A New Stochastic Engine for PERT

by

Dan Trietsch[1]

Lilit Mazmanyan[1]

Lilit Gevorgyan[1]

and

Kenneth R. Baker[2]

July, 2009

Revised, February 2010[3]

Abstract

We obtained field data from two project organizations, including historical estimates and historical realizations of activity processing times. We use the data from one organization to validate the applicability of the lognormal distribution for project activity times. We use the data from the other organization to demonstrate the applicability of the Parkinson distribution in certain environments. Based on these empirical findings, we update and validate the systemic error model for activity time distributions in PERT, and we show that the classical PERT model is not supported by the data we collected. The main deficiency we expose is the stochastic independence assumption, but the traditional estimation method is also problematic. As a result, the stochastic engine in classical PERT is neither effective nor efficient. By contrast, the updated systemic error model provides more powerful results with less user input. It obtains all the necessary information using only elementary estimates and historical regression.

[1] College of Engineering, American University of Armenia, Yerevan
[2] Tuck School, Dartmouth College, Hanover NH

A New Stochastic Engine for PERT

Introduction

In this paper we develop and begin to validate a new approach to stochastic analysis of project activity times. The major tools used in project scheduling are Critical Path Method (CPM) and Program Evaluation and Review Technique (PERT). In particular, PERT is the tool of choice when scheduling must recognize the probabilistic nature of activity times within a project. At the heart of the PERT method is a set of assumptions that facilitates a systematic and intuitively appealing method for modeling stochastic behavior in projects. We refer to these assumptions and the method built on them as the *stochastic engine* in PERT. Unfortunately, the stochastic engine in PERT is neither effective nor efficient. It is ineffective because it provides unreliable results, and it is inefficient because it demands too much input from users. Therefore, the development of a new engine would be a welcome addition to the techniques available for project scheduling. In this research, we develop a radically different design that provides more powerful and useful results than the PERT method and requires less input from users.

The PERT method, introduced by Malcolm et al. (1959), prescribes a series of steps that enables the scheduler to model a project in stochastic terms. After 50 years of application, PERT is a familiar staple in textbooks on project scheduling (e.g., Meredith and Mantel, 2009) and management science (e.g., Hillier and Hillier, 2007, and Ragsdale, 2008). The steps have been reproduced so often that they constitute a familiar "recipe" for scheduling stochastic projects, starting with three estimates for the duration of each activity, an approach we call the *triplet method*. We provide an outline of the typical textbook recipe in Appendix A.

Over the years, various improvements have been proposed to enhance the stochastic analysis in PERT. The first of these—Clark (1961)—provides an approximation that compensates for the tendency of PERT to underestimate the expected project length due to network interactions. (We use the term *Jensen gap*[4] to refer to the difference between the expected project length and the length of the deterministic critical path obtained by replacing all activity times by their expected values.) Clark's work shows how to approximate the size of the Jensen gap. Van Slyke (1963) proposes Monte Carlo simulation as a practical solution for the same problem and as a way to estimate the criticalities of various activities. We consider such enhancements as part of PERT.

Ever since the debut of the PERT model, a cornerstone of conventional stochastic project scheduling has been the assumption that processing times of project activities are statistically independent. However, this assumption leads to an untenable conclusion: as the number of activities along the critical path grows large, the coefficient of variation (*cv*) of the project length tends to zero. Thus, we should not be surprised when such a model leads to poor estimates of project duration. Indeed, many authors have noticed that PERT often yields disappointing results (e.g., Klingel, 1966; Schonberger, 1981). The most prevalent explanation offered is the Jensen gap. However, the implication that the Jensen gap explains the discrepancy between planning and reality does not appear to be

[4] The term derives from a specialized version of Jensen's inequality, which states that the expected value of a maximum is at least as large as the maximum of the corresponding expected values.

supported by any published empirical evidence. On the contrary, by running simulation studies with independently distributed processing times, it is possible to verify that *cv* of project length still tends to zero for large projects even after accounting for the Jensen gap. The Jensen gap is real and often important, but it does not explain the practical shortcomings of PERT as a model for stochastic projects, and certainly not when simulation is employed. We assert that a more serious culprit is the independence assumption.

The simplest model that accounts for stochastic dependence invokes *basic linear association* (Esary, et al., 1967, Baker and Trietsch, 2009). This model employs a multiplicative stochastic error that is constant for each project but varies across projects. In our context, this error term represents estimation bias. Trietsch (2005) proposed using the model, also known as the *systemic error model*, as the basis for estimating processing time distributions and using the results to set optimal project safety times. The present work was originally aimed at validating that model with field data. The validation was promising (Gevorgyan, 2008), and the analysis led to a simplified model based on the lognormal distribution. In this paper, we describe the new version of the systemic error model as well as the empirical data that support it.

A few technical building blocks help guide our work. We summarize them here and elaborate in Appendix B. First, the traditional beta distribution formulas have a structural tendency to underestimate the variance of activity durations. Second, the lognormal distribution is more plausible than the beta distribution as a model for an activity time. Moreover, the lognormal supports its own central limit theorem and can be used as a model for the project length. Third, we must also represent the Parkinson effect (Parkinson, 1957), which occurs when activities rarely finish early but a large proportion completes precisely on time. We introduce the Parkinson distribution as a simple and intuitive model that works well in our empirical tests.

