Arthur Nádas

Probabilistic PERT

A solution is offered to the problem of determining a probability distribution for the length of the longest path from source (start) to sink (finish) in an arbitrary PERT network (directed acyclic graph), as well as determining associated probabilities that the various paths are critical ("bottleneck probabilities"). It is assumed that the durations of delays encountered at a node are random variables having known but arbitrary probability distributions with finite expected values. The solution offered is, in a certain sense, a worst-case bound over all possible joint distributions of delays for given marginal distributions for delays. This research was motivated by the engineering problem of the timing analysis of computer hardware logic block graphs where randomness in circuit delay is associated with manufacturing variations. The probability distribution of the critical pathlength turns out to be a solution of an unconstrained minimization problem, which can be recast as a convex programming problem with linear constraints. The probability that a given path is critical turns out to be the Lagrange multiplier associated with the constraint determined by the path. The discrete version of the problem can be solved numerically by means of various parametric linear programming formulations, in particular by one which is efficiently solved by Fulkerson's network flow algorithm for project cost curves.

Introduction

This paper brings together recent results in probability with older techniques in mathematical optimization in order to offer a practical solution to the probabilistic PERT (Program Evaluation and Review Technique) problem. The probabilistic results are due to Meilijson and Nádas [1] and the required mathematical optimization is based on suggestions by Johnson and Wolfe [2]. This research was motivated by the following model for the timing analysis of computer hardware logic block graphs: Each block plays the role of a delay node, as can each "wire" of sufficient length to contribute significant delay. The quantity of interest here is not delay; rather, it is the excess delay (lack or surplus) of signal arrival time at storage elements gated by a clock (*i.e.*, a late signal). The clock signal is also propagated through a network of blocks; consequently, the appropriate PERT model is constructed by replacing each delay with its negative for every delay random variable associated with propagating a clock signal. An early signal model is constructed in an analogous way; in this case one is concerned with shortest paths.

The paper is organized as follows. The general probabilistic and graph-theoretic features of the problem are discussed in the next section. The following section introduces, still in general terms, the proposed probabilistic solution together with Fulkerson's "Project cost curve" problem (Fulkerson [3]) whose solution, unexpectedly, also solves the optimization problem for the discrete probabilistic PERT. The next two sections describe the probabilistic solution. The concluding section describes the discrete optimization problem, shows that Fulkerson's algorithm is applicable, and reports some modest numerical experience.

The problem

PERT or CPM (Critical Path Method) networks are directed acyclic graphs having a single source node (start) and a single sink node (finish). In this paper the nodes are regarded as the places where tasks or jobs are accomplished at the cost of some delay. The directed arcs describe the precedence relations among the tasks (nodes); it is assumed that a task begins at the precise time when all preceding tasks have been completed. The completion time at a node is then simply the longest pathlength (*i.e.*, sum of delays on the path) of all paths leading to and including the node. In particular, the critical pathlength is

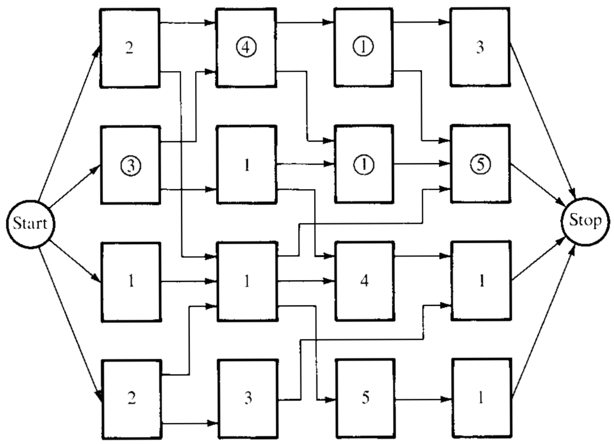

Figure 1 A PERT problem. The critical pathlength is 13; it is attained on the two paths shown. Numbers in boxes (nodes) denote delay.

the completion time at the sink node. Figure 1 illustrates this model. The literature on PERT and CPM is enormous and no attempt is made here to summarize it.

Suppose that the nodes are numbered in any order as $1, 2, 3, \cdots, n$, and that the delay incurred at node i is a random variable X_i (random variables will be denoted by capital letters) whose distribution function $F_i(x)$ is known; *i.e.*, for all x

$$F_i(x) = \Pr(X_i \le x) \qquad i = 1, \cdots, n$$

is a family of n given probability distribution functions. The F_i are the marginal distributions of the delays X_i at node i; these functions together with a specification of the PERT graph are the data of the model. Models of this nature may arise in various ways. These include the case where randomness of delay is due to an "explainable" random mechanism such as random physical phenomena, random events in manufacture, etc., and the case where randomness is introduced as a tool for describing uncertainty about actual nonrandom delays. Figure 2 illustrates this model.

