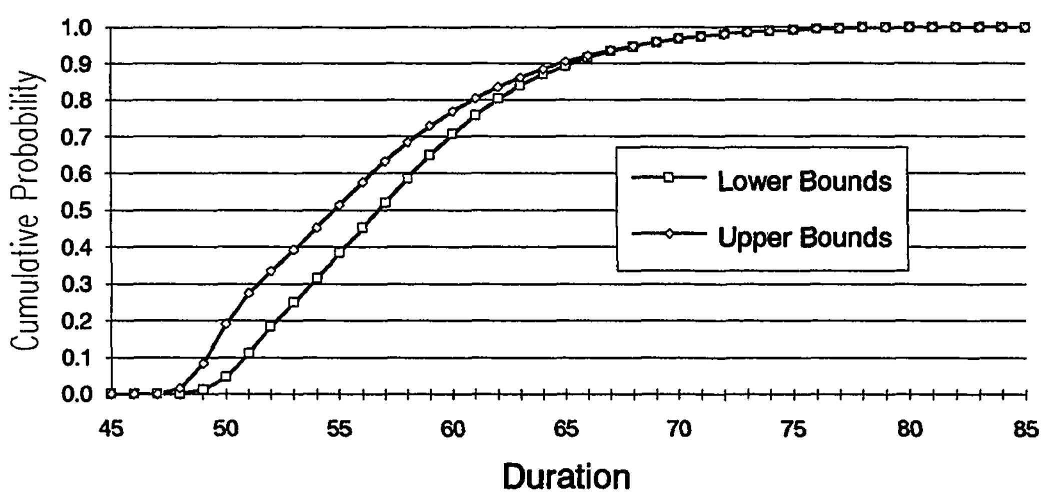


Figure 9: Shogan (1977) c.d.f. Bounds on Kleindorfer (1971) Network.

the 51 Kleindorfer network paths, the path durations are calculated by summing the minimum duration of its arrows, and then the path durations are repeated for the maximum arrow durations. The minimum and maximum durations of each path is listed in Table 10, together with the path number and its defining sequence of arrows.

The Kleindorfer paths in Table 10 have been sorted according to algorithm rules. Path 30 is ranked number 1 because its minimum duration of 48 is greater than the minimum duration of any other path. The network minimum duration must also be 48, and the criticality of any path duration below 48 is zero.

The paths ranked 2 through 6 were chosen by the algorithm because of their high maximum durations. In this case, the top ranked path, the one with the longest minimum duration, was not also amongst the longest maximum duration paths. Path 27 had the maximum path duration of 89 by summing the maximum arrow durations from Arrows 1, 3, 11, 26, 36, 39, and 40. The 89 also determines the theoretical maximum limit for the network duration distribution. It is interesting to note that the maximum duration of 89 is greater than the 82 that Shogan (1977) offered as the duration at which his bounding cumulative distribution converged to 1.000. The difference between a duration of 89 and the third ranked path's duration of 75 is 14. The probability of any duration greater than 75 depends only on Path 27 -- all other paths are dominated. Likewise, the network duration probabilities of 73, 74, and 75 depend only on Path 27 and Path 11. The 5 paths with the largest maximum durations explain the range of network durations from 68 to 89, and the 5-path interactions explains the durations' probabilities. This means that the probabilities of over $\frac{1}{2}$ of the network durations are calculated exactly!

The arrows from the 6 paths have been constructed into the skeletal network appearing in Figure 10a. By using a skeletal approach, the algorithm reduced the number of Kleindorfer network arrows from 40 to 20 and reduced the combinations of arrow durations from 2.3×10^{24} to 2.7×10^{13}. The top 6 paths accounted for 50% of the network arrows, well above the 10 - 20% that Moder, Phillips, & Davis (1983) claim are of interest to managers. By combining the arrows from 6 paths into a skeletal network, the paths may criss-cross. The 6 paths taken from the Kleindorfer network formed 8 paths when the arrows were made into the skeletal network. The 2 additional paths are Paths 38 and 39 which are ranked 8 and 9, respectively.

Table 10: Path Maximums of the 40-Arrow Kleindorfer (1971) Network.

Path				Ordered Path Arrow Numbers										
Rank	Number	Min.	Max.	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th	11th
1	30	48	53	1	2	5	24	38	40					
2	27	14	89	1	3	11	26	36	39	40				
3	11	27	75	1	3	10	16	29	37	40				
4	12	26	72	1	3	10	16	31	36	39	40			
5	37	22	70	1	2	6	9	16	29	37	40			
6	13	28	68	1	3	10	16	28	34	39	40			
7	2	27	67	1	3	10	16	28	32	35	39	40		
8	38	21	67	1	2	6	9	16	31	36	39	40		
9	39	23	63	1	2	6	9	16	28	34	39	40		
10	18	22	62	1	2	6	9	16	28	32	35	39	40	
11	45	23	61	1	2	5	23	27	34	39	40			
12	23	21	61	1	2	4	12	22	29	37	40			
13	41	21	61	1	2	4	12	22	29	37	40			
14	8	12	61	1	3	11	20	25	30	37	40			
15	14	38	60	1	3	10	16	28	32	38	40			
16	16	37	60	1	2	5	24	35	39	40				
17	3	31	58	1	2	5	19	25	30	37	40			
18	46	22	58	1	2	6	8	14	31	36	39	40		
19	24	20	58	1	2	4	12	22	31	36	39	40		
20	42	20	58	1	2	6	7	12	22	31	36	39	40	
21	34	30	57	1	2	6	8	13	24	35	39	40		
22	28	28	56	1	2	5	23	27	34	39	40			
23	10	25	56	1	3	10	18	25	30	37	40			
24	40	33	55	1	2	6	9	16	28	32	38	40		
25	15	27	55	1	2	5	23	27	32	35	39	40		
26	17	24	55	1	2	6	8	13	19	25	30	37	40	
27	6	21	55	1	2	4	12	21	27	34	39	40		
28	21	21	55	1	2	6	7	12	21	27	34	39	40	
29	9	24	54	1	3	10	17	30	37	40				
30	47	24	54	1	2	6	8	14	28	34	39	40		
31	25	22	54	1	2	4	12	22	28	34	39	40		
32	43	22	54	1	2	6	7	12	22	28	34	39	40	
33	1	20	54	1	1	2	4	12	21	27	32	35	39	40
34	4	20	54	1	2	6	7	12	21	27	32	35	39	40
35	31	23	53	1	2	6	8	14	28	32	35	39	40	
36	5	21	53	1	2	4	12	22	28	32	35	39	40	
37	20	21	53	1	2	6	7	12	22	28	32	35	39	40
38	49	21	53	1	2	6	8	13	23	27	34	39	40	
39	33	20	52	1	2	6	8	13	23	27	32	35	39	40
40	32	24	51	1	2	6	8	15	30	37	40			
41	36	20	51	1	2	6	8	13	23	27	34	39	40	
42	51	41	50	1	2	6	8	13	24	38	40			
43	35	19	49	1	2	6	9	17	30	37	40			
44	29	38	48	1	2	5	23	27	32	38	40			
45	7	31	47	1	2	4	12	21	27	32	38	40		
46	22	31	47	1	2	4	12	22	28	32	38	40		
47	48	34	46	1	2	6	8	14	28	32	38	40		
48	26	32	46	1	2	4	12	22	28	32	38	40		
49	44	32	46	1	2	6	7	12	22	28	32	38	40	
50	19	23	46	1	2	6	33	37	40					
51	50	31	45	1	2	6	8	13	23	27	32	38	40	

