D. Ray Fulkerson and Project Scheduling

Rolf H. Möhring

2010 Mathematics Subject Classification: 90B35, 90B36, 05C21
Keywords and Phrases: Stochastic scheduling, underestimation error, time-cost tradeoff

1 Introduction

D. Ray Fulkerson (1922–1976) made fundamental and lasting contributions to combinatorial mathematics, optimization, and operations research [2]. He is probably best known for his work on network flows and in particular for the famous max flow–min cut theorem, stating that the maximum amount of a flow from a node s to a node t in a directed graph equals the minimum capacity of a cut separating s from t.

Less known is the fact that he also made important contributions to project scheduling. One deals with time-cost tradeoff analysis of project networks, which he solved with min-cost flow techniques. This method has meanwhile entered standard text books such as [1] (often as an exercise of application of flow methods) and will not be discussed here.

The much less known contribution concerns project planning when the individual job times are random variables. Fulkerson was one of the first to

Figure 1: Ray Fulkerson at Robert Bland's wedding

Figure 2: Polaris A-3 at Cape Canaveral (©Wikimedia Commons)

recognize the deficiency of the then state-of-the-art operations research techniques, and he developed a method for better analysis that has started a whole stream of research on risk analysis in project planning.

This chapter tells the story of this contribution.

2 The background [10, 3]

During the Cold War, around the late fifties and early sixties, Lockheed Corporation developed and built the first version of the Polaris missile for the United States Navy as part of the United States arsenal of nuclear weapons. It was a two-stage solid-fuel nuclear-armed submarine-launched intercontinental ballistic missile with a range of 4.600 km that replaced the earlier cruise missile launch systems based on submarines [3].

The complexity of this and similar projects required new planning tools that could deal with research and development programs for which time is an uncertain but critical factor. To support the Polaris project, the Navy's Special Projects Office developed the Program Evaluation and Review Technique (PERT), which still is applied as a decision-making tool in project planning. Willard Fazar, Head of the Program Evaluation Branch of the Special Projects Office [4] recalls:

> The Navy's Special Projects Office, charged with developing the Polaris-Submarine weapon system and the Fleet Ballistic Missile capability, has developed a statistical technique for measuring and

> forecasting progress in research and development programs. This Program Evaluation and Review Technique (code-named PERT) is applied as a decision-making tool designed to save time in achieving end-objectives, and is of particular interest to those engaged in research and development programs for which time is a critical factor.
>
> The new technique takes recognition of three factors that influence successful achievement of research and development program objectives: time, resources, and technical performance specifications. PERT employs time as the variable that reflects planned resource-applications and performance specifications. With units of time as a common denominator, PERT quantifies knowledge about the uncertainties involved in developmental programs requiring effort at the edge of, or beyond, current knowledge of the subject – effort for which little or no previous experience exists.
>
> [...]
>
> The concept of PERT was developed by an operations research team staffed with representatives from the Operations Research Department of Booz, Allen and Hamilton; the Evaluation Office of the Lockheed Missile Systems Division; and the Program Evaluation Branch, Special Projects Office, of the Department of the Navy.

I will explain the main idea underlying PERT in the next section. Fulkerson noticed that PERT makes a systematic error, as it generally underestimates the expected makespan of a project. He worked at the RAND Cooperation at that time and wrote in research memorandum RM-3075-PR prepared for the United States Air Force [6] and later published in slightly revised form in [5]:

> The calculation of project duration times and project cost by means of network models has become increasingly popular within the last few years. These models, which go by such names as PERT (<u>P</u>rogram <u>E</u>valuation <u>R</u>eview <u>T</u>echnique), PEP (<u>P</u>rogram <u>E</u>valuation <u>P</u>rocedure), Critical Path Scheduling, Project Cost Curve Scheduling, and others, have the common feature that uncertainties in job times are either ignored or handled outside the network analysis, usually by replacing each distribution of job times by its expected value.

He continues his criticism of PERT in the follow-up report RM-3075-PR [7]:

> The PERT model of a project usually assumes independent random variables for job times, instead of deterministic times [...]. But the usual practice has been to replace these random variables by their expected values, thereby obtaining a deterministic problem. The solution of this deterministic problem always provides an optimistic estimate of the expected length of the project.

> [...]
> Although the analysis of a PERT model, with fixed job times, is trivial from the mathematical point of view, the model itself appears to be a useful one, judging from its widespread acceptance and use throughout industry today. But it should be added that it is difficult to assess the usefulness of PERT on this basis alone, since the model has been the subject of much hard-sell advertising and exaggerated claims.