The jth path of the k paths that run from source to sink is denoted by P_j, the list of those node numbers that belong to the jth path. Evidently the random completion time T_j of the jth path is

$$T_j = \sum_{P_j} X_i \qquad j = 1, 2, \cdots, k.$$

The random completion time M at the sink node is then the completion time of the network; *i.e.*, the critical pathlength is

$$M = \max_j T_j = \max_j \sum_{P_j} X_i.$$

In addition to finding the distribution of M, or finding for every t

$$\Pr(M > t),$$

it is of interest to find out which paths (and nodes) are critical in some sense; for example, by using

$$\lambda_j(t) = \Pr(M = T_j > t)$$

for some interesting t, such as a "statistical worse case" value, t_α, for which $\Pr(M > t_\alpha) = \alpha$. (For design purposes one is interested in the identity of the worst path, *provided it is too long*.)

It is evident that the probability distribution of M cannot be determined without knowledge of the joint distribution of the delays $(X_1, \cdots, X_n)$. The marginal distributions $F_1, \cdots, F_n$ do not carry enough information to do that. It is also clear that the distribution of M is an extremely complicated functional of the joint distribution except in the case of very special simple graphs. The only simple nondeterministic example (involving a general graph) known to the author is the "perfect tracking" model, $X_i = \mu_i + \sigma_i Z$ for given constants μ_i, σ_i $(\sigma_i > 0)$ with $i = 1, \cdots, n$ and a random variable Z with known distribution function F. In this case, letting m_j, s_j denote the path sums of μ_i, σ_i for $j = 1, 2, \cdots, k$, that is,

$$m_j = \sum_{P_j} \mu_i, \qquad s_j = \sum_{P_j} \sigma_i,$$

one finds immediately that for all t,

$$\Pr(M > t) = 1 - F[z(t)],$$

where

$$z(t) = \min_j \frac{t - m_j}{s_j}.$$

Aside from such simple models, the exact distribution of M appears intractable no matter what sort of dependence structure is assumed. It is interesting to observe that even if the delays $\{X_i\}$ were assumed to be mutually independent random variables (an untenable assumption in many important problems), the joint distribution of the path sums $\{T_j\}$, *a fortiori* of M, will still be intractable. This is due to the dependence introduced by the multiple assignment of nodes to paths. For the same reason, the problem of finding bottleneck probabilities, *i.e.*, paths that are likely to be critical, also appears intractable.

The assumption of mutually independent random delays has been considered by several authors. Sielken and Hartley and colleagues ([4] and its references) make the implicit additional assumption that the graph can be usefully represented as a certain "supergraph" of subgraphs. It is assumed that 1) each subgraph is of sufficiently simple structure (parallel, series, Wheatstone bridge) so that the critical path through the subgraph will have a simply computed probability distribution, and 2) the supergraph has the very simple structure of having all its nodes (subgraphs of original graph) in parallel and series combinations. This approach is valid for graphs satisfying both 1 and 2. Another technique to exploit the supposed simplicity of the mutual independence assumption was suggested by Robillard and Trahan [5]. This technique consists of first computing (by convolution) each of the k marginal distributions of the path completion times $T_1, \cdots, T_k$ and then using these to obtain a bound on $\Pr(M > t)$. The technique works best on graphs having disjoint parallel paths but can be unduly pessimistic in general (see the discussion in Meilijson and Nádas [1]). Also based on independence assumptions, in earlier work Fulkerson [6] has obtained some lower bounds on the expected length of critical paths.

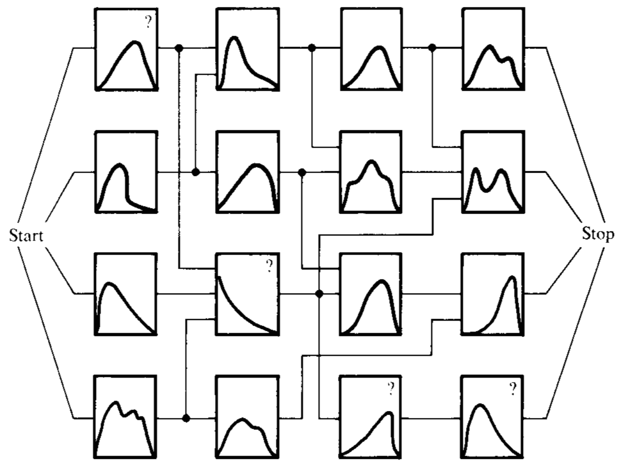

Figure 2 A probabilistic PERT problem. The flow shown, assumed left-to-right, is the same as in Fig. 1. Here, any path may be critical. Shapes in boxes refer to probability distributions of delay.

The proposed solution

In the present work, no assumptions are made about the joint distribution of the delays $X_1, \cdots, X_n$ beyond the constraint that the X_i have the correct marginal distributions F_i. Similarly, no assumptions are made regarding the graph beyond the statement that it represents a PERT problem. (Actually, the mathematical result presented below does not even require that the problem be realizable as a directed graph; a path is merely a set of nodes.)