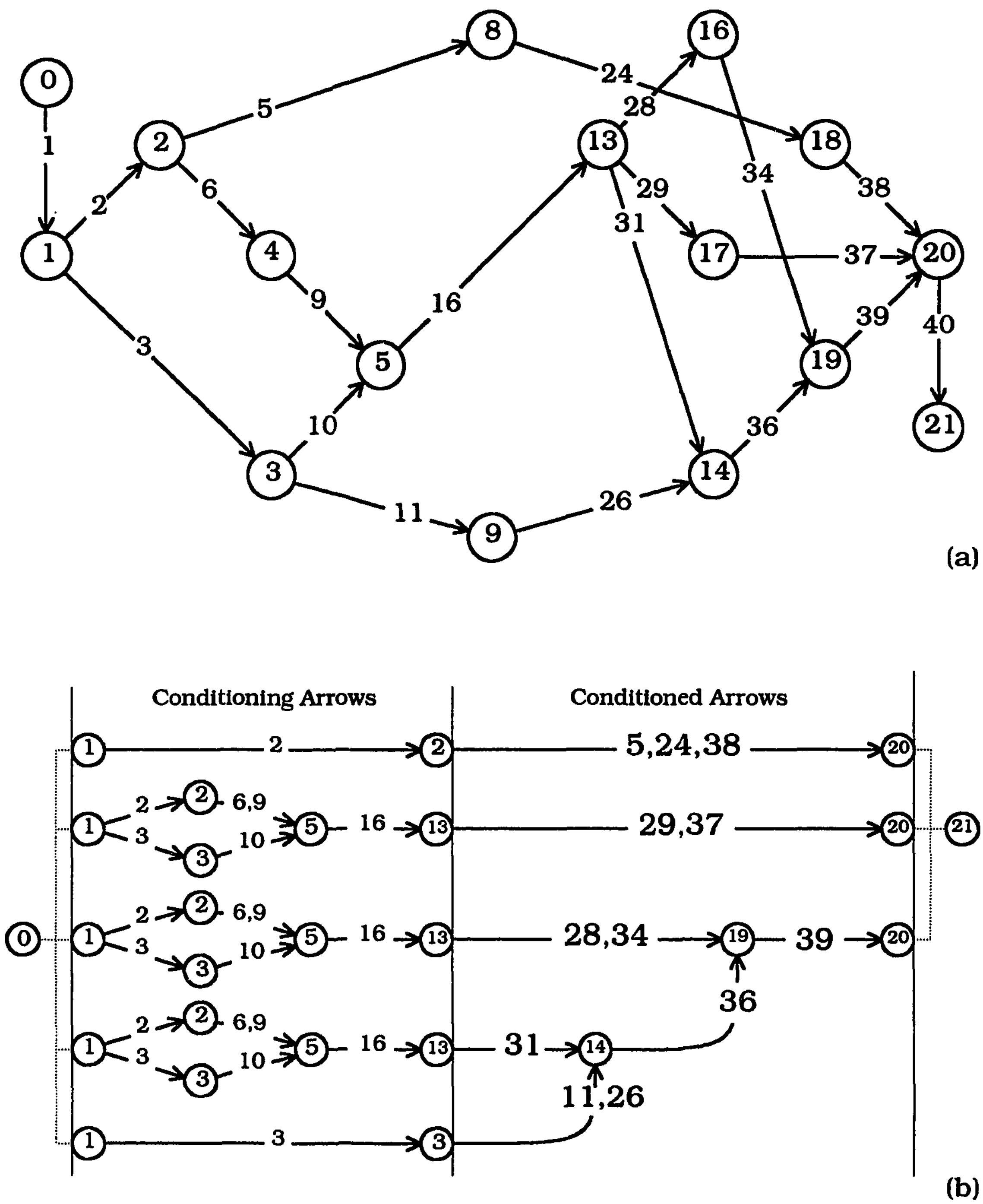


Figure 10: The Top 6-Path Skeleton of the Kleindorfer (1971) Network.
(a) The best skeleton network. (b) The conditioned skeletal network.

Repeating the series reduction operation 6 times reduces the number of skeletal arrows to 14 and the number of arrow duration combinations to 4.663×10^{10}. The 6 series reduction operations are shown in Table 11. The first column gives the number of the new duration distribution calculated by series reducing the 2 arrow duration distributions appearing in the second and third columns. The fourth column expresses the reduction operations in symbolic form. The "A" before the numbers in Table 11 define arrow duration distributions. If the number following the "A" is 40 (the number of arrows in the Kleindorfer network) or less, the arrow duration distribution is given as part of the Kleindorfer network input. Arrow numbers over 40 represent derived arrow duration distributions. In the symbolic equation, the "~" is read "is distributed as" while the "⊕" represents the distribution series reduction operation of Equation (1). After 6 series reduction operations, no more series or parallel reduction operations are possible. Since the skeletal network has more than 1 arrow at this point, it is not a parallel-series network. Conditioning the network is required for analysis.

Conditioning the 14-arrow skeletal Kleindorfer network is accomplished by conditioning on arrows that come before Splitting Nodes or after Merging Nodes. The skeletal network Splitting Nodes are Nodes 1, 2, 3, 5, and 13, and the Merging Nodes are Nodes 5, 13, 14, 19, and 20. Arrows 1, 2, 3, 6, and 9 precede the Splitting Nodes and have 3,120 combinations of arrow durations, after series reducing Arrows 6 and 9. Arrows 16, 28, 29, 31, 34, 36, 37, 38, 39, and 40 follow the Merging Nodes and have 1,033,695 combinations of arrow durations, after series reducing Arrows 28 and 34 and reducing Arrows 29 and 37. The Kleindorfer Conditioning Arrows are assigned to the arrows between the Source Node and before the Splitting Nodes. Arrows 1 and 40 have zero durations with a probabilities of 1.000. They have no effect on the network

Table 11: Series Reduced Duration Distributions in Skeletal Network.

Reduced Distribution	Arrow Duration Distributions		Reduction Operation
A41	A5	A24	$A41 \sim A5 \oplus A24$
A42	A6	A9	$A42 \sim A6 \oplus A9$
A43	A11	A26	$A43 \sim A11 \oplus A26$
A44	A28	A34	$A44 \sim A28 \oplus A34$
A45	A29	A37	$A45 \sim A29 \oplus A37$
A46	A41	A38	$A46 \sim A41 \oplus A38$

statistics. Modern computers are quite capable of performing analysis on 3,120 combinations in reasonable time. Figure 10b illustrates the skeletal Kleindorfer network arrangement of Conditioning Arrows and Conditioned Arrows. Series reduced arrow distributions are indicated by listing the original Kleindorfer arrow numbers.

For each combination of Conditioning Arrow durations, the Conditioned Arrow duration distributions are combined in parallel and series distribution reduction operations to calculate the conditional network distribution. Table 12 lists the conditional reduction operations for the skeletal network's Merging Parallel Tree. In Table 12, an "A" indicates an original Kleindorfer network arrow distribution, an "N" indicates a Kleindorfer network node number, and an "S" indicates either a Starting Arrow duration distribution or another Accumulated Arrow Duration Distribution. The . ⊗ . symbol indicates a parallel distribution reduction operation via Equation (3) and Equation (4).

Duration probabilities from the conditional network duration distributions are multiplied by the conditional probability and summed across conditional combinations to calculate the unconditional probabilities for all network durations. The network duration distribution from the algorithm is shown in Figure 11.

Table 12: Conditioned Arrow Duration Distribution Reduction Operations.

Reduced Distribution	Reduction Duration Distributions			Reduction Operations
S38	N2		A46	S38 ~ N2 ⊕ A46
S37	N13		A45	S37 ~ N13 ⊕ A45
S34	N13		A44	S34 ~ N13 ⊕ A44
S31	N13		A31	S31 ~ N13 ⊕ A31
S26	N3		A43	S26 ~ N3 ⊕ A43
N14	S26		S31	N14 ~ S26 ⊗ S31
S36	N14		A36	S36 ~ N14 ⊕ A36
N19	S34		S36	N19 ~ S34 ⊗ S36
S39	N19		S39	S39 ~ N19 ⊕ A39
N20	S38	S37	S39	N20 ~ S38 ⊗ S37 ⊗ S39
S40	N20		A40	S40 ~ N20 ⊕ A40
N21	S40			N21 ~ S40

Up to duration 70, the algorithm's network duration distribution falls within Shogan's bounds. For durations 71 to 89, the algorithm's cumulative distribution function is below Shogan's. Since the algorithm probabilities in that range are exact, Shogan's formula is wrong. To be sure, the region of error only involves 4% of the distribution, but that is the 4% which schedulers are most concerned about!

The SLAM II simulation program was constructed according the to procedure described by Pritsker (1995). The simulated network duration distribution in Figure 11 is the collective results of 10 simulation runs of 10,000 network samples each. Each run used a different random number stream using the default random seeds. Out of a total of 100,000 simulations, SLAM II observed a maximum duration of 87 with 1 realization. From algorithm calculations, network durations being greater than 87 occur on average 1:623,135 realizations. The probability of SLAM II observing either duration 88 or 89 is about 0.148 according to the Poisson distribution (Snedecor & Cochran, 1989). With over 85% probability of not observing the 2 highest durations, the simulation runs are not invalidated.

A Chi-square Goodness-of-fit test was performed on the right-tail portion of the Kleindorfer network duration distribution over the durations 68 to 89. The null hypothesis is that the distributions are identical, and the alternative hypothesis is that they are not. The SLAM II simulation provided the observed values, and the algorithm

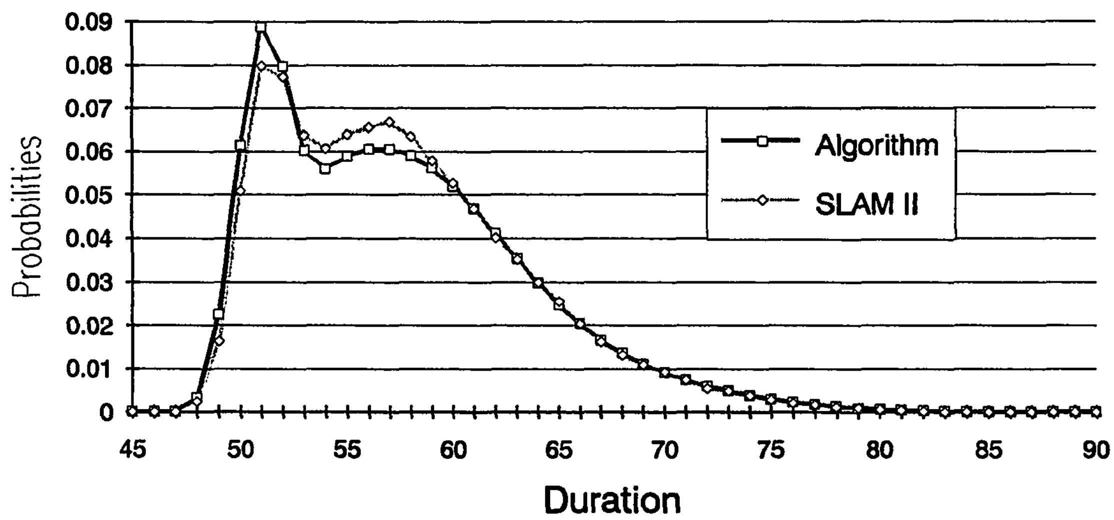


Figure 11: The Duration Distribution for the Kleindorfer (1971) Network.

provided the expected values by multiplying its calculated duration probabilities times 100,000, the number of simulation runs. The durations from 85 to 89 were combined into one cell because Walpole & Myers (1978) recommends at least 5 observations per cell. The chi-square statistic with 18 degrees of freedom was 18.804. Whereas the 95% critical value of the Chi-square distribution is 28.869, the null hypothesis was not rejected, and the distributions are assumed to be statistically identical. Sampling error, as described in the previous paragraph, explains why the SLAM II simulation did not observe the duration of 88 and 89.