Fulkerson instead suggests an algorithm that uses discrete random job times and calculates a much better lower bound on the expected project makespan than the one obtained by the PERT. It was published in 1962 [5] and has become one of the fundamental papers in the area of project risk analysis.

I will outline some of the underlying mathematics of this development in the next section. Part of that exposition is taken from [11].

3 Coping with uncertainty in scheduling: The math

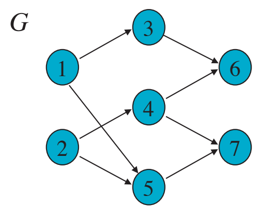
In real-life projects, it usually does not suffice to find good schedules for fixed deterministic processing times, since these times mostly are only rough estimates and subject to unpredictable changes due to unforeseen events such as weather conditions, obstruction of resource usage, delay of jobs and others.

In order to model such influences, PERT assumes that the processing time of a job $j \in V = \{1, \ldots, n\}$ is assumed to be a random variable $\mathbf{p}_j$. Then $\mathbf{p} = (\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n)$ denotes the (random) vector of job processing times, which is distributed according to a joint probability distribution Q. This distribution Q is assumed to be known, though sometimes, also partial information may suffice. In general, Q may contain stochastic dependencies, but most methods require that the job processing times are stochastically independent. (Fulkerson allows some dependencies in his method, see below.))

Jobs are subject to precedence constraints given by a directed acyclic graph $G = (V, E)$. We refer to G also as the *project network*. Now consider a particular realization $p = (p_1, \ldots, p_n)$ of the random processing time vector $\mathbf{p} = (\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n)$. Since there are no resource constraints, every job j can complete at its earliest possible completion time $C_j = C_j(p)$, which is equal to the length of a longest path in G that ends with j, where the length of a job j is its processing time p_j.

The *earliest project completion* or *makespan* for the realization p is then $C_{\max}(p) := max_j C_j(p) = \max_P \sum_{j \in P} p_j$, where P ranges over all inclusion-maximal paths of G. Since the processing times $\mathbf{p}_j$ are random, the makespan C_{max} is also a random variable, and it may be written as $C_{\max} = \max_P \sum_{j \in P} \mathbf{p}_j$, i.e., as the maximum of sums over subsets of a common set of random variables. An example is given in Figure 3.

The main goal of project risk analysis is to obtain information about the distribution of this random variable C_{max}.



$$C_{\max} = \max \{\ \mathbf{p}_1 + \mathbf{p}_3 + \mathbf{p}_6\,,\ \mathbf{p}_1 + \mathbf{p}_5 + \mathbf{p}_7\,,\ \mathbf{p}_2 + \mathbf{p}_4 + \mathbf{p}_6\,,\ \mathbf{p}_2 + \mathbf{p}_4 + \mathbf{p}_7\,,\ \mathbf{p}_2 + \mathbf{p}_5 + \mathbf{p}_7\ \}$$

Figure 3: An example project network and its makespan $C_{\max}$

Fulkerson noticed the systematic underestimation

$$C_{\max}\big(E(\mathbf{p}_1), \ldots, E(\mathbf{p}_n)\big) \leq E\big(C_{\max}(\mathbf{p}_1, \ldots, \mathbf{p}_n)\big)$$

when one compares the "deterministic makespan" $C_{\max}(E(\mathbf{p}_1), \ldots, E(\mathbf{p}_n))$ obtained from the expected processing times $E(\mathbf{p}_j)$ with the expected makespan $E(C_{\max}(\mathbf{p}))$. This error may in fact become arbitrarily large with increasing number of jobs or increasing variances of the processing times [9]. Equality holds if and only if there is one path that is the longest with probability 1, see Theorem 1 below. The error becomes even worse if one compares the deterministic value $C_{\max}(E(\mathbf{p}_1), \ldots, E(\mathbf{p}_n))$ with quantiles t_q such that $Prob\{C_{\max}(\mathbf{p}) \leq t_q\} \geq q$ for large values of q (say $q = 0.9$ or 0.95).

A simple example is given in Figure 4 for a project with n parallel jobs that are independent and uniformly distributed on [0,2]. Then the deterministic makespan $C_{\max}(E(\mathbf{p}_1), \ldots, E(\mathbf{p}_n)) = 1$, while $Prob(C_{\max} \leq 1) \to 0$ for $n \to \infty$. Similarly, all quantiles $t_q \to 2$ for $n \to \infty$ (and $q > 0$).

This is the reason why good practical planning tools should incorporate stochastic methods.