The point of view adopted herein is that 1) the joint distribution is not known and 2) even if 1 is assumed away (by *e.g.*, mutual independence), the problem remains intractable on account of a complicated graph. The solution appropriate to this point of view begins by considering all possible joint distributions for the delays consistent with the given marginal distributions $F_1, F_2, \cdots, F_n$. Joint distributions which make M as large as possible (in a certain sense, discussed in the next section) are then singled out to obtain a sharp (*i.e.*, realizable) upper bound for distribution of M.

In the special case of a simple, purely parallel graph (disjoint paths each having a single delay node plus dummy source and sink), these results have also been obtained by Lai and Robbins [7, 8].

Fulkerson [3] solved the following "project cost curve" problem associated with deterministic PERT networks. The problem is to trade off time against budget. Consider a project which consists of jobs (tasks, activities). The precedence relations among jobs are induced by restrictions inherent in the project, *e.g.*, precedence of successive stages in manufacturing or transportation. Suppose that the cost of delay for any given job varies inversely and linearly with the delay and that the delay may be chosen to be any value between two given extremes, a "normal" and a "crash" task completion time for that job. What is the least project cost (the sum of all job costs) satisfying the constraint that the project completion time t (the critical path length) is specified in advance? For varying t this minimum project cost traces out the project cost curve. Fulkerson's algorithm [3] computes the entire project cost curve in an efficient way using the celebrated network flow approach (see, *e.g.*, [9]). Although this problem bears little resemblance to the probabilistic PERT problem, the numerical solution of the discrete version of probabilistic PERT can be formulated in such a way that Fulkerson's algorithm is applicable. The reason for this is that both the "project cost curve" of deterministic PERT and the "expected optimal residual completion time" of discrete probabilistic PERT are decreasing, convex, piecewise linear functions of time. This relationship was first recognized by E. L. Johnson [2] and we shall describe it in the section on computation.

Comparison of random variables

The distribution of the random critical pathlength will vary with the possible joint distributions of the delays $(X_1, X_2, \cdots, X_n)$ even though the X_i have the given marginal distribution functions F_i. For example, if M_1 denotes the M corresponding to one joint distribution and M_2 denotes the M corresponding to another joint distribution, then M_1 and M_2 are different random variables (r.v.'s) because they have different distribution functions (d.f.'s).

If X, Y are two r.v.'s, it is quite possible that $X \le Y$ in one instance by $X > Y$ in another. One might consider

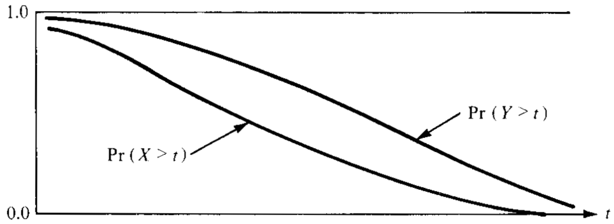

Figure 3 Stochastic inequality, $X \leq Y$ (stochastic).

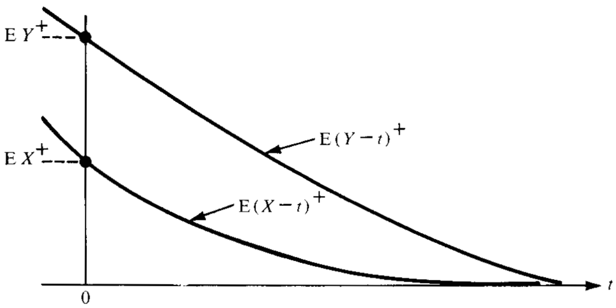

Figure 4 Convex inequality, $X \leq Y$ (convex).

various definitions for saying, "$X \leq Y$ on the average," such as $\mathrm{E}\, X \leq \mathrm{E}\, Y$ ($\mathrm{E}\,(\cdot)$ denotes the mathematical expectation of its argument) or such as $\Pr\,(X \leq Y) > (1/2)$ and so forth. Average comparisons, however, tell us little about the tails of the distribution of $X - Y$ and it is necessary to adopt somewhat more complicated methods of comparison. A usual tool for this is the *stochastic inequality*

$X \leq Y$ (stochastic)

if and only if

$\Pr\,(X > t) \leq \Pr\,(Y > t)$

for all t so that the two curves never cross as in Fig. 3.

Such a stochastic comparison would solve the probabilistic PERT problem if we could find a joint distribution such that the corresponding random critical path length M would be stochastically as large as possible while observing the constraint that the X_i have the prescribed distributions F_i. In the simple problem where the graph consists of n nodes in parallel, one has

$$M = \max_{1 \leq i \leq n} X_i.$$

In this case, the stochastically worst distribution of M is given by

$$\Pr\,(M > t) = \sum_{i=1}^{n} \Pr\,(X_i > t) = \sum_{i=1}^{n} 1 - F_i(t)$$

for all t where the sum is less than unity; it is unity elsewhere. This result can be proved by induction on n with respect to the construction of joint distributions by repeatedly "opposing" the next F_{i+1} to the distribution of the previously constructed maximum of the previous X_1, $X_2, \cdots, X_i$ ([1], p. 7). Although the joint distribution so constructed is not the only one to yield the above distribution for M, the stochastically worst distribution of M is well defined.