Both the algorithm and SLAM II predict a bimodal distribution. Compared to the SLAM II estimate of the Kleindorfer network duration distribution, the algorithm overestimates the probabilities associated with the first mode and underestimates probabilities in the second mode. This is due to limited information from the middle part of the distribution. The run time of the algorithm without calculating Van Slyke criticalities was 3-minutes and 15-seconds on a same 486DX/50 personal computer using Microsoft's FORTRAN PowerStation. The total run time of the 100,000 SLAM II simulations was over 1-hour and 45-minutes on the same 486DX/50 personal computer. The Pritsker (1995) PERT simulation model does not calculate criticalities.

Algorithm Van Slyke Criticalities In Kleindorfer Network

Arrow Van Slyke criticality calculations for the Kleindorfer network are based upon the skeletal network depicted in Figure 10b and the Van Slyke criticality calculation methods introduced in the previous chapter. Van Slyke criticality calculation formulas based upon Equation (16) for the Conditioned Arrows are given in Table 13. The equations take, as input, the conditional duration distributions.

The Conditioning Arrow Van Slyke criticalities are joint probabilities of the Conditioned Arrow Van Slyke criticalities. The skeletal network Conditioning Arrows end in three Splitting Nodes: Node 2, Node 13, and Node 3 (see Figure 10b). Paths containing the Conditioning Arrows may be critical to 0 or 1 or 2 of these nodes depending on the relative values of the Conditioning Arrow durations. The skeletal network precedents do not allow an arrow to be on critical paths that reach all 3 nodes. Arrow 16 is always Van Slyke critical when at least 1 of the Starting Arrows starting with Node 13 are critical. The Arrow 16 Van Slyke criticality is the union of Van Slyke

Table 13: Van Slyke Criticality Equations for Conditioned Arrows.

Arrows	Van Slyke Criticality Equation
5,24,38	$\sum_{i \in S38} f_{S38_i} F_{S37_i} F_{S39_i}$
29,37	$\sum_{i \in S37} f_{S37_i} F_{S38_i} F_{S39_i}$
28,34	$\sum_{i \in S34} f_{S34_i} F_{S36_i} \left[\sum_{j \in A39} f_{A39_j} F_{S37_{i+j}} F_{S38_{i+j}} \right]$
36	$\sum_{i \in S36} f_{S36_i} F_{S34_i} \left[\sum_{j \in A39} f_{A39_j} F_{S37_{i+j}} F_{S38_{i+j}} \right]$
39	$\sum_{i \in S39} f_{S39_i} F_{S37_{i+j}} F_{S38_{i+j}}$
11,26	$\sum_{i = S26} f_{S26_i} F_{S31_i} \left\{ \sum_{j \in A36} f_{A36_j} F_{S34_{i+j}} \left[\sum_{k \in A39} f_{A39_k} F_{S37_{i+j+k}} F_{S38_{i+j+k}} \right] \right\}$
31	$\sum_{i = S31} f_{S31_i} F_{S26_i} \left\{ \sum_{j \in A36} f_{A36_j} F_{S34_{i+j}} \left[\sum_{k \in A39} f_{A39_k} F_{S37_{i+j+k}} F_{S38_{i+j+k}} \right] \right\}$

criticalities for Starting Arrows 37, 34, and 31. The Van Slyke criticalities and Van Slyke tying criticalities to sum and subtract to calculate Arrow 16 are given in Table 14.

The Conditioning Arrows portion of the skeletal network has a Merging Node. The merger necessitates determining whether the merging paths are critical to Merging Node 5 (and subsequently to Node 13). A Conditioning Arrows on a critical path must add the Node 13's Starting Arrow Van Slyke criticalities to the arrow's union of Starting Arrow Van Slyke criticalities. If 1 of the Conditioning Arrows 6, 9, and 10 is not on a critical path to Node 13, its Van Slyke criticality is 0.000. Arrow 2 is always on the critical path to Node 2, and is critical to Node 13 if the sum of the conditional durations of Arrows 2, 6, and 9 (the Arrows 6 and 9 are actually represented by the series reduced Arrow 42). Arrow 3 is always on a critical path to Node 3 and is sometimes on a critical path to Node 13.

Table 14: Van Slyke Criticalities for Conditioning Arrows.

Arrows	Conditional Van Slyke Criticality Equation	Effectiveness
A2	$Crit_{A2}=Crit_{S38}$	$S9 \geq S10$
	$Crit_{A2}=Crit_{S37}+Crit_{S34}+Crit_{S31}+Crit_{S38}$ $-Crit_{S37,S34}-Crit_{S37,S31}-Crit_{S37,S38}-Crit_{S34,S31}-Crit_{S34,S38}-Crit_{S31,S38}$ $+Crit_{S37,S34,S31}+Crit_{S37,S34,S38}+Crit_{S37,S31,S38}+Crit_{S34,S31,S38}$ $-Crit_{S37,S34,S31,S38}$	$S9 < S10$
A16	$Crit_{A16}=Crit_{S37}+Crit_{S34}+Crit_{S31}$ $-Crit_{S37,S34}-Crit_{S34,S31}-Crit_{S31,S37}$ $+Crit_{S37,S34,S31}$	Everywhere
A6,A9	$Crit_{A6}=Crit_{A9}=Crit_{S37}+Crit_{S34}+Crit_{S31}$ $-Crit_{S37,S34}-Crit_{S34,S31}-Crit_{S31,S37}$ $+Crit_{S37,S34,S31}$	$S9 \geq S10$
	$Crit_{A6}=Crit_{A9}=0.000$	$S9 < S10$
A10	$Crit_{A10}=Crit_{S37}+Crit_{S34}+Crit_{S31}$ $-Crit_{S37,S34}-Crit_{S34,S31}-Crit_{S31,S37}$ $+Crit_{S37,S34,S31}$	$S9 \leq S10$
	$Crit_{A10}=0.000$	$S9 > S10$
A3	$Crit_{A3}=Crit_{S26}$	$S10 \geq S9$
	$Crit_{A3}=Crit_{S39}+Crit_{S37}$ $-Crit_{S39,S37}$	$S10 < S9$

The probabilities of Van Slyke criticality unions for the Conditioning Arrows are given in Table 14. The Conditioning Arrow durations To determine whether the merging paths formed from Conditioning Arrows are critical, the conditional path durations to Node 5 are calculated. Table 2 paths are referenced in the Effectiveness column of Table 14. The path durations are defined as $S9 \sim A2 \oplus A6 \oplus A9$ and $S10 \sim A3 \oplus A10$.

The union of Van Slyke Starting Arrow criticalities involve calculating the probabilities that some sets of Conditioning Arrows are both Van Slyke Critical and identical in path duration to each other. Formulas for the tying Van Slyke criticalities listed in Table 14 are given in Table 15. The probabilities calculated from Table 15 are substituted into the formulas of Table 14 to calculate the Conditioning Arrow Van Slyke criticalities.

The algorithm calculations of the Van Slyke arrow criticalities are listed in Table 16. Table 16 also contains the simulated estimates of Van Slyke arrow criticalities from 1,000,000 FORTRAN network simulations and the highest ranking path number that the arrows appear. The algorithm assumes that arrows not in the skeletal network have zero probabilities. This, of course, is not always the case, but the goal is to identify the top arrow criticalities. An index to the importance of arrows not quantified by the skeletal network is the highest path rank of the arrow. Arrows with higher criticalities tend to be on higher ranked paths.

The algorithm identified all arrows with Van Slyke criticalities above 0.100000. These same arrows are particularly important when considering the right-end of the network duration distribution. At lower network durations, many more arrows may be critical. While it is difficult to predict before hand which arrow will be critical, it is probable that the duration will not be extreme. It is only the arrows that contribute to project overrun that must be monitored and managed.