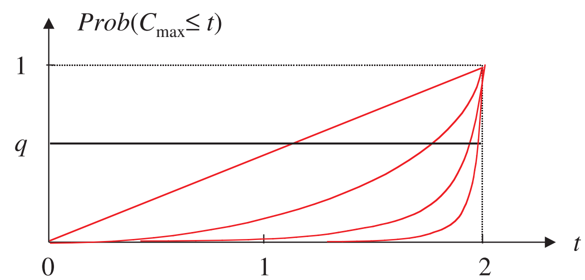


Figure 4: Distribution function of the makespan for $n = 1, 2, 4, 8$ parallel jobs that are independent and uniformly distributed on [0,2].

Theorem 1. *Let $G = (V, E)$ be a project network with random processing time vector* $\mathbf{p}$. *Then*

$$C_{\max}\big(E(\mathbf{p}_1), \dots, E(\mathbf{p}_n)\big) \leq E(C_{\max}\big(\mathbf{p}_1, \dots, \mathbf{p}_n)\big).$$

Equality holds iff there is one path that is the longest with probability 1.

Proof. Since $C_{\max}$ is the maximum of sums of processing times, it is obviously a convex function of p. Thus the inequality is a special case of Jensen's inequality for convex functions. We give here an elementary proof for C_{max}.

Let $P_1, \dots, P_k$ be the inclusion-maximal paths of G and let $Y_1, \dots, Y_k$ denote their (random) length, i.e., $Y_i := \sum_{j \in P_i} \mathbf{p_j}$. Then $C_{\max} = \max_i Y_i$, and

$$\begin{aligned} C_{\max}(E(\mathbf{p})) = \max_i \sum_{j \in P_i} E(\mathbf{p}_j) &= \max_i E(\sum_{j \in P_i} \mathbf{p}_j) = \max_i E(Y_i) \\ &= E(Y_{i_0}) \quad \text{assume that the maximum is attained at } i_0 \\ &\leq E(\max_i Y_i) \quad \text{since } Y_{i_0} \leq \max_i Y_i \quad \text{as functions of } p \\ &= E(C_{\max}(\mathbf{p})). \end{aligned}$$

Now assume that Y_1 is the longest path with probability 1. Then, with probability 1, $C_{\max} = Y_1 \geq Y_i$. Hence $E(C_{\max}) = E(Y_1) \geq E(Y_i)$ and the above calculation yields $C_{\max}(E(\mathbf{p})) = \max_i E(Y_i) = E(Y_1) = E(C_{\max})$.

In the other direction assume that $E(C_{\max}(\mathbf{p})) = C_{\max}(E(\mathbf{p}))$. Let w.l.o.g. P_1 be the longest path w.r.t. expected processing times $E(\mathbf{p}_j)$. Then $E(Y_1) = E(C_{\max}(\mathbf{p}))$ and

$$\begin{aligned} 0 = E\big(C_{\max}(\mathbf{p})\big) - C_{\max}\big(E(\mathbf{p})\big) &= E\big(\max_i Y_i - \max E(Y_i)\big) \\ = E(\max E(Y_i) - Y_1) &= \int \big(\max E(Y_i) - Y_1\big) dQ. \end{aligned}$$

Since the integrand in non-negative, it follows that it is 0 with probability 1. Hence $Y_1 = \max E(Y_i) = C_{\max}$ with probability 1. □

The probabilistic version of PERT is based on the second statement of this theorem. It only analyzes the distribution of the path with the longest expected path length. It thus fails when there are many paths that are critical with high probability.

The algorithm of Fulkerson uses the *arc diagram* of the precedence graph G, which is common also to PERT. It considers jobs of a project as arcs of a directed graph instead of vertices. This construction uses a directed acyclic graph $D = (N, A)$ with a unique source s and a unique sink t. Every job j of G is represented by an arc of D such that precedence constraints are preserved, i.e., if (i, j) is an edge of G, then there is a path from the end node of i to the start node of j in D. Figure 5 gives an example. Such a representation is called an *arc diagram* (sometimes also *PERT network*) of the project. In

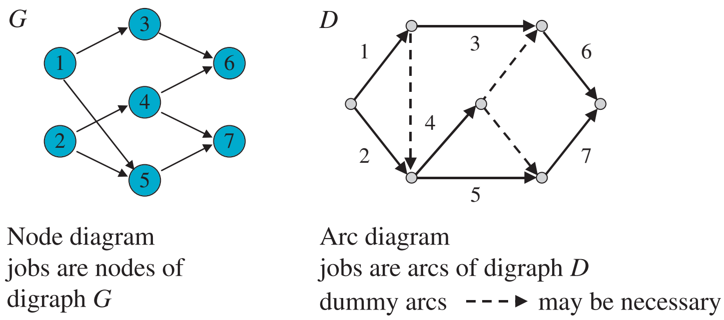


Figure 5: Arc diagram of the project network of Figure 3

general, one needs additional arcs (so-called *dummy arcs*) to properly represent the precedence constraints. Arc diagrams are thus not unique, but as dummy arcs obtain processing time 0, this ambiguity has no influence on the makespan.