In PERT problems of practical interest, such as our model for the timing analysis of a logic block graph, the graphs are often large and complicated. In these problems there are no simple formulae available for the distribution of M_i; in fact, it is easy to see that in general there is no joint distribution which stochastically maximizes M. [As an example, if X_1, X_2 both have the uniform distribution $(1/3, 1/3, 1/3)$ on $\{1, 2, 3\}$ and $M = X_1 + X_2$, then $\Pr\,(M \geq 6) = 1/3$ is maximal but forces $\Pr\,(M \geq 5) = 1/3$ also; while $\Pr\,(M \geq 6) = 0$ allows $\Pr\,(M \geq 5) = 2/3$.] Moreover, it seems exceedingly difficult to find good bounds for $\Pr\,(M > t)$ even for a single value t. The solution proposed in the next section is based on the notion of *convex inequality*, introduced in Meilijson and Nádas [1], and defined as follows. For r.v.'s X, Y with $\mathrm{E}|X|$ and $\mathrm{E}|Y|$ finite,

$X \leq Y$ (convex)

if and only if

$$\int_t^\infty \Pr\,(X > u)du \leq \int_t^\infty \Pr\,(Y > u)du$$

for all t. This amounts to comparing the "residual expectations" of the r.v.'s X and Y:

$$\mathrm{E}\,(X - t)^+ \leq \mathrm{E}\,(Y - t)^+ \qquad -\infty < t < \infty,$$

as is seen from the identity

$$\mathrm{E}\,(X - t)^+ = \int_t^\infty \Pr\,(X > u)du,$$

where α^+ denotes $\max\,\{\alpha, 0\}$. (See Fig. 4.)

It is well known that $X \leq Y$ (stochastic) if and only if for every "test function" g ($g \geq 0$),

$\mathrm{E}\, g(X) \leq \mathrm{E}\, g(Y)$,

provided that g is increasing. It turns out that in an analogous way

$X \leq Y$ (convex)

if and only if $\mathrm{E}\, g(X) \leq \mathrm{E}\, g(Y)$ for all increasing and *convex* ($g'' > 0$ if there are derivatives) test functions. Obviously a stochastic inequality implies a convex one, but the converse is not true. This characterization indicates that in a certain sense, the convex inequality is the "next best thing'' to stochastic in situations (such as PERT) where stochastic comparison is not possible.

Perhaps the simplest measure of dependence between two r.v.'s is the correlation coefficient. It is not hard to show that if X and Y have given marginal distributions, the correlation coefficient is maximized by the same joint distribution which convexly minimizes the difference $|X - Y|$, namely the "lineup" distribution. This is defined constructively as the joint distribution of the pair (X, Y) when it is generated by a *simultaneous Monte Carlo* procedure, as in Fig. 5. In a similar way, the correlation coefficient is minimized by the same joint distribution which convexly maximizes $|X - Y|$, namely the "opposed" distribution, Fig. 6. To see this, note that it suffices to bound E XY. For simplicity let $X, Y \geq 0$ so that

$$XY = \int_0^\infty \int_0^\infty I_{XY}(x, y)dxdy,$$

where $I_{XY}(x, y)$ is 1 if $0 \leq x \leq X$ and $0 \leq y \leq Y$ and it is zero otherwise. Then

$$\mathrm{E}\ XY = \mathrm{E}\ I_{XY} = \int_0^\infty \int_0^\infty \Pr\ (X > x,\ Y > y)dxdy,$$

and E XY (hence the correlation coefficient) is minimized (maximized) by bounding the integrand. This in turn is accomplished by joint distributions constructed by the "opposition" ("lineup") of the two distributions. For example, the maximum of $\Pr\ (X > x, Y > y)$ is $\Pr\ (X > x) + \Pr\ (Y > y)$ or unity. Of course, X, Y must have finite variances to have the correlation coefficient defined. This characterization of extremal correlations indicates that this convex method of comparing r.v.'s is well suited to the problem of maximizing over possible joint distributions of delays in a PERT network.

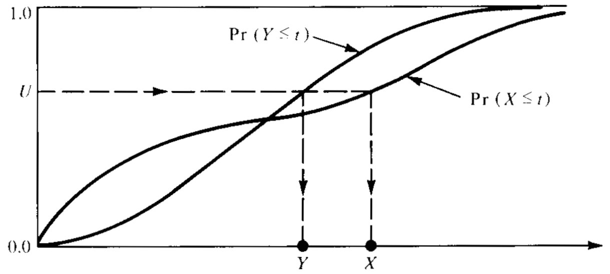

Figure 5 The "lineup" algorithm. U has the rectangular distribution. (X, Y) has the lineup distribution.