Some of the Kleindorfer arrows actually had zero criticalities. Arrow 33 was predicted to have zero criticality by the algorithm because its Highest Path Rank was 50, lower than rank 45 rank needed to compete for the lower bound of the network distribution. The other arrows with zero criticality were Arrows 13, 15, 21, 23, and 27. Arrow 13 is never critical because Arrow 5 on the path ranked number 2 dominates the series of Arrows 6, 8, and 13. Arrow 14 has some probability of being critical , so Arrow 8 has the same criticality. Arrow 15 is dominated because the maximum duration of the path subset of Arrows 6, 8, and 15 is 14 which is shorter than the path subset of 5, 19, and 25. Arrows 21, 23, and 27 are dominated by Path 27, the one with the longest duration, whose minimum duration from Node 2 to Node 18 is 26 whereas the maximum duration from Node 2 to Node 18 on Path 1 is only 22. Node 15 and its connecting arrows have no effect on the duration of the network. Besides Arrow 33, the other arrows whose criticality was zero were dominated by the top ranked paths. This is an excellent example of the significant few versus the trivial many concept.

Table 15: Equations to Calculate Tying Van Slyke Criticalities.

Tying Critical Arrows	Tying Van Slyke Criticality Equation
S34,S37	$\sum_{i \in N20} f_{S37_i} F_{S38_i} \left(\sum_{j \in A39} f_{A39_j} f_{S34_{i-j}} f_{S36_{i-j}} \right)$
S31,S34	$\sum_{i \in N20} F_{S37_i} F_{S38_i} \left[\sum_{j \in A39} f_{A39_j} f_{S34_{i-j}} \left(\sum_{k \in A36} f_{A36_k} f_{S31_{i-j-k}} F_{S26_{i-j-k}} \right) \right]$
S31,S37	$\sum_{i \in N20} f_{S37_i} F_{S38_i} \left[\sum_{j \in A39} f_{A39_j} F_{S34_{i-j}} \left(\sum_{k \in A36} f_{A36_k} f_{S31_{i-j-k}} F_{S26_{i-j-k}} \right) \right]$
S37,S38	$\sum_{i \in N20} f_{S37_i} f_{S38_i} F_{S39_i}$
S34,S38	$\sum_{i \in N20} F_{S37_i} f_{S38_i} \left(\sum_{j \in A39} f_{A39_j} f_{S34_{i-j}} F_{S36_{i-j}} \right)$
S31,S38	$\sum_{i \in N20} F_{S37_i} f_{S38_i} \left[\sum_{j \in A39} f_{A39_j} F_{S34_{i-j}} \left(\sum_{k \in A36} f_{A36_k} f_{S31_{i-j-k}} F_{S26_{i-j-k}} \right) \right]$
S37,S39	$\sum_{i \in N20} f_{S37_i} F_{S38_i} f_{S39_i}$
S31,S34,S37	$\sum_{i \in N20} f_{S37_i} F_{S38_i} \left[\sum_{j \in A39} f_{A39_j} f_{S34_{i-j}} \left(\sum_{k \in A36} f_{A36_k} f_{S31_{i-j-k}} F_{S26_{i-j-k}} \right) \right]$
S34,S37,S38	$\sum_{i \in N20} f_{S37_i} f_{S38_i} \left[\sum_{j \in A39} f_{A39_j} f_{S34_{i-j}} F_{S36_{i-j}} \right]$
S31,S37,S38	$\sum_{i \in N20} f_{S37_i} f_{S38_i} \left[\sum_{j \in A39} f_{A39_j} F_{S34_{i-j}} \left(\sum_{k \in A36} f_{A36_k} f_{S31_{i-j-k}} F_{S26_{i-j-k}} \right) \right]$
S31,S34,S38	$\sum_{i \in N20} F_{S37_i} f_{S38_i} \left[\sum_{j \in A39} f_{A39_j} f_{S34_{i-j}} \left(\sum_{k \in A36} f_{A36_k} f_{S31_{i-j-k}} F_{S26_{i-j-k}} \right) \right]$
S31,S34,S37,S38	$\sum_{i \in N20} f_{S37_i} f_{S38_i} \left[\sum_{j \in A39} f_{A39_j} f_{S34_{i-j}} \left(\sum_{k \in A36} f_{A36_k} f_{S31_{i-j-k}} F_{S26_{i-j-k}} \right) \right]$

Table 16: Arrow Criticalities of the Kleindorfer (1971) Network.

Arrow Number	Criticalities		Highest Path Rank
	Algorithm	Simulation	
1	1.000000	1.000000	1
2	0.200265	0.229722	1
3	0.841144	0.829699	2
4	0.000000	0.000202	12
5	0.197979	0.227529	1
6	0.002559	0.002552	5
7	0.000000	0.000136	13
8	0.000000	0.000220	11
9	0.002559	0.002225	5
10	0.472573	0.471929	3
11	0.394500	0.382064	2
12	0.000000	0.000309	12
13	0.000000	0.000000	21
14	0.000000	0.000220	11
15	0.000000	0.000000	40
16	0.472875	0.472158	3
17	0.000000	0.000013	29
18	0.000000	0.000072	23
19	0.000000	0.013110	17
20	0.000000	0.001818	14
21	0.000000	0.000000	27
22	0.000000	0.000309	12
23	0.000000	0.000000	22
24	0.197979	0.217371	1
25	0.000000	0.014727	14
26	0.394500	0.380556	2
27	0.000000	0.000000	22
28	0.063258	0.103776	6
29	0.321469	0.300266	3
30	0.000000	0.014730	14
31	0.113210	0.096355	4
32	0.000000	0.070019	7
33	0.000000	0.000000	50
34	0.063258	0.042040	6
35	0.000000	0.074506	7
36	0.497730	0.468502	2
37	0.321469	0.310896	3
38	0.197979	0.203606	1
39	0.544076	0.551898	2
40	1.000000	1.000000	1

Testing Of A Bimodal Distribution

The indication of a bimodal distribution for the duration of the Kleindorfer (1971) network is a somewhat surprising result. Literature would seem to indicate the resulting distribution should be normal because of the Central Limit Theorem (Anklesaria & Drezner, 1986; Kaufmann & Desbazeille, 1969; Malcolm, Rosenbloom, Clark, & Fazar, 1959). Much of the literature concentrates on finding the moments, especially the mean and standard deviation of the distribution. Many practitioners unknowingly use the mean and standard deviation to calculate probabilities using the Normal distribution. PERT does that.

The authenticity of the bimodal distribution was investigated. First, the Shogan (1977) bounds for the Kleindorfer network cumulative distribution were converted into probability distribution functions. Both the upper and lower bounds had bimodal distributions. Shogan did not mention this. Perhaps this is why the results were presented in tabular cumulative form. Another verification was the 10 runs of 10,000 SLAM II simulations. In every one of the 10 runs using different random number streams, the bimodal distribution was apparent. Some of the runs showed small additional modes, but 2 modes were apparent in all 10 runs. The probability of this happening by chance is small. The distribution generated by the 1,000,000 FORTRAN simulation runs also showed 2 modes.

Although the Central Limit Theorem has worked well adding many random variables, there is another phenomenon in networks. There are maximums of variables in parallel, not ones in series. The resulting distribution is more analogous to the Extreme Value Distribution (Dodin & Sirvanci, 1990). The mechanisms of bimodal distribution generation warrants additional comment. The next chapter does this.

A BIMODAL DISTRIBUTION FROM NORMAL DISTRIBUTIONS

Bimodal distributions are not typically found in statistical analysis procedures. Typical modeling distributions are the Exponential, Gamma, Poisson, Beta, Normal, the t-distribution, the F-distribution, the Chi-Squared distribution, binomial, geometric, Weibull, triangular, and uniform. None of which are bimodal.

PERT uses the Normal distribution for its estimate of the project duration (Malcolm, Rosenbloom, Clark, & Fazar, 1959). Since then, publications explaining the PERT method and much of the subsequent research have perpetuated the normality assumption. Some who have made this assumption are Anklesaria & Drezner (1986) and Kaufmann & Desbazeille (1969). Dodin & Sirvanci (1990) suggest the duration of projects are somewhere between the Normal and the Extreme Value distributions, both of which are unimodal.

The possibility of more than one mode of a duration distribution was noted by Charnes, Cooper, & Thompson (1964): "The distribution of completion times ... may often be multimodal, contrasting with (erroneous) central limit theorem usages in the literature" (p. 460). They summarize "multimodality is to be expected whenever there are parallel links or chains that alternate in criticality and that involve sufficiently different times" (Charnes, Cooper, & Thompson, 1964, p. 468).

Because of the central limit theorem, the sum of many random variables tends to be normally distributed. Therefore, the distribution of a single path, without regarding its interactions with other paths, will tend to be normally distributed because its duration is the sum of its constituent arrow durations which are randomly distributed variables. The complication comes when parallel paths contend for the maximum project duration.

Clark (1961), one of the original PERT authors, addressed the problem of estimating four distribution moments from the maximum of two paths that have joint normally distributed durations. He starts his analysis by transforming the distribution of one path into a standard, normal distribution. He then represents jointly distributed normal distributions by expressing three parameters as a function of the transformed distribution. Clark recognized that the maximum of two normals would not be normal, but he did not report any bimodal distributions.