Our new design explicitly models stochastic dependence among activity times and uses regression analysis of historical results to estimate parameters for processing time distributions. We draw on the properties of linear association to construct a representation of estimation bias. We develop the systemic error model relying on the assumption that activity durations are inherently lognormal but possibly influenced by the Parkinson effect. We apply our model in two empirical datasets, validating our theory with field data. Our validation not only supports the new design but also reinforces the perception that the PERT model is flawed. Finally, we conclude by emphasizing the differences between our approach and the PERT approach.

We refer to the result as *PERT 21*—PERT for the 21st Century. Our aim is not only to improve the quality of the PERT stochastic analysis but also to open the door to other sophisticated algorithms that have been developed for PERT/CPM over the years. For example, Baker and Trietsch (2009) show how to apply stochastic sequencing and stochastic crashing to projects *provided* valid distributions are given; here we show how to obtain such valid distributions.

The data we collected from one project organization demonstrate a pure Parkinson effect (meaning that no activities finish early). Although the number of historical data points in our field study was small, our model was able to "predict" the performance of each project based solely on information garnered from the *other* projects. That information does not include optimistic and pessimistic estimates as in PERT. In fact, our analysis showed that traditional PERT is inadequate even with perfect variance estimates.

In the other set of projects, earliness was reported often, so we applied the general form of the Parkinson distribution for that set. In the general form, early activities are reported correctly with some probability and reported "on-time" otherwise. We should mention, however, that the data for this set of projects were given in crude units (all estimates were in months and all realizations in weeks). This is just one reason why the results we report should be re-validated with other data sets. Nonetheless, our model at least works for the particular organizations we studied, so it shows promise for other cases.

Estimation Bias

Estimates of activity durations are subject to *estimation bias* due to such factors as the following:

- optimism and pressure to produce attractive estimates from project champions who tend to emphasize opportunities,
- pessimism and pressure to produce cautious estimates from skeptics who tend to emphasize risks,
- mistakes caused by human error, or
- failure to anticipate possible problems.

Often, several activities are simultaneously affected by such factors as weather conditions, general economic conditions, and such unpredictable events as accidents or employee turnover. For convenience, we treat the combination of all these causes as estimation bias.

Estimation bias is not known in advance for any project, so it must be treated as random. Therefore, in addition to adjusting for the *average* bias, we must also account for the *variance* of the bias. This variance is the cause of statistical dependence among the deviations from predicted durations in a given project. If all estimates in all projects were equally biased, we could eventually learn to adjust for bias perfectly. But because the deviations are random and cannot be predicted precisely, estimation errors are positively correlated with each other even if the underlying distributions are independent. Specifically, optimistic estimates will lead to consistently longer-than-expected activity durations throughout the project, whereas pessimistic estimates will lead to consistently shorter-than-expected activity durations. In either case, the deviations are positively correlated.

Modeling Bias using Linear Association

Let the random variable p_j denote the processing time for activity j, and let μ_j and σ_j be its mean and standard deviation. Let X_j be an *estimate* of p_j, where we treat X_j as a random variable. Let the *nominal* estimate be $e_j = \mathrm{E}(X_j)$, and let $\mathrm{V}(X_j)$ denote the variance of the estimate. Random effects that are specific to activity j are modeled by $\mathrm{V}(X_j)$. We assume that the estimates are independent because it would be difficult to come up with an adequately posed set of dependent estimates. (Importantly, we do not impose such an assumption on the actual processing times.) We model estimation bias by introducing an additional independent random variable B that multiplies the estimate to obtain the distributions of the processing time. In other words, we assume that $p_j = BX_j$. Let μ_B and $V_B = \sigma_B^2$ denote the mean and variance of B.

For present purposes, we assume that our projects are serial, so the project length is given by the sum of all activity durations. Serial projects have no Jensen gaps. Therefore, we can study the independence assumption without confounding it with the Jensen gap. Real projects are seldom serial, however, so in practice activity time distributions should be used mainly as inputs for simulation. Consider a chain of activities, π, and let L_π denote its length. Temporarily, we ignore the possibility that any activity along π will be delayed by a predecessor outside the chain. The main task in project scheduling is to characterize the distribution of L_π. Because B and X_j are independent, we have $\mu_j = \mu_B e_j$. However, the multiplication of the processing times by the same realization, $B = b$, introduces dependence between any pair of processing times. Specifically, we obtain:

$$\sigma_j^2 = \mu_B^2 \mathrm{V}(X_j) + \mathrm{V}(X_j)V_B + V_B e_j^2 \quad \text{and} \quad \mathrm{COV}(p_i, p_k) = V_B e_i e_k$$

We can separate σ_j^2 into two parts, $\mu_B^2\mathrm{V}(X_j) + \mathrm{V}(X_j)V_B$ and $V_B e_j^2$. The former equals $\mathrm{E}(B^2)\mathrm{V}(X_j)$ and the latter is a special case of $V_B e_i e_k$. Now assume path π is the nominal critical path, characterized by the longest sum of expectations among all paths, and let the mean and variance of its length be denoted μ and σ^2. Temporarily, assume that the nominal critical path will actually be critical; that is, it will not be delayed by any activity that is not part of it. Then

$$\mu = \mu_B \Sigma_{j\in\pi} e_j \quad \text{and} \quad \sigma^2 = \mathrm{E}(B^2)\Sigma_{j\in\pi}\mathrm{V}(X_j) + (\sigma_B \Sigma_{j\in\pi} e_j)^2$$

The element $(\sigma_B\Sigma_{j\in\pi}e_j)^2$ imposes a lower bound of $\sigma_B\Sigma_{j\in\pi}e_j$ on σ. This bound is a multiple of the expected length of path π; namely, a multiple of σ_B/μ_B. In contrast to the case of independent activity durations, the *cv* of project length does not tend to zero as the number of activities on the chain grows large: it always exceeds σ_B/μ_B. Similarly, the bias in the estimate of mean project length is a multiple of the estimate, namely, $(\mu_B - 1)$.