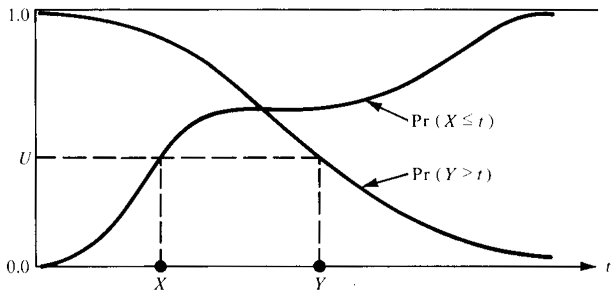

Figure 6 The opposition algorithm. U has the rectangular distribution. (X, Y) has the "opposed" distribution.

Mathematical solution

The announced solution to the probabilistic PERT program can now be stated as follows.

Theorem Assume all delays X_i have E$|X_i|$ finite. Then

1. There exists a r.v. M^* such that

$$\mathrm{E}\ (M^* - t)^+ = \min_{\{x_i\}} \left(\max_j \sum_{P_j} x_i - t \right)^+ + \sum_{i=1}^{n} \mathrm{E}\ (X_i - x_i)^+$$

i.e., the right-hand side is indeed the integral of some distribution's right tail.

2. If M is the critical pathlength corresponding to an arbitrary joint distribution having the correct marginal distributors, then M is convexly no larger than M^*, *i.e.*, M^* is a convex bound for all possible M.

3. This bound is sharp: For every t there is a joint distribution having the correct marginals such that E $(M - t)^+ =$ E $(M^* - t)^+$. In other words E $(M^* - t)$ is, for varying t, the convex envelope of all possible residual expectation functions.

4. If it happens that the graph consists only of parallel and series structures, then $M^* = M$ for some joint distribution, *i.e.*, the envelope in one of the residual expectations.

5. The solution can be recast as the following constrained minimization problem:

$$\mathrm{E}\ (M^* - t)^+ = \min_{\{x_i\}} \sum_{i=1}^{n} \mathrm{E}\ (X_i - x_i)^+$$

subject to

$$\sum_{P_j} x_i \leq t \qquad \text{for all } j.$$

6. Let λ_j denote the Lagrange multiplier associated with the jth path constraint in the above. Then for M defined in 3

$$\lambda_j = \Pr \left(M = \sum_{P_j} X_i > t \right) \qquad j = 1, \cdots, n$$

is the probability that the jth path is critical, *i.e.*, the jth bottleneck probability.

The idea for this theorem came from observing that in the purely parallel case of $M = \max X_i$ one has (say $X_i \geq 0$ for all i)

$$\mathrm{E}\, M = \int_0^\infty \Pr\,(M > t)dt = t_0 + \sum_{i=1}^{n} \mathrm{E}\,(X_i - t_0)^+$$

$$= \min_{0<t}\, t + \sum_{i=1}^{n} \mathrm{E}\,(X_i - t)^+,$$

where t_0 is the least upper bound of values t for which

$$\sum_{i=1}^{n} \Pr\,(X_i > t) = 1.$$

The following simple example illustrates the result in the parallel-series situation where no numerical optimization is required. Suppose no delays at source or sink nodes and assume a graph on four other nodes $\{1, 2, 3, 4\}$ such that nodes 1 and 2 each precede both nodes 3 and 4. Then there are four paths $(1, 3)$, $(1, 4)$, $(2, 3)$, $(2, 4)$ from source to sink. Here M can be expressed as $\max\{X_1, X_2\} + \max\{X_3, X_4\}$. Suppose all X_i have normal distributions with variance one, X_1 and X_4 have mean zero while X_2 and X_3 have mean one. Let Φ denote the standard normal integral. Then the convex (and stochastic) bound on $\Pr\,(\max\{X_1, X_2\} > t)$ is $\min\{1, \Phi(-t) + \Phi(1 - t)\}$. Since $\max\{X_3, X_4\}$ has the same bound, lining these up yields the convex bound for M: $\Pr\,(M^* > t) = \min\{1, \Phi(-t/2) + \Phi(1 - t/2)\}$. Observe that if one were to erroneously first compute convex bounds on the separate paths and then find the best stochastic bound for these path bounds, the resulting bound $\Pr\,(M > t) \leq \min\{1, \Phi(-t/2) + \Phi(1 - t/2) + 2\Phi((1 - t)/2)\}$ is larger than the best convex bound given by the theorem. The theorem was proved by showing that the solution is indeed a convex bound and then constructing a joint distribution which, for a given t, attains that bound ([1], p. 6).