The bimodal phenomenon observed for the Kleindorfer (1971) network was the result of taking the maximums from at least 2 distributions. The algorithm combined the arrows from 6 different paths into a skeletal network. The paths with the longest maximum and minimum durations (the paths are ranked 1 and 2) are independent from each other -- they have no arrows in common except for Arrows 1 and 40 which have fixed durations. The algorithm's bimodal duration distribution (see Figure 11) appears to have the peaks of 2 distributions. The peak with shorter durations corresponds to the midpoint between the minimum and maximum durations of the number 1 ranked path. The other peak appears to be a continuation of the distribution that forms the right duration distribution tail. The right hump in the bimodal distribution must be from the 5 paths having the longest maximum durations.

The maximum covariance between the number 1 ranked path and any other path is 0.5, the variance of Arrow 2. The variance of the number 1 ranked path, Path 30, is 1.0 if the path duration range is assumed to be 6 standard deviations wide. Path 39 has the smallest duration range of the 3 other paths that include Arrow 2. Similarly, Path 39 has a variance of 46.7. The maximum correlation between Path 30 and Path 39 is then 0.07. For all practical purposes, Path 30 is independent of all other paths through the skeletal Kleindorfer network.

The analysis of bimodal duration distributions will assume the phenomenon is a function of 2 parallel paths. The path durations are further assumed to be normally distributed. Thus, the analysis will be the effects of taking the maximums of 2 independent, normal, and discrete distributions. The analysis proceeds similarly to the study performed by Clark (1961), but the focus is on distribution modes, and no correlation effects will be examined. Figure 12 shall be the model network for analysis. Network duration distributions are computed by the Parallel Reduction Operation, using Equations (3) and (4).

Like Clarke (1961), the bimodal analysis adapts parametric measures based upon Arrow 1's standard deviation, σ_1. Assume, without loss of generality, that Arrow 2's standard deviation is greater than or equal to Arrow 1's standard deviation ($\sigma_1 \leq \sigma_2$). The ratio of $\frac{\sigma_2}{\sigma_1}$ defines a parameter. Call this parameter "R" for ratio.

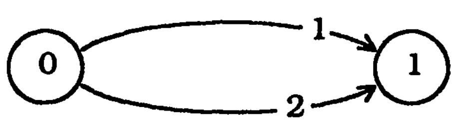


Figure 12: Condensed Network Model for Bimodal Distribution Analysis.

$$\boxed{R = \frac{\sigma_2}{\sigma_1}} \qquad (17)$$

The only other statistic that could effect the comparison between the distribution shapes of Arrows 1 and 2 is the difference in their means. The difference, rather than the absolute value, of the means is important because the absolute-valued means shift the distribution along the abscissa, but the difference in the means determine where one mean falls on the distribution of the other arrow's duration. This difference is also expressible as a function of σ_1: $\mu_2 - \mu_1 = D\sigma_1$.

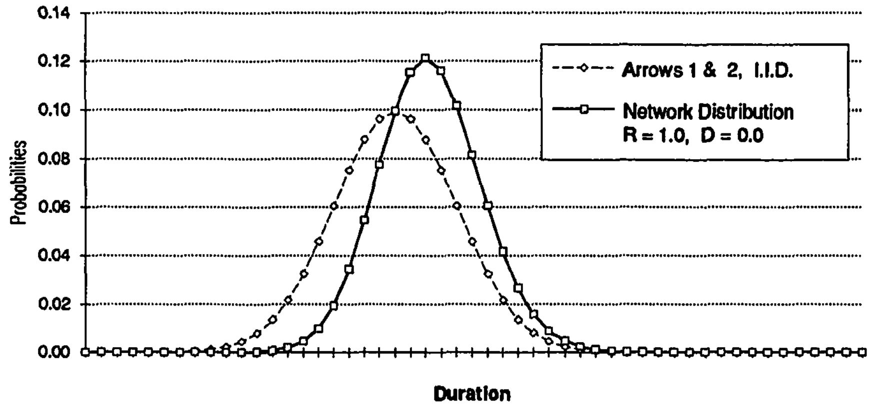
$$\boxed{D = \frac{\mu_2 - \mu_1}{\sigma_1}} \qquad (18)$$

The distribution shape of the duration of the project in Figure 12 is represented by 3 parameters: R and D.

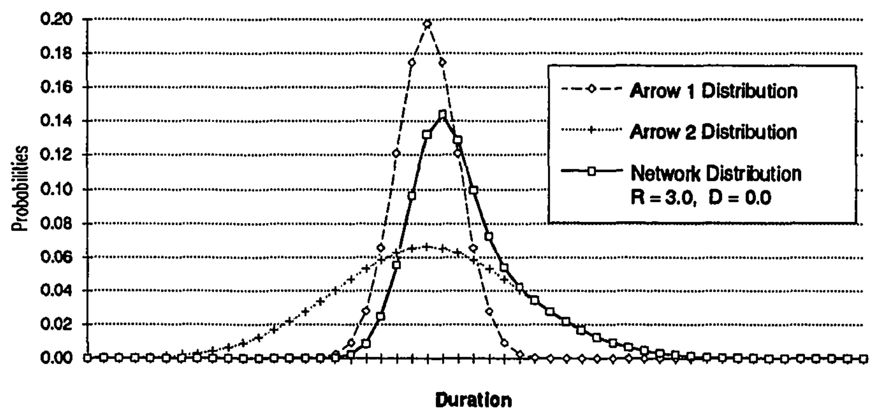
For the case of R=1.00 and D=0.00, the durations of Arrows 1 and 2 are identically distributed. The resulting distribution appears in Figure 13. The maximum duration distribution is shifted to the right of the 2 original duration distributions. The Extreme Value distribution (Dodin & Sirvanci, 1990) predicts this. If a large number of parallel, identically distributed were reduced with the parallel operation, the results would be the Extreme Value distribution. The reduced distribution is clearly unimodal.

As the ratio "R" is increased while the means of Arrows 1 and 2 remain the same, the effect is to gather up the left tail of the larger variance distribution into the distribution of the smaller variance arrow (see Figure 13 a, b, & c). Changing the shift of means between Arrows 1 and 2, on the other hand, tends to skew the lower mean's distribution toward the right. As the difference between the means tends to get above 2 standard deviations (they both have the same standard deviation at this point), the maximum distribution becomes, essentially, the larger mean arrow's distribution (see Figure 14).

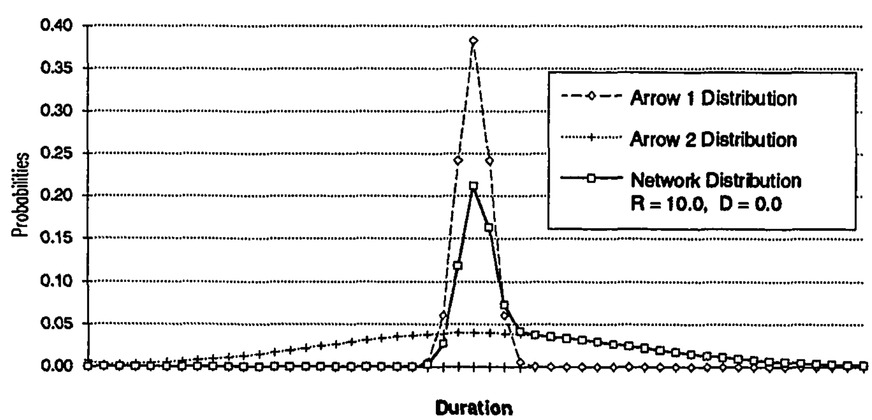
As "R" and "D" are varied, the bimodal distribution is created. Figure 15 defines indicators for unusually shaped duration distributions. Figure 15a and Figure 15d shows instances where the distribution in transition from a unimodal to a bimodal distribution. The author of this thesis calls these "lumpy" distributions. If these lumps and bimodals are encoded, they may be simulated and tabulated. The lumpy left tailed distribution of Figure 15a is labeled as "-1". The two bimodal distributions, Figure 15b



(a)

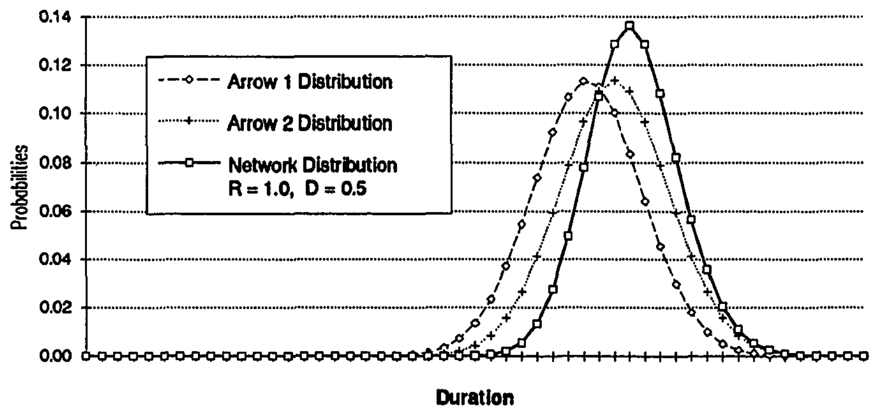


(b)