Now we revisit the assumption that there are no delays due to activities outside the chain. That assumption allowed us to ignore the difference between the expected project length and the sum of the μ_j values along the nominal critical path—that is, to ignore the Jensen gap. However, it can be shown that the Jensen gap itself is subject to the exact same bias (Baker and Trietsch, 2009). Therefore, the results we obtain here for serial projects apply in general, but instead of using convolutions we must use simulation. Alternatively, one of the existing approximation methods could be employed to account for the Jensen gap.

Estimates for the Systemic Error Model

Consider the task of estimating $\mathrm{V}(X_j)$. The triplet method of PERT is notoriously dysfunctional. As we discuss in Appendix B, it is likely to lead to systematic underprediction of variability. In addition, Woolsey (1992) provides anecdotal evidence that it is often not based on objective analysis and that practitioners consider it onerous. Therefore, we limit the input elicited from experts to the absolute minimum: a single-point estimate, e_j, for each activity time. (Indeed, in our field work we were able to obtain historical estimates of means and the actual realizations, but not variance estimates.) This approach requires that we estimate $V(X_j)$ based on history, as a function of e_j.

The systemic error model presented by Trietsch (2005) relies on the classical central limit theorem. Here, we simplify it slightly and adapt it to the lognormal distribution. We assume lognormal distributions for both B and X_j. The product of two independent lognormal random variables is lognormal, so our assumption implies that p_j is lognormal. But because B is shared by all activities in the project, the processing times p_j are positively correlated. When p_j is lognormal and we divide it by a constant such as e_j, it remains lognormal, and its *cv* does not change. Therefore, it is convenient to work with the appropriate logarithm, $\ln(p_j/e_j)$.

Suppose we have historical records for $K > 1$ projects, and we amend our earlier notation to include a double index, jk, where $j = 1, 2, \ldots, n_k$ and $k = 1, 2, \ldots, K$. Here j denotes the jth activity of project k, and project k has n_k activities. (Optionally, we may elect to treat some subprojects as projects in their own right. That approach would be appropriate if a subproject is estimated relatively independently of other project activities or if it relies on a different set of resources.) For project k, our historical data consist of n_k pairs (p_{jk}, e_{jk}), where p_{jk} is the realization and e_{jk} is the original estimate. We can estimate the logarithm of the bias for project k, $\ln(b_k)$, by

$$\hat{\ln}(b_k) = \frac{\sum_{j=1}^{n_k} \ln\left(p_{jk} / e_{jk}\right)}{n_k}$$

(Later we discuss an approach to estimating these values that works better when the Parkinson effect is involved.) If we give each of these estimators a weight proportional to n_k, we obtain the following estimator of $\ln(\mu_B)$,

$$\hat{\ln}(\mu_B) = \frac{\sum_{k=1}^{n_k} n_k \hat{\ln}(b_k)}{\sum_{k=1}^{n_k} n_k}$$

and we obtain the following unbiased estimator of s_B^2 (the variance of $\ln(\mu_B)$)

$$\hat{s}_B^2 = \frac{\sum_{k=1}^{K} n_k \left(\hat{\ln}(b_k) - \hat{\ln}(\mu_B)\right)^2}{\sum_{k=1}^{K} n_k - K}$$

One way to estimate s_k, the standard deviation of $\ln(p_{jk}/e_{jk})$, is to use the variance of the set $\{\ln(p_{jk}/e_{jk})\}$ for $j = 1, \ldots, n_k$. (Again, we discuss an alternative approach later.) We expect that any new project in the same family will possess a logarithmic bias drawn from a normal distribution with mean $\hat{\ln}(\mu_B)$ and variance $\hat{s}_B^2$. Similarly, we expect that new project to have a lognormal processing time distribution with *s* drawn from the same distribution that generated the s_k values for the *K* projects in the history. Accordingly, we can calculate the mean and the variance of the s_k estimates similarly to the way we calculated $\hat{\ln}(\mu_B)$ and $\hat{s}_B^2$. We denote these estimators by $\hat{s}_X$ and V($\hat{s}_X$), respectively. It is also necessary to check whether the historical s_k estimates and the bias estimates are correlated, in which case we must account for such correlation in the model.

After calculating the values of these estimators, we have all the data we need to simulate the distribution of a new project. Suppose we wish to generate a sample containing *r* repetitions. (Think of *r* rows for the replications and *n* columns for the *n* activities.) Using $\hat{s}_X$ and V($\hat{s}_X$), we generate one s_X realization per row and store it in an auxiliary column. Similarly, using $\hat{\ln}(\mu_B)$ and $\hat{s}_B^2$, we generate a bias element for each row and store it in another auxiliary column. Using the original e_j estimates, we generate for each of the *n* columns a normal random variable with mean ln(e_j) and standard deviation $\hat{s}_X$. Before actually storing it, we incorporate the bias element simulated for that row, keeping in mind that adding logarithms corresponds to multiplying their exponents. Thus, all elements in the row share the same bias element and the same variance, but no two rows have the same parameters.