Computational algorithms

At first glance the mathematical result seems impossible to compute for large n (say $n > 50$) since the minimization required is n-dimensional. However, the objective function, as described in 5. in the previous section, has certain special features which, together with the special nature of the constraints, allow a practical solution. Note first that since each variable x_i appears alone as the argument of the ith summed function, the sum is *separable*. Next observe that the objective function is also *convex* because each summand is convex (integrals of decreasing functions are automatically convex). The constraints are all *path-sum* (linear) *constraints* on a PERT network. These are the three special features to be exploited.

There are several distinct linear programming formulations available for the separable convex objective with linear constraints. Using convexity, each $\mathrm{E}\,(X_i - x)^+$ can be approximated by a piecewise linear majorant. This linearization of the residual expectation is equivalent to approximating the given distribution F_i by another distribution that concentrates all the probability at the breakpoints of the majorant. Since this is a minimization problem, the majorization of the objective function ensures that the approximate solution is still a correct convex bound. Observe that in a linear programming formulation, the Lagrange mulipliers (bottleneck probabilities) are obtained directly as dual variables in the problem.

Let $\phi_i(x)$ denote the piecewise linear majorant of $\mathrm{E}\,(X_i - x)^+$ and suppose that the breakpoints of ϕ_i are equally spaced as $x_{i0} + \Delta_i, \cdots, x_{i0} + m\Delta_i$ over the support $(x_{i0}, x_{i0} + m\Delta_i)$ of ϕ_i. Then with

$$L_{i\nu}(x) = A_{i\nu}x + B_{i\nu} \qquad \nu = 1, \cdots, m,$$

one has for $i = 1, \cdots, n$

$$\phi_i(x) = L_{i\nu}(x) \qquad x_0 + (\nu - 1)\Delta_i < x \leq x_0 + \nu\Delta_i.$$

Our task is to minimize

$$\phi_1(x_1) + \phi_2(x_2) + \cdots + \phi_n(x_n)$$

by choice of $x = (x_1, \cdots, x_n)$ subject to

$$\sum_{P_j} x_i \leq t \qquad j = 1, 2, \cdots, k.$$

The first linear program considered was based on the well known "delta method" for convex objective functions. Partition each x_i as $x_i = x_{i1} + x_{i2} + \cdots + x_{im}$ with the constraints $0 \leq x_{i\nu} \leq \Delta_i$ so that

$$\sum_{i=1}^{n} \phi_i(x_i) = \sum_{i=1}^{n} \sum_{\nu=1}^{m} A_{i\nu}x_{i\nu} + \mathit{constant}.$$

This defines the linear program

$$\max\, \mathbf{cx} \text{ subject to } \mathbf{Ax} \leq \mathbf{b}, \qquad \mathbf{x} \geq \mathbf{0},$$

where

$$\mathbf{c} = (-A_{11}, \cdots, -A_{nm}),$$

$$\mathbf{b} = (t, \cdots, t, \Delta_1, \cdots, \Delta_1, \cdots, \Delta_n, \cdots, \Delta_n),$$

and where $\mathbf{A}$ is a $k + mn$ by mn matrix of zeros and ones, k rows for the path constraints and mn rows for the Δ_i constraints. If the bounding technique (available in some linear programming computer programs) is used, then $\mathbf{A}$ is k by mn. If this problem is solved in the dual space, then the bounding technique cannot be used but one may perform column generation via any longest path algorithm.

The second linear program considered was based on an idea of Hoffman [10]. Do not partition the x_i; rather, introduce new variables $z_1, \cdots, z_n$ and find $z \geq 0$, $x \geq 0$ to minimize

$$\sum_{i=1}^{n} 1 \cdot z_i + \sum_{i=1}^{n} 0 \cdot x_i = \sum_{i=1}^{n} z_i$$

subject to

$$z_i \geq A_{i\nu} x_i + B_{i\nu} \qquad \text{all } i, \nu$$

and

$$\sum_{P_j} x_i \leq t \qquad \text{all } j.$$

This works because $z_i \geq \phi_i(x_i)$ if and only if $z_i \geq L_{i\nu}(x_i)$ for all $\nu = 1, \cdots, m$. The matrix form of the primal problem is

min $\mathbf{cx}$

subject to

$$\mathbf{x} \geq \mathbf{0}, \qquad \mathbf{Ax} \geq \mathbf{b}$$

where

$$\mathbf{c} = (1, 1, \cdots, 1, 0, 0, \cdots, 0),$$

$$\mathbf{b} = (B_{11}, B_{12}, \cdots, B_{nm}, -t, \cdots, -t)$$

and where $\mathbf{A}$ is $k + mn$ by $2n$.

Notice that in this formulation the size of **A** grows only linearly with m and that in the dual formulation the number of rows is $2n << k + mn$ for a graph of the expected complexity, a desirable shape for **A**.

These solutions appear to require a new minimization computation for each desired timepoint t. Considerable reduction in computing is achieved by using a *parametric linear programming* algorithm which exploits the fact that the same problem is repeatedly re-solved for a systematically varying t.