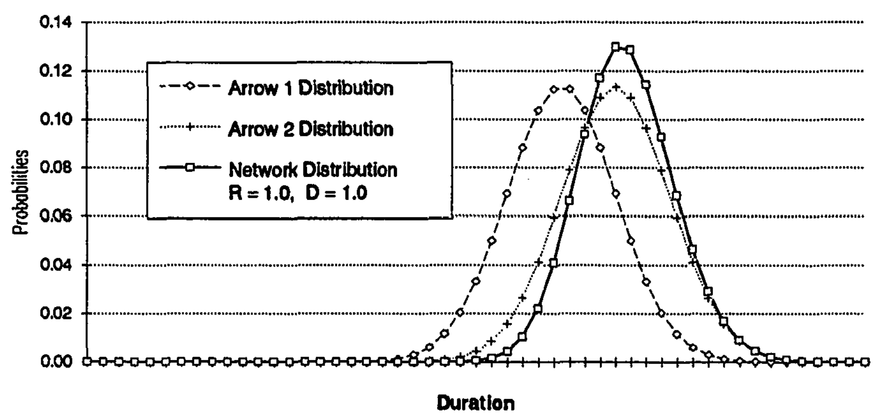


(c)

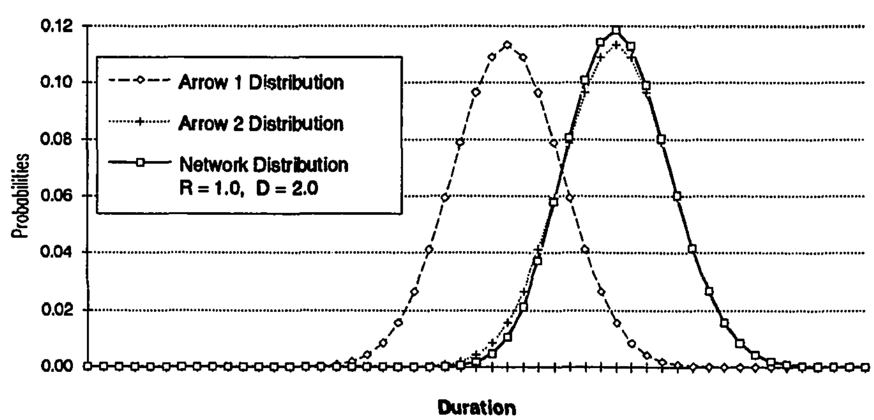
Figure 13: Effects of Standard Deviation on the Maximum of 2 Distributions.
(a) The maximum of 2 Independent, Identically distributed Normal Distributions.
(b) Maximum Distribution of common means, unequal variance distributions.
(c) A maximum distribution with a sharply curtailed left-tail.



(a)



(b)



(c)

Figure 14: Effects of Mean Differences on the Maximum of 2 Distributions.
(a) The effects of a small shift in parallel means on the maximum distribution.
(b) The effects of a moderate shift in parallel means on the maximum distribution.
(c) Moderately large shift in mean differences result in decreasing impact.

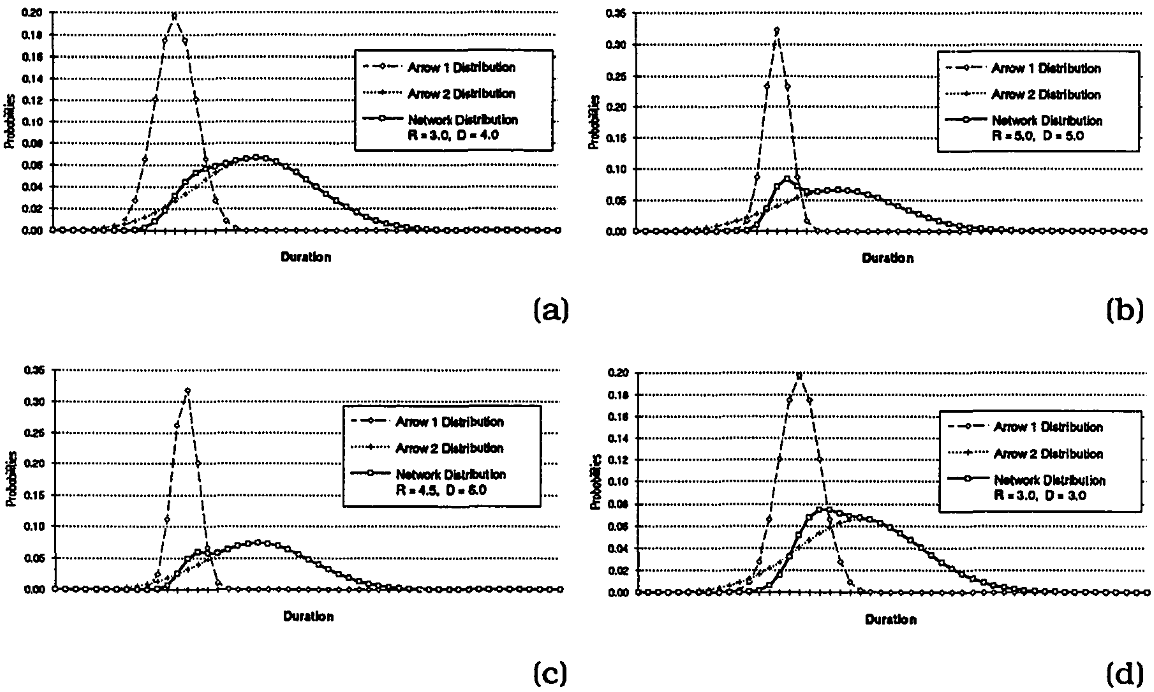


Figure 15: Bimodal and Lumpy Distributions.
(a) A lumpy left-tailed distribution. Coded as "-1".
(b) A bimodal distribution with the greater mode on the right. Coded as "2".
(c) A bimodal distribtion with the greater mode on the left. Coded as "-2".
(d) A lumpy right-tailed distribution. Coded as "1".

and Figure 15c are encoded with a "2", with a plus or minus in front depending on whether the left mode or the right mode is greater. The minus is indicated when the lesser mode is greatest. A right lumpy tail gets a "1" encoding. The distribution results of varying "R" and "D" are shown in Table 17. Unimodal, nonlumpy distributions are encoded by a "0" in Table 17.

Bimodal distributions may occur when the distribution of Arrow 1 has a higher variance than Arrow 2, but Arrow 1's mean is much lower. Mixing the distribution from Arrow 2 with the distribution of Arrow 1 will yield a progression of resulting distributions. If Arrow 1's mean is much less than Arrow 2's mean, say $6*(\sigma_1+\sigma_2)$, the resultant distribution will be just that of Arrow 2. The distribution progresses from a lump in the left tail (coded as "-1") to a bimodal distribution with the left mode smaller than the right (coded as "2") as Arrow 1's mean approaches Arrow 2's. As the mean of

Table 17: Bimodal and Lumpy Distributions for Parallel Paths.

	$D = \frac{\mu_2 - \mu_1}{\sigma_1}$																				
$R = \frac{\sigma_2}{\sigma_1}$	0.0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.5	6.0	6.5	7.0	7.5	8.0	9.0	10.0	11.0	12.0
0.0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2.5	0	0	0	1	1	1	0	0	0	-1	-1	0	0	0	0	0	0	0	0	0	0
3.0	0	0	0	1	1	1	1	2	-1	-1	-1	-1	-1	-1	0	0	0	0	0	0	0
3.5	0	0	1	1	1	1	1	1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0	0
4.0	0	0	1	1	1	1	1	1	-2	-2	2	2	-1	-1	-1	-1	-1	-1	-1	0	0
4.5	0	1	1	1	1	1	1	1	-2	-2	-2	2	2	2	-1	-1	-1	-1	-1	-1	-1
5.0	1	1	1	1	1	1	1	1	-2	-2	-2	-2	2	2	2	2	2	-1	-1	-1	-1
5.5	1	1	1	1	1	1	1	1	1	-2	-2	-2	-2	-2	2	2	2	2	-1	-1	-1
6.0	1	1	1	1	1	1	1	1	1	-2	-2	-2	-2	-2	-2	2	2	2	2	2	-1
6.5	1	1	1	1	1	1	1	1	1	-2	-2	-2	-2	-2	-2	-2	-2	2	2	2	2
7.0	1	1	1	1	1	1	1	1	1	-2	-2	-2	-2	-2	-2	-2	-2	-2	2	2	2
7.5	1	1	1	1	1	1	1	1	1	-2	-2	-2	-2	-2	-2	-2	-2	-2	2	2	2
8.0	1	1	1	1	1	1	1	1	1	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	2	2
9.0	1	1	1	1	1	1	1	1	1	1	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2
10.0	1	1	1	1	1	1	1	1	1	1	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2
11.0	1	1	1	1	1	1	1	1	1	1	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2
12.0	1	1	1	1	1	1	1	1	1	1	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2

Arrow 1 increases further, the maximum distribution advances from a bimodal with a higher right mode to a bimodal with a higher left mode (coded −2). As Arrow 1's mean gets very much greater than Arrow 2's, the distribution becomes just that of Arrow 1's. In this respect, it makes no difference whether A's is the bigger distribution or vice versa.