Validation of the Lognormal Model

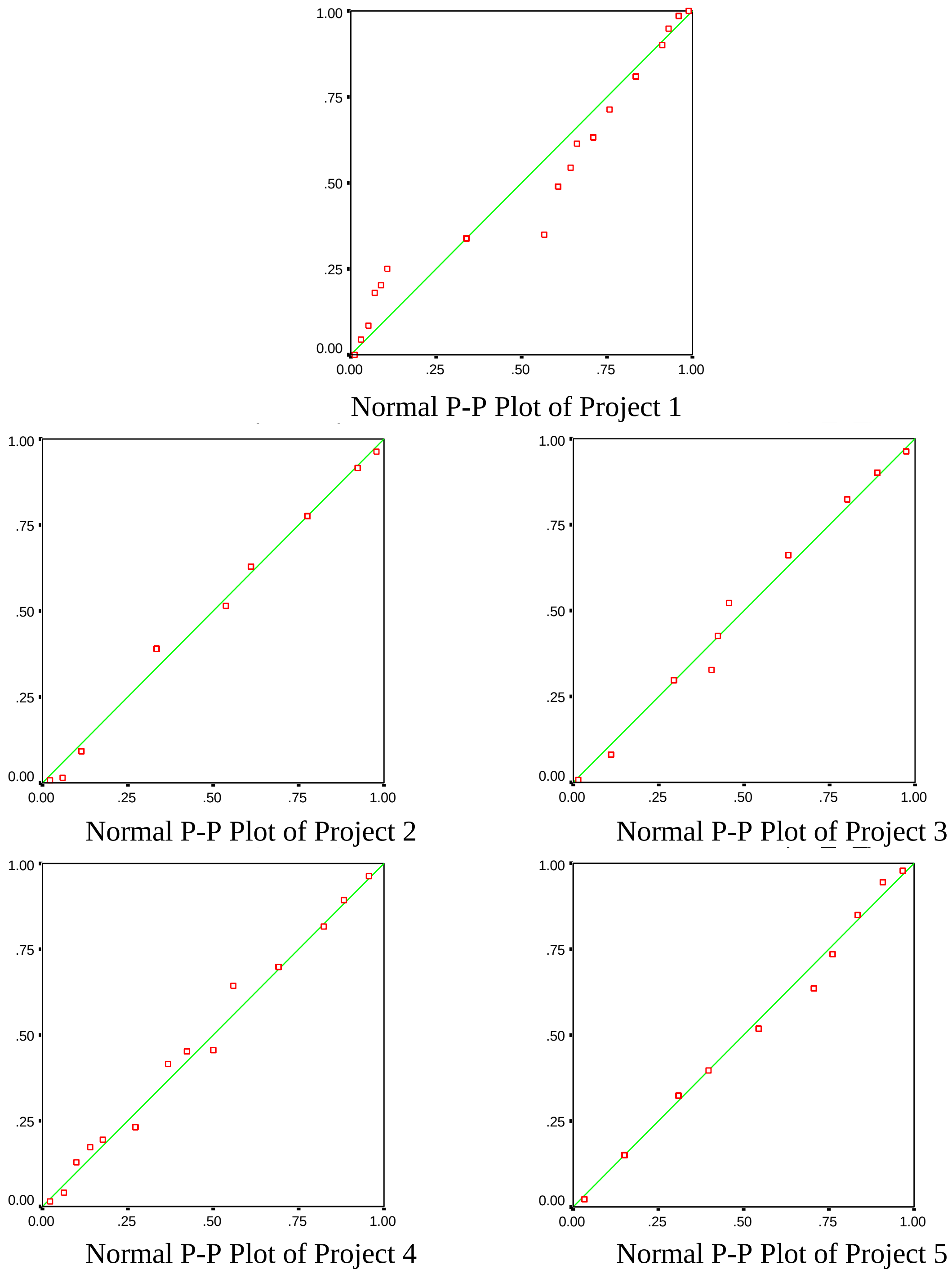
We obtained data from two NGOs about the estimated and actual durations of activities in two families of projects with five and nine projects, respectively, all completed within the last few years. Both organizations coordinate humanitarian projects using subcontractors to perform various activities or subprojects. A fraction of the activities at one NGO was not subcontracted. One of the two organizations subcontracts numerous low-level activities per project (measured in weeks), whereas the other organization subcontracts larger sub-projects (measured in months). As a result we obtained one set with fewer projects but more activities per project and another set with more projects but fewer activities. In general, the projects were quite eclectic, including community organizing, productivity skills training, agricultural infrastructure development and even software development. There is a conceptual similarity between most of these projects and the Polaris project (for which PERT was originally developed), in the sense that activities were subcontracted to outside providers. Therefore, the processing time visible to the project manager might include queueing time within the supplier organization. Perhaps as a result, we measured high coefficients of variation (roughly between 0.6 and 0.95). Furthermore, we do not claim that our results are automatically valid for other types of environments. Such validation would require new sets of data and further research.

Because they were managed similarly, all projects in each family are assumed to be related and subject to the same estimation bias distribution, but the two families are different. All projects in the larger set exhibit the pure Parkinson effect, and we discuss them later. For the five projects in the smaller set, Figure 1 depicts normal P-P plots for ln(p_j/e_j) as generated by SPSS (SPSS, 2001). If the distribution of ln(p_j/e_j) is normal (that is, if p_j/e_j is lognormal), then the P-P plot should be close to the diagonal from the left bottom corner of the square to the top right corner. We observe that diagonal configuration in four of these plots.