The most convenient and, to the author's knowledge, most efficient formulation of this problem is a linear program which can be computed by (a variant of) Fulkerson's [3] network flow algorithm for project cost curves. To describe this, following Fulkerson we switch to the conventional description of PERT where delays are associated with the arcs of a directed graph and where a node is interpreted as the event that all jobs on arcs preceding it have been completed. Let t_{ij} denote the delay on arc (i, j) and let t_j denote the event time at the jth node. Each arc is associated with three nonnegative integers $a_{ij} \leq b_{ij}$ and c_{ij} with the interpretation that $a_{ij} \leq t_{ij} \leq b_{ij}$ for all (i, j) and that the cost of doing job (i, j) in time t_{ij} is $k_{ij} - c_{ij}t_{ij}$ where k_{ij} are given constants. If the source and sink nodes are labeled 1 and n, we may assume $t_1 = 0$ so that the project completion time is t_n. The problem is to minimize

$$\sum_i \sum_j k_{ij} - c_{ij} t_{ij}$$

by choice of $\{t_{ij}\}$ subject to $t_n \leq t$ and

$$a_{ij} \leq t_{ij} \leq b_{ij} \qquad \text{all } i, j,$$

where t is the desired project completion time. This is a linear program with a lot of redundant variables. The reason for formulating the problem this way is that in the dual space this linear programming problem is solved more efficiently ([3], p. 172) by a network flow algorithm than by a parametric linear programming algorithm. The network flow algorithm generates the project cost curve as follows. The algorithm begins with the largest t of interest, namely $t = t_n$ for $t_{ij} = b_{ij}$, all i, j. It then computes sequentially all the breakpoints of the curve and these turn out to be the optimal event times t_j. The optimal job times are then obtained simply as $t_{ij} = \min\{b_{ij}, t_j - t_i\}$. The algorithm is described and illustrated in [3, 11], and the more recent book of Lawler [12].

It remains to re-cast the discrete probabilistic PERT problem, as promised, in the form of the above budget *versus* time tradeoff problem. This is done as follows. Consider the partition of delay used in the "delta method,"

$$x_i = \sum_{\nu=1}^{m} x_{i\nu},$$

and introduce m delays in series in place of each original delay. Denoting the larger set of $\{x_{i\nu}\}$ as $\{x^*_l\}$, identify an arc delay t_{ij} of the new graph with each node delay x^*_l of the old graph. Put

$$a_{ij} = x_{k0} + (\nu - 1)\Delta_k,$$

$$b_{ij} = a_{ij} + \Delta_k,$$

$$c_{ij} = A_{k\nu},$$

if $x^*_l = x_{k\nu}$. It is then clear that, except for a constant, the cost objective of the project cost curve problem coincides with the residual expectation objective of the discrete probabilistic PERT problem. A variation: Fulkerson's algorithm can be modified to handle multiple arcs (instead of multiple nodes); this has not been tried on the present problem.

Not only is the network flow algorithm an efficient way to compute $E(M^* - t)^+$ for all t, but by a fortunate happenstance, in terms of the optimal residual expectation, the network flow algorithm proceeds sequentially, from computing $E(M^* - t)$ for a largest t [$E(M^* - t)^+$ is zero at this point] and then computing successively smaller "breakpoints" and corresponding values of $E(M - t)^+$. The exact solution is the piecewise linear convex function defined by these points. Thus if one is interested in a *sta-*

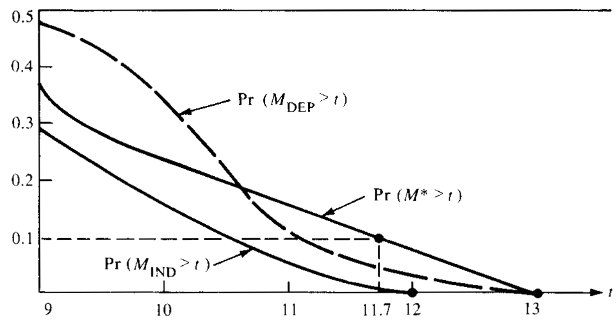

Figure 7 Distributions of critical pathlength.

tistical worst case for the critical pathlength, one need only continue the computation until this point is bracketed by the last two computed breakpoints, *i.e.*, until the first breakpoint t_B for which Pr $(M^* > t_B) \geq 0.1$, for example.

These ideas are illustrated in Fig. 7. The graph used was that shown in Fig. 1. The marginal distributions used were all uniform (rectangular) on the interval from 0 to the delay shown in Fig. 1. Strictly speaking, the graph of Pr $(M^* > t)$ so obtained is a step function with jumps at the breakpoints. However we can and do revert to the given continuous distribution functions at this stage of the calculation, the only purpose of using a discrete function having been to facilitate computation. For this example, each E $(X_i - x)^+$ was majorized by a 3-segment polygon, the breakpoints in Fig. 7 occurring at 13, 12, and 9. There is no need to continue any further.