This analysis shows that the bimodal distribution is real. It is a tribute to the algorithm that it can generate bimodal distributions. In general, if the top 2 paths have means closer than 3 standard deviations, bimodality does not occur. From algorithm data, the possibility of the network duration distribution being either lumpy or bimodal can be predicted by whether or not the path with the longest minimum duration is also one of the longest maximum duration paths. If the top ranked path does not also have a long maximum duration, there is chance for a lumpy or bimodal duration distribution. If the longest minimum duration path is also one of the longest maximum duration paths, there is no danger of a lumpy distribution. By the top ranked path being long in both minimum and maximum durations, its probabilities close to the minimum network duration are reduced and the other long maximum duration paths can interact to make a smooth unimodal distribution.

SUMMARY

A procedure was developed that improves upon CPM and PERT (see Table 18). CPM calculates schedule durations and the criticality of activities but only for deterministic inputs. PERT starts with probabilistic activity durations and calculates project duration distributions but consistently estimates shorter-than-actual project durations, and does not provide a measure of arrow criticality. The Singleton procedure uses integer-valued activity duration distributions, and calculates duration distributions and Van Slyke (1963) criticalities for network arrows.

The Singleton analysis is relevant to industry practice in at least the following three ways. First, integer-valued durations are assumed because industry typically monitors project progress and allocates resources on a periodic basis. Integer durations also allow for the explicit analysis of tied path durations--industry experiences simultaneous task completions, but analysis with continuous distributions does not allow for ties. Second, Van Slyke criticality analysis focuses on arrows rather than paths, in keeping with the fact that industry allocates resources to activities, not paths. Third, emphasis is placed upon the project duration range and upon the probabilities of the largest project durations when complete network analysis is impractical. This, again, is in keeping with project schedulers selecting dates to include all but the largest project durations, and project managers endeavoring to prevent the longest durations from occurring.

Probabilistic integer inputs complicate analysis of scheduling networks. For example, the CPM method of taking the maximum path duration to any node is complex when path durations have distributions and the distribution of the maximum of possibly correlated path durations involved. The distribution of the maximum of path durations has previously been studied. Singleton adapts the work of Martin (1965) in calculating project duration distributions. Martin's methods of calculating duration probability

Table18: Comparison of Singleton Algorithm with CPM and PERT.

	Singleton	CPM	PERT
Probability Input?	Yes	No	Yes
Critical Activities?	Yes	Yes	No
Distribution Output?	Yes	No	Yes

distributions by using polynomial equations are converted into calculations using discrete probabilities. These calculations become demanding for large networks.

Singleton develops methods to calculate Van Slyke criticalities. The notion of cumulative distributions is natural here, in regard to the treatment of dominance and ties. Other key concepts are those of joint distributions, conditional probabilities, and the Law of Total Probability. The Singleton methods to calculate the Van Slyke criticalities are limited by the number of computations required for larger networks, as are the duration distribution computations.

By observing that a few long path durations typically are dominant, Singleton develops a large-network algorithm that limits the number of calculations needed for large networks. The algorithm finds and ranks paths through a network. Arrows from paths with long durations are reconstructed into a skeletal network and analyzed by the Singleton duration distribution and criticality methods.

The large-network algorithm is applied to the 40-arrow, 22-node Kleindorfer (1971) network and was compared against extensive simulations. The algorithm was found to exactly calculate the network duration range and right tail probabilities. The algorithm also correctly selected the activity arrows with the highest Van Slyke criticalities and avoided serious miscalculation of any criticalities. Finally, the algorithm correctly identified the location of the 2 modes of the Kleindorfer duration distribution.

The general mechanism responsible for the creation of such bimodality was investigated by performing an analysis of the maximum duration of 2 parallel paths--normally distributed parallel paths can yield bimodal distributions when both the mean and variance of one path are lower than the other path's mean and variance, because the left tail probabilities of the path with the larger statistics is "accumulated" into the distribution of the path with the lesser statistics.

Extrapolating from these schematic findings, it is seen that the Singleton algorithm will tend to predict a bimodal duration distribution when the path with the longest minimum duration is not also one of the paths with the longest maximum durations. Stronger clues regarding duration distribution bimodality are provided by relevant algorithm-reported path statistics.

BIBLIOGRAPHY

Adlakha, V. G. & Kulkarni, V. G. (1989). A classified bibliography of research on stochastic PERT networks: 1966-1987. INFOR: Canadian Journal of Information Systems and Operations Research, 27, 272-296.

Ahuja, H. N. and Nandakumar, V. (1984). Enhancing reliability of project duration forecasts. AACE Transactions, E.3.6.1-E.3.6.12.

Ahuja, R. K., Magnanti, T. L., & Orlin, J. B. (1993). Network Flows: Theory, Algorithms, and Applications. Englewood Cliffs, NJ: Prentice Hall, Inc.

Anklesaria, K. P. & Drezner, Z. (August, 1986). A multivariate approach to estimating the completion time for PERT networks. Journal of the Operational Research Society, 37, 811-815.

Aubrey, C. A. II & Gryna, D. S. (1991). Revolution through effective improvement projects. ASQC 45th Annual Quality Congress Transactions (pp. 8-13). Milwaukee, WI: American Society for Quality Control, Inc.

Barnetson, P. (1968). Critical Path Planning: Present and Future Techniques. London: Newnes Books.

Berman, E. B. (1964). Resource allocation in PERT network under continuous activity time-cost functions. Management Science, 10(4), 734-745.

Bishop, Y. M. M., Fienberg, S. E., & Holland, P. W. (1975). Discrete Multivariate Analysis. Cambridge, MA: MIT Press.

Bonett, D. G. & Deckro, R. F. (July, 1993). A multinomial project evaluation and review technique for information systems analysis and design. Information & Management, 25, 51-55.

Bowman, R. A. (January, 1995). Efficient estimation of arc criticalities in stochastic activity networks. Management Science, 41(1), 58-67.

Burt, J. M., Jr., & Garman, M. B. (1971). Conditional Monte Carlo: a simulation technique for stochastic network analysis. Management Science, 18, 207-217.

Camp, G. M. (1989). Accelerate or not? - the contractors decision. The Journal of the American Institute of Constructors, 13(2), 5-8.

Carr, R. I. & Maloney, W. F. (June, 1983). Basic research needs in construction engineering. Journal of Construction Engineering and Management, 109 (2), 181-189.

Charnes, A., and Cooper, W. W. (1959). Chance-constrained programming. Management Science, 6, 73-79.

Charnes, A., Cooper, W. W., & Thompson, G. L. (1964). Critical path analysis via chance constrained and stochastic programming. Operations. Research, 12, 460–470.

Chen, J. T. (1983). Modeling learning curve and learning complementarity for resource allocation and production scheduling. Decision Sciences, 14, 170–186.

Cheng, R. C. H. & Iles, T. C. (May, 1988). One-sided confidence bands for cumulative distribution functions. Technometrics, 30(2). 155–159.

Clark, C. E. (March-April, 1961). The greatest of a finite set of random variables. Operations Research, 9(2), 145–162.

Clark, C. E. (May–June, 1962). The PERT model for the distribution of an activity time. Operations Research, 10(3), 405–406.

Coskunoglu, O. (December, 1984). Optimal probabilistic compression of PERT networks. Journal of Construction Engineering and Management, 110. 437–46.

Crandall, D. C. (September, 1976). Probabilistic time scheduling. Journal of Construction Division, ASCE, 102(CO3), 415–423.

Dessouky, M. I. and Dunne, E. J. (1971). Cost-duration analysis with the cut network. AIIE Transactions, 3(1), 123–132.

Diaz, C. F. & Hadipriono, F. C. (March, 1993). Nondeterministic networking methods. Journal of Construction Engineering and Management, 119, 40–57.

Dodin, B. M. (July–August, 1984). Determining the K most critical paths in PERT networks. Operations Research, 32, 859–877.

Dodin, B. M. (May, 1985a). Reducibility of stochastic networks. OMEGA, 13(3), 223–232.

Dodin, B. M. (May–June, 1985b). Approximating the distribution functions in stochastic networks. Computer Operations Research, 12, 251–264.

Dodin, B. M. (July–August, 1985c). Bounding the project completion time distribution in PERT networks. Operations Research, 33, 862–881.

Dodin, B. M. & Elmaghraby, S. E. (February, 1985). Approximating the criticality indices of the activities in PERT networks. Management Science, 31(2), 207–223.

Dodin, B, & Sirvanci, M. (1990). Stochastic networks and the extreme value distribution. Computers & Operations Research, 17, 397–409.

Eisner, H. (1962). A generalized network approach to the planning and scheduling of a research project. Operations Research, 10(1), 115–125.

Elmaghraby, S. E. (1964). An algebra for the analysis of generalized activity networks. Management Science, 10(3), 494–514.

Elmaghraby, S. E. (1967). On the expected duration of PERT type networks. Management Science, 13, 299-306.

Elmaghraby, S. E. (1977). Activity Networks: Project Planning and Control by Network Models. New York: John Wiley & Sons.

Esary, J. D., Proschan, F. & Walkup, D. W. (1967). Association of random variables, with applications. Annals of Mathematical Statistics, 38, 1466-1474.

Fisher, D. L., Saisi, D., & Goldstein, W. M. (1985). Stochastic PERT networks: OP diagrams, critical paths and the project completion time. Computers & Operations Research, 12(5), 471-482.