P-P plots are central to our validation, so we elaborate. If we sample a continuous random variable with a given cumulative distribution function (cdf), the cdf can be used to transform the sample to an equivalent sample from a uniform distribution on the interval (0, 1), denoted U[0, 1]. The transformed sample can then be sorted and compared to the cdf of U[0, 1] , which forms the diagonal in the chart. A normal P-P plot is one way to display the results of such a comparison; but here, instead of using a *given* cdf, we estimate the mean and variance of the cdf from the sample. The convention used by SPSS divides the horizontal scale into n equal segments and places the sorted points at the centers of those segments. Thus, point j (the jth smallest realization) has a horizontal value of $(j - 0.5)/n$. The vertical value of point j is the transformed sampled value (probability) of that realization. The transformed sample can be tested for uniformity using standard statistical tests for goodness of fit. The most common test for this purpose is the Kolmogorov-Smirnov (K-S) test, on which we rely.

Figure 2 depicts the Q-Q plots associated with these P-P plots. Q-Q plots are related to P-P plots but do not transform the sample (SPSS, 2001). Instead, the vertical axis denotes actual realizations (logarithms, in our case) and the horizontal axis denotes their expected values, plotted as standard normal z-values with cdf probabilities of $(j - 0.5)/n$. Figure 2 reveals many ties where several sorted sample points have the same logarithm. Our data were provided in integer units, and the occurrence of such ties is a consequence. Project 1 exhibits suspiciously many points with $\ln(p_j/e_j) = 0$ (indicating completion precisely on time) and very few early points. Projects 4 and 5 also have several points with $\ln(p_j/e_j) = 0$, but they are less dominant. Furthermore, from the trend line depicted, it seems that all points in projects are arranged linearly. Given that the data were limited to points on an integer grid, the good linear fit of the lognormal distribution is especially encouraging. A similar observation applies to the subset of tardy activities in Project 1. Furthermore, the K-S test rejected the normality of the logarithmic values in project 1 with a p-value of 1% but accepted projects 2 through 5 with p-values of 8.8%, 9.0%, 83.2% and 37.7%, respectively). Based on these results we excluded project 1 from most of our initial analysis. (However, the p-values 8.8% and 9.0% are quite low. That observation motivated us later to revisit these projects and fit the Parkinson distribution to them.)

We continue our analysis of projects 2-5 under the assumption that the lognormal distribution applies. The P-P plots show that the $\ln(p_j/e_j)$ values are consistent with the hypothesis that they are sampled from a normal distribution with the same standard deviation, s. As mentioned in Appendix B, the cv of the lognormal is related to s by the relationship $s^2 = \ln(1 + cv^2)$. Therefore, linearity in the plots, indicating a common s for all activities, implies a constant cv for all activities in each project. Moreover, in all cases, the standard deviations were close to each other, roughly between $s = 0.53$ and $s = 0.78$. (The standard deviation of Project 1, when estimated by its tail only, is also in the same range.) Nonetheless, we do not recommend assuming that all projects share the same standard deviation.