Fulkerson assumes that stochastic dependencies may only occur in job bundles, where a bundle consists of all jobs with the same end node in the arc diagram. His algorithm then computes for each node v a value t_v that is iteratively obtained along a topological sort of the arc diagram as

$$t_v = E_{Q_v}\big(\max_{(u,v)\in E}\{t_u + \mathbf{p}_{(u,v)}\}\big),$$

where Q_v is the joint distribution of the processing times in the bundle of jobs ending in v, and the maximum is taken over all arcs in this bundle. A simple inductive argument shows that this gives indeed a lower bound on the expected makespan.

Fulkerson applies this to discrete job processing times, and so his algorithm is exponential in the maximum size of a bundle. He already noticed that it is computationally difficult to compute the exact value of the expected makespan, which was later mathematically confirmed by Hagstrom [8]. Hagstrom considers the following two problems:

> MEAN: Given a project network with discrete, independent processing times $\mathbf{p}_j$, compute the expected makespan $E(C_{\max}(\mathbf{p}))$.
>
> DF: Given a project network with discrete, independent processing times $\mathbf{p}_j$ and a time t, compute the probability $Prob\{C_{\max}(\mathbf{p}) \le t\}$ that the project finishes by time t.

She shows that DF and the 2-state versions of MEAN, in which every processing time $\mathbf{p}_j$ has only two discrete values, are $\#\mathcal{P}$-complete.

The complexity status of the general version of MEAN is open (only the 2-state version, which has a short encoding, is known to be $\#\mathcal{P}$-complete).

If the processing times $\mathbf{p}_j$ may take more than 2 values, the problem has a longer encoding that in principle could admit a polynomial algorithm for solving Mean. However, Hagstrom provides some evidence that problems with a long encoding may still be difficult, since Mean and DF cannot be solved in time polynomial in the number of values of $C_{\max}(\mathbf{p})$ unless $\mathcal{P} = \mathcal{NP}$.

These results show that efficient methods for calculating the expected makespan or quantiles of the distribution function of the makespan are very unlikely to exist, and thus justify the great interest in approximate methods such as bounds, simulation etc. that started with the work of Fulkerson. The search for "expected completion time" +network in Google Scholar currently shows more than 1,500 results.

References

[1] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows. Theory, Algorithms, and Applications*. Prentice Hall, Englewood Cliffs, NJ, 1993.

[2] Robert G. Bland and James B. Orlin. IFORS' Operational Research Hall of Fame: Delbert Ray Fulkerson. *Intl. Trans. in Op. Res.*, 12:367–372, 2005.

[3] Grant R. Cates. *Improving Project Management With Simulation And Completion Distribution Functions*. PhD thesis, University of Florida, Orlando, Florida, 2004.

[4] Willard Fazar. Program evaluation and review technique. *The American Statistician*, 13(2):10, 1959.

[5] D. R. Fulkerson. Expected critical path lengths in PERT networks. *Oper. Res.*, 10:808–817, 1962.

[6] D. Ray Fulkerson. Expected critical path lengths in PERT networks. Technical Report RM-3075-PR, RAND Cooperation, March 1962.

[7] D. Ray Fulkerson. Scheduling in project networks. Technical Report RM-4137-PR, RAND Cooperation, June 1964.

[8] Jane N. Hagstrom. Computational complexity of PERT problems. *Networks*, 18:139–147, 1988.

[9] Ursula Heller. On the shortest overall duration in stochastic project networks. *Methods Oper. Res.*, 42:85–104, 1981.

[10] J. J. Moder and C. R. Phillips. *Project management with CPM and PERT*. Reinhold, New York, 1964.

[11] Rolf H. Möhring. Scheduling under uncertainty: Bounding the makespan distribution. In Helmut Alt, editor, *Computational Discrete Mathematics: Advanced Lectures*, volume 2122 of *Lecture Notes in Computer Science*, pages 79–97. Springer-Verlag, 2001.

Rolf H. Möhring
Technische Universität Berlin
Institut für Mathematik
Straße des 17. Juni 136
10623 Berlin
`rolf.moehring@tu-berlin.de`