Frank, H. (1969). Shortest paths in probabilistic graphs. Operations Research, 17(3), 583-599.

Freund, J. E. (1992). Mathematical Statistics. Englewood Cliffs, NJ: Prentice Hall.

Fulkerson, D. R. (1961). A network flow computation for project cost curves. Management Science. 7, 167-178.

Fulkerson, D. R. (1962). Expected critical path lengths in PERT networks. Operations Research, 10, 808-817.

Gallagher, C. (1987). A note on PERT assumptions. Management Science, 33, 1360.

Granof, M. H. (1983). Accounting for Managers and Investors. Englewood Cliffs, NJ: Prentice-Hall, Inc.

Grimaldi, R. P. (1994). Discrete and Combinatorial Mathematics: An Applied Introduction (3rd ed.). Reading, MA: Addison-Wesley Publishing Company.

Grubbs, F. E. (1962). Attempts to validate certain PERT statistics or 'picking on PERT'. Operations Research, 10, 912-915.

Hagstrom, J. N. (1988). Computational complexity of PERT problems. Networks, 18, 139-147.

Hartley, H. O. & Wortham, A. W. (1966). A statistical theory for PERT critical path analysis. Management Science, 12(10), B469-B481.

Hax, A. C. & Candea, D. (1984). Production and Inventory Management. Englewood Cliffs, NJ: Prentice-Hall, Inc.

Healy, T. L. (1960). Activity subdivision and PERT probability statements. Operations Research, 8, 341-350.

Juran, J. M. (1951). Quality-Control Handbook (1st ed.). New York: McGraw-Hill Book Company, Inc.

Kapur, K. C. (1973). An algorithm for project cost-duration analysis problems with quadratic and convex cost functions. AIIE Transactions, 5, 314-322.

Kaufmann, A. & Desbazeille, G. (1969). The Critical Path Method: Application of the PERT method and its variants to production and study programs. New York: Gordon and Breach Science Publishers.

Keefer, D. L. & Bodily, S. E. (1983). Three-point approximations for continuous random variables. Management Science, 29, 595-609.

Keefer, D. L. & Verdini, W. A. (September, 1993). Better estimation of PERT activity time parameters. Management Science, 39, 1086-1091.

Kelley, J. E. (1961). Critical-path planning and scheduling: mathematical basis. Operations Research, 9, 296-320.

Kelley, J. E. and Walker, M. R. (1959). Critical-Path Planning and Scheduling: An Introduction. Ambler, PA: Mauchly Associates, Inc.

Kleindorfer, G. B. (1971). Bounding distributions for stochastic acyclic networks. Operations Research, 19, 1586-1601.

Kulkarni, V. G. & Adlakha, V. G. (September-October, 1986). Markov and Markov-regenerative PERT Networks. Operations Research, 34(5), 769-781.

Kwerel, S. M. (1983). Frechet bounds. In Encyclopedia of Statistical Sciences (Vol. 3, pp. 202-209). New York: John Wiley & Sons, Inc.

Lamberson L. R. (1970). Optimum time compression in project scheduling. Management Science, 16(10), B571 - B606.

Liebman J., Lasdon L., Schrage L. and Waren A. (1986). Modeling and Optimization with GINO. Palo Alto, CA: The Scientific Press.

Lindsey, J. H. (1972). An estimate of expected critical-path length in PERT networks. Operations Research, 20, 800-812.

Littlefield, T. K. & Randolph, P. H. (1987). Another note on PERT times. Management Science, 33, 1357-1359.

MacCrimmon, K. R. & Ryavec, C. A. (1964). An analytical Study of the PERT assumptions. Operations Research, 12, 16-37.

Malcolm, D. G., Rosenbloom, J. H., Clark, C. E., & Fazar, W. (1959). Application of a technique for research and development program evaluation. Operations Research, 7, 646-669.

Martin, J. J. (1965). Distribution of the time through a directed, acyclic network. Operations Research, 13, 46-66.

McCann, M. J. (1986). An Application of convex programming to construction scheduling, (Doctoral dissertation, Iowa State University, 1986). Dissertation Abstracts International, 47(04B), 1672.

Miller, R. W. (1963). Schedule, Cost, and Profit Control with PERT: A Comprehensive Guide for Program Management. New York: McGraw-Hill Book Company, Inc.

Moder, J. J., Phillips, C. R., & Davis, E. W. (1983) Project Management with CPM, PERT and Precedence Diagramming (3rd ed.). New York: Van Nostrand Reinhold Company Inc.

Neil, J. M. (1982). Construction Cost Estimating for Project Control. Englewood Cliffs, NJ: Prentice-Hall, Inc..

Ord, J. K. (November, 1991). A simple approximation to the completion time distribution for a PERT network. Journal of the Operational Research Society, 42, 1011-1017.

Pearson, E. S. & Tukey, J. W. (1965). Approximate means and standard deviations based on distances between percentage points of frequency curves. Biometrika, 52, 533-546.

Perry, C. & Greig, I. D. (1975). Estimating the mean and variance of subjective distributions in PERT and decision analysis. Management Science, 21(12), 1477-1480.

Pritsker, A. A. B. (1986). Introduction to Simulation and SLAM II (3rd ed.). West Lafayette, IN: Systems Publishing Corporation.

Pritsker, A. A. B. (1995). Introduction to Simulation and SLAM II (4th ed.). New York: John Wiley & Sons.

Pritsker, A. A. B. & Kiviat, P. J. (1969). Simulation with GASP II: A FORTRAN Based Simulation Language. Englewood Cliffs, NJ: Prentice-Hall, Inc.

Remer, D. S., Stokdyk, S. B. & Van Driel, M. (August, 1993). Survey of project evaluation techniques currently used in industry. International Journal of Production Economics, 32, 103-115.

Retherford, R. D. & Choe, M. K. (1993). Statistical Models for Causal Analysis. New York: John Wiley & Sons, Inc.

Reuter, V. G. (1979). Using Graphic Management Tools. Journal of Systems Management, 30(4), 6-17.

Richter, I. and Mitchell, R. S. (1982). Hand Book of Construction Law and Claims. Reston, VA: Reston Publishing Company..

Ringer, L. J. (1969). Numerical operators for statistical PERT critical path analysis. Management Science, 16(2), B136-B143.

Robillard, P. & Trahan, M. (1977). The completion time of PERT networks. Operations Research, 25, 15-29.

Ross, S. M. (1980). Introduction to Probability Models (2nd ed.). New York: Academic Press, Inc.

Santner, T. J. & Duffy, D. E. (1989). The Statistical Analysis of Discrete Data. New York: Springer-Verlag.

Sasieni, M. W. (1986). A note on PERT times. Management Science, 32, 1652-1653.

Sculli, D. (February, 1983). The completion time of PERT networks. Journal of the Operational Research Society, 34, 155-158.

Sculli, D. (1989). A historical note on PERT times. OMEGA, 7, 195-196.

Sculli, D. & Shum, Y. W. (1991). An approximate solution to the PERT problem. Computers & Mathematics with Applications, 21(8), 1-7.

Sculli, D. & Wong, K. L. (1985). The maximum and sum of two beta variables and the analysis of PERT networks. Omega, 13, 233-240.

Selvidge, J. E. (1980). Assessing the extremes of probability distributions by the fractile method. Decision Science, 11, 493-502.

Shewhart, W. A, (1980). Economic Control of Quality of Manufactured Product. Milwaukee, WI: ASQC Quality Press. (Original work published 1931, New York: D. Van Nostrand Company, Inc.)

Shogan, A. W. (1977). Bounding distributions for a stochastic PERT network. Networks, 7, 359-381.

Sigal, C. E., Pritsker, A. B. & Solberg, J. J. (1979). The use of cutsets in Monte Carlo analysis of stochastic networks. Mathematics & Computers in Simulation, 21, 379-384.

Sigal, C. E., Pritsker, A. A. B., & Solberg, J. J. (1980). The Stochastic Shortest Route Problem. Operations Research, 28(5), 1122-1129.

Siotani, M., Hayakawa, T., & Fujikoshi, Y. (1985). Modern Multivariate Statistical Analysis: A Graduate Course and Handbook. Columbus, OH: American Sciences Press, Inc.

Smith, J. (1989). Learning Curve for Cost Control. Norcross, GA: Industrial Engineering and Management Press.

Snedecor, G. W. & Cochran, W. G. (1989). Statistical Methods (8th ed.). Ames, IA: Iowa State University Press.

Taylor, H. M. & Karlin, S. (1984). An Introduction to Stochastic Modeling. San Diego, CA: Academic Press, Inc.

Van Slyke, R. M. (1963). Monte Carlo methods and the PERT problem. Operations Research, 11(5), 839-860.

Vardeman, S. B. (1994). Statistics for Engineering Problem Solving. Boston, MA: PWS Publishing Company.

Wagner, H. M. & Whitin, T. M. (1958). Dynamic version of the economic lot size model. Management Science, 5(1), 89-96.

Walpole, R. E. & Myers, R. H. (1978). Probability and Statistics for Engineers and Scientists (2nd ed.). New York: Macmillan Publishing Co., Inc.

Williams, T. M. (April, 1992). Criticality in stochastic networks. Journal of the Operational Research Society, 43, 353-357.