Normal P-P Plot of Project 1

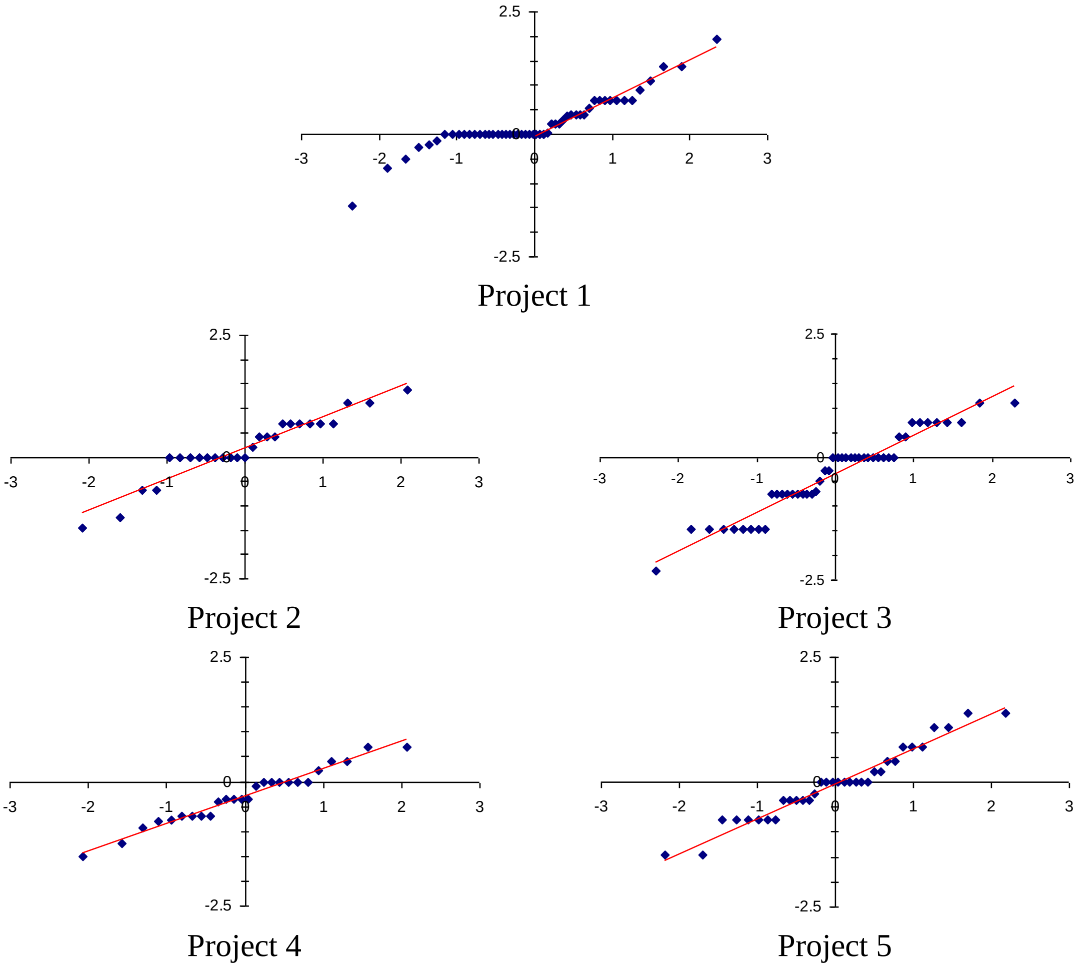
Normal P-P Plot of Project 2

Normal P-P Plot of Project 3

Normal P-P Plot of Project 4

Normal P-P Plot of Project 5

FIGURE 1



Project 1

Project 2

Project 3

Project 4

Project 5

FIGURE 2

Figure 2 suggests an alternative approach to estimating the parameters of the lognormal distribution of a project by fitting a regression line to the Q-Q plot. The slope of that line for project *k* is $\hat{s}_k$, and the value at which it intersects the *y*-axis serves as the bias estimate, $\hat{\ln}(b_k)$. Ignoring the error term, a typical equation in such a regression, relating to the activity with the *j*-smallest ln(*p*/*e*) of project *k*, has the form

$$\hat{\ln}(b_k) + z_{jk}\hat{s}_k = \ln \frac{p_{jk}}{e_{jk}} \tag{1}$$

where $z_{jk} = \Phi^{-1}((j - 0.5)/n_k)$, the *z*-value for which the standard normal distribution cdf, $\Phi(z)$, yields a probability of $(j - 0.5)/n_k$. That is, we treat the $\ln(p_{jk}/e_{jk})$ values as our dependent variables, and we obtain estimators for $\ln(b_k)$ and for s_k. We adopt this approach—in spite of the fact that it is not based on the minimal sufficient statistic—because it also works when the Parkinson effect is involved. In the pure case, we include only equations for tardy activities.

Having shown that the lognormal distribution fits the data reasonably well in all but one of the projects, we next wanted to demonstrate that our approach can estimate the cdf of the project completion time. Let $F_k(x)$ denote the cdf of project k. In our context, the challenge is not just to estimate each $F_k(x)$ but also to demonstrate validity. This is not an easy task with so few projects, but we outline a procedure that can help reject the approach if the cdfs are patently invalid. Suppose we estimate $F_k(x)$ using information in the other three projects. Under the traditional PERT assumptions, such history should be useless, but under the systemic error model, that history is relevant. The null hypothesis is that our model is valid. Given $F_k(x)$, we can find the estimated $F_k(\sum p_{jk})$ values. If we repeat the process for all four projects, then under the null hypothesis we should obtain a random sample of four draws from a U[0, 1] distribution. The resulting P-P plot is given on the left-hand side of Figure 3. The K-S test does not reject the hypothesis that this is a legitimate sample of four from a U[0, 1] distribution, although there are too few data points to really say that this is meaningful.

We repeated the process using the traditional PERT assumptions, but because we had no original PERT triplet estimates, we used the lognormal distribution with the actual *cv* that we obtained from the sample. In other words, we assumed that PERT generated perfect variance estimates. That assumption maximizes the chance that PERT will perform well. Furthermore, by using the same distribution, we obtain a fair comparison (regardless of whether the lognormal is indeed better than the beta, as we assert). The relevant P-P plot is given on the right-hand side of Figure 3. The PERT P-P plot starts much lower and ends much higher than the one on the left. This suggests that the variance estimate of PERT is too low. However, this sample passed the K-S test too, so we could not reject the hypothesis that the result is also a valid sample from the distribution U[0, 1]. Nonetheless, the fact that the plot starts lower and ends higher is what we expected, because PERT ignores the variance introduced by systemic error.

This result motivated us to look further. By using simulation results with 10,000 repetitions, we devised a new test based on measuring the sum of the two smallest vertical distances in the plot. We refer to it as the *complementary test* because the K-S test is triggered (indirectly) by large vertical distances; so the new test complements the K-S test. In this instance, the two minimal distances occurred between the bottom of the plot and the first point and between the last point and the top. The new test rejects this sample with a *p*-value of 4.7%. It does not reject the plot on the left, however.