The random critical pathlengths M_{IND} and M_{DEP}, whose distributions are shown in Fig. 7, refer to joint distributions which refer to complete mutual independence and a strong "big together, small together" sort of dependence (high positive correlation) among the $\{X_i\}$ respectively. It can be seen that the sharp convex bound is not a stochastic bound but that it does dominate in the far right tail. In this example the "statistical worst case" corresponds to a probability of $\alpha = 0.1$, and α is arbitrarily chosen. This example was repeatedly solved for various fixed t using the first two linear programming formulations described above and a package of APL programs by Crowder and Wolfe [13]. For any fixed t tried, the dual form of Hoffman's formulation gave the fastest computation. This example was also solved by the network flow algorithm for project cost curves using an APL program developed by Schmidt [14]. Schmidt's program was also exercised on a more interesting example, to wit, the timing analysis of a logic chip. The graph in this case had 484 nodes, 1681 edges and 4217 paths from source to sink. Schmidt's program, running under CMS on an IBM 370/168 with an attached processor, consumed approximately seven CPU seconds for each breakpoint computed.

Suggestions for future work

- *Probability*

In some problems max Pr $(M > t)$ for a single t may be of major interest. This problem is unsolved. Results giving bounds on $|\Pr(M > t) - \Pr(M^* > t)|$ or on $|\mathrm{E}\, M - M^*|$ would be useful.

- *Optimization*

The number of breakpoints needed depends not only on the chosen probability level α but also on the number of breakpoints used in approximating the functions E $(X_i - x)^+$. A numerical study of the 16-node example showed that there was little improvement in computing E $(M^* - t)^+$ when using more than three breakpoints (plus two endpoints) per node. It is clear, however, that a systematic way of choosing the degree of approximation is needed.

Acknowledgments

The author's greatest debt is to I. Meilijson, whose mathematical skill, spirit and energy were indispensable during the work that led to [1]. The author is also indebted to several colleagues at IBM for many useful conversations and other help. P. S. Wolfe was a patient teacher and consultant; he was also the first to solve our small example using a direct optimization technique as well as linear programming. E. L. Johnson made a significant contribution to the efficiency of the computational solution by recognizing that the discrete version of the minimization problem can be related to Fulkerson's project cost curve problem. H. Crowder and M. Held consulted on optimization. A. J. Hoffman gave an alternate linear programming formulation. T. R. Puzak and S. Schmidt both contributed algorithms and computer programs at various stages of this work. A. G. Konheim proposed this problem and was first to analyse our small example by Monte Carlo simulation; the M_{IND} and M_{DEP} curves in Fig. 7 are taken from his work.

References

1. I. Meilijson and A. Nádas, "Convex Majorization," *Research Report RC6762*, IBM Thomas J. Watson Research Center, Yorktown Heights, NY, 1977. To appear in *J. Appl. Prob.*
2. E. L. Johnson and P. S. Wolfe, IBM Thomas J. Watson Research Center, Yorktown Heights, NY, private communication.
3. D. R. Fulkerson, "A Network Flow Computation for Project Cost Curves," *Manage. Sci.* **7**, 167–178 (1961).

4. R. L. Sielken and H. O. Hartley, "A New Statistical Approach to Project Scheduling," *Technical Report THEMIS-TR-56*, Institute of Statistics, Texas A&M University, College Station, TX, 1977.
5. P. Robillard and M. Trahan, "The Completion Time of PERT Networks," *Oper. Res.* **25,** 15-29 (1977).
6. D. R. Fulkerson, "Expected Critical Path Lengths in PERT Networks," *Oper. Res.* **10,** 808-817 (1962).
7. T. L. Lai and H. E. Robbins, "Maximally Dependent Random Variables," *Proc. Nat. Acad. Sci. USA* **73,** 286-288 (1976).
8. T. L. Lai and H. E. Robbins, "A Class of Dependent Random Variables and Their Maxima," to appear.
9. R. L. Ford and D. R. Fulkerson, *Flows in Networks*, Princeton University Press, Princeton, NJ, 1962.
10. A. J. Hoffman, IBM Thomas J. Watson Research Center, Yorktown Heights, NY, private communication.
11. D. R. Fulkerson, "Scheduling in Project Networks," *Proc. IBM Scientific Computing Symposium*, IBM Thomas J. Watson Research Center, Yorktown Heights, NY, March 16-18, 1964, pp. 73-92.
12. E. L. Lawler, *Combinatorial Optimization*, Holt, Rinehart, & Winston, New York, 1976.
13. H. Crowder and P. Wolfe, "VSLPS: A Users Guide to Solving Linear Programming Problems in APL," to appear.
14. S. Schmidt, "Efficient Minimum Cost Flow Implementation in APL," to appear.

Received October 30, 1978; revised December 22, 1978

The author is located at the IBM Data Systems Division laboratory, East Fishkill (Hopewell Junction), New York 12533.