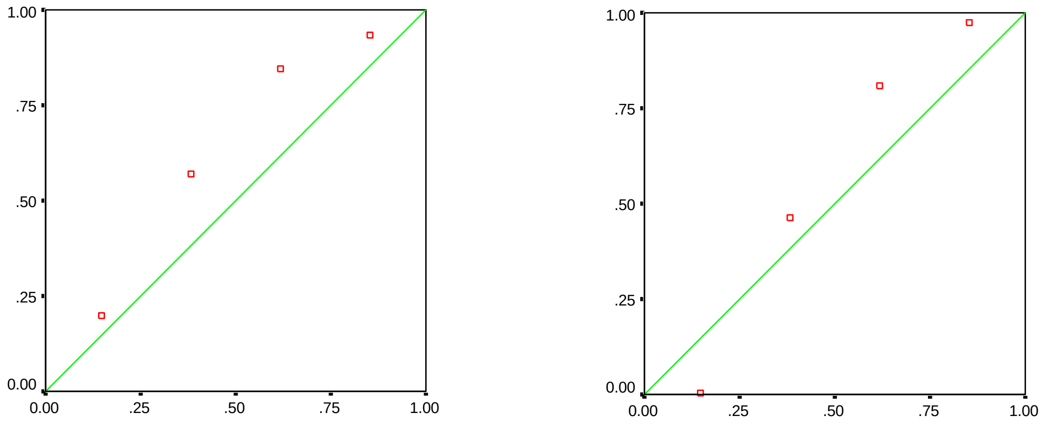
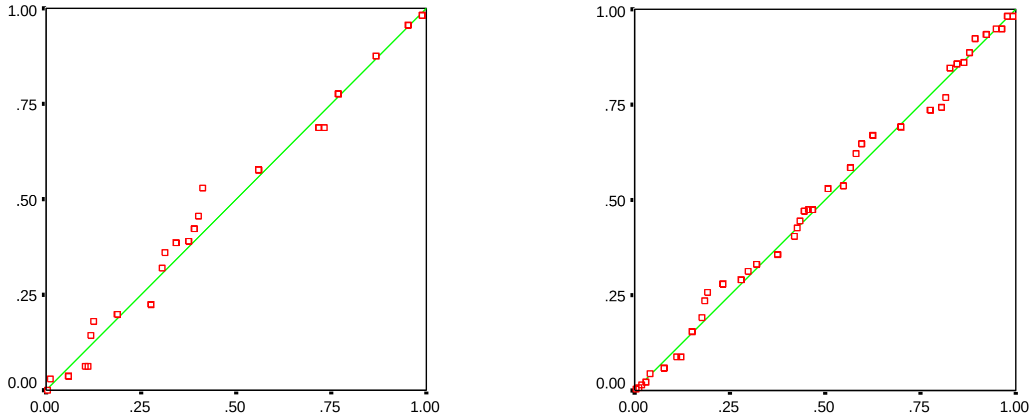


FIGURE 3

In addition, we obtained evidence that the four projects are different from each other and specifically that these differences are mainly due to different biases. To that end, we tested whether all distributions were identical without the bias correction by pooling the $\ln(p_{jk}/e_{jk})$ values from all four projects. The normality assumption of the combined set of $\ln(p_{jk}/e_{jk})$ was rejected with a p-value of 0.2%. We then corrected these values for bias—that is, we pooled the values of $\ln(p_{jk}/e_{jk}) - \hat{\ln}(b_k)$. This time, the hypothesis that all values come from a single normal distribution could not be rejected. The p-value was 31.9%. Figure 4 shows the relevant P-P plots. On the left, we show the results without bias correction and on the right, with the correction. Furthermore, the normality of the combined set was accepted after bias correction even though we ignored the differences in variance among the projects. As we noted above, these differences are not large.



Combined Data from Projects 2-3-4-5 without bias correction

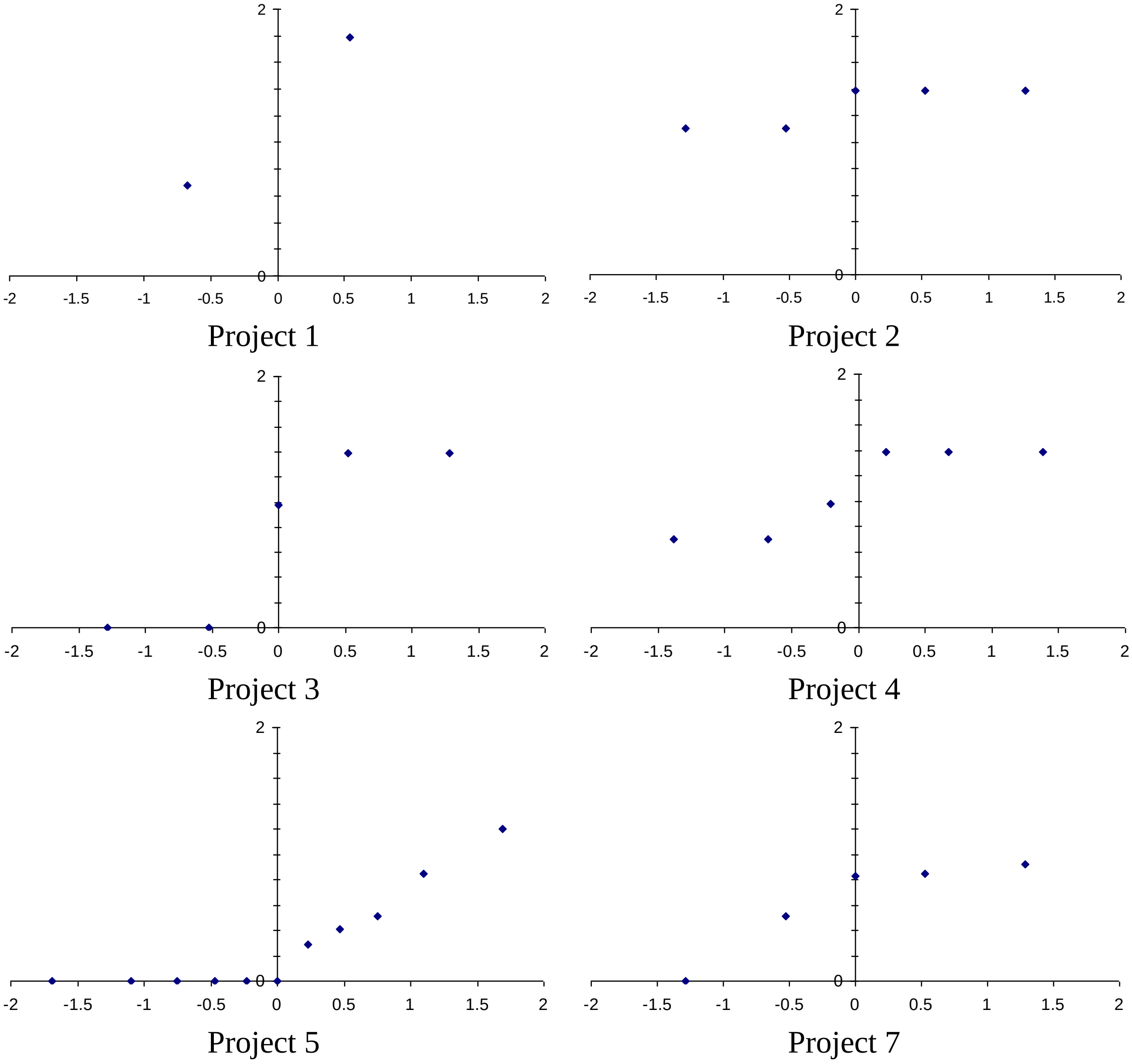
Combined Data from Projects 2-3-4-5 with bias correction

FIGURE 4

The analysis of the four projects was encouraging in the sense that it yielded reasonable results and demonstrated a clear bias effect. However, it did not provide strong evidence that PERT is inadequate. To explore this aspect of our findings further, we return to our earlier observation that the triplet model tends to favor low values of the *cv*. In the regression analysis of our four projects we obtained *cv* estimates of 0.5706, 0.9140, 0.5926 and 0.7883 (corresponding to estimates of *s* between 0.5309 and 0.7793). Judging by its tail, Project 1 would have *cv* = 0.8978, which is in the same range. As we point out in Appendix B, the triplet method leads to $cv \leq 0.8333$, and to reach that limit we must tolerate deviations of 5% as well as relatively small values of *min* and *mode*. However, two of our five estimates fell outside that range. Thus the PERT approach leads to estimates of the *cv* that are too restrictive. Indeed, if we were to use lower *cv* estimates in our check of PERT for these examples we might have even stronger evidence. But we used the *cv* values corresponding to our variance estimates. That is, we used *cv* values that would not likely have been obtained using PERT. (Our subsequent Parkinson analysis indicates even higher *cv* values.)

Validation of the Parkinson Model

Recall that the first project in the smaller set did not fit the lognormal assumptions as evaluated by the K-S test. A cursory analysis of the activity times quickly revealed that many realizations fit the estimates precisely, whereas only few were early. We took this as evidence that the Parkinson effect was involved. In the other set of nine projects, however, *all* activity times exhibited the pure Parkinson effect. (No activity was early.) The vertical axis in Figure 5 shows the $\ln(p_{jk}/e_{jk})$ values of those nine projects as a function of z_{jk}. Most projects had small n_k values (project 6—not depicted—had just one activity) but we decided to analyze them nonetheless. We see that no activities have negative logarithms (associated with $p_{jk} < e_{jk}$) and several activities have values of zero, associated with "perfect" executions.



Project 1

Project 2

Project 3

Project 4

Project 5

Project 7

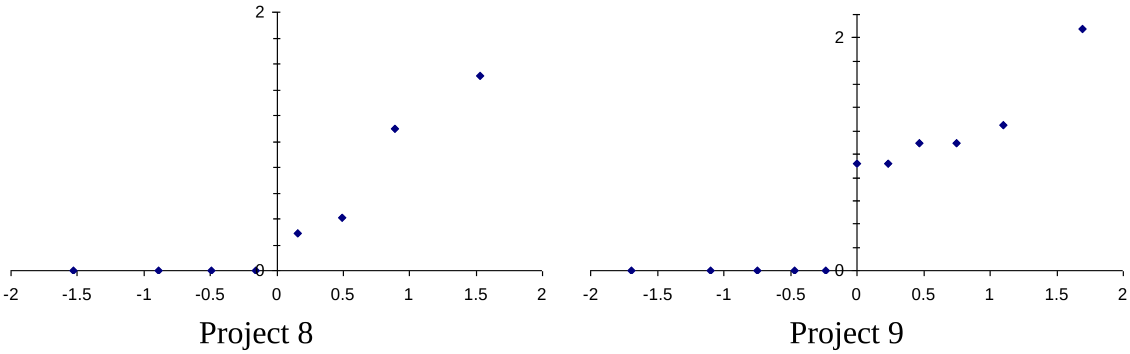


FIGURE 5

Although it is evident that these distributions are not lognormal, some of those that had sufficiently many tardy activities exhibited relatively straight tails. (Projects 5 and 8 are good examples.) That feature suggested that there may be a hidden lognormal variable behind the observable results. Accordingly we analyzed the distributions based only on their tails. For this purpose we ignored two of the projects that had too few tardy points to use (projects 1 and 6). Altogether we had 34 equations and we built regressions using 14 parameters: a bias ($\ln(b_k)$) and a standard deviation (s_k) for each of the seven remaining projects. Using the regression results, we identified a negative correlation between the bias values and the standard deviations. Such correlations should be accounted for during simulations. Initially, we used these estimates from the seven projects to simulate cdfs for the other two projects. For each of these two projects we calculated $F_k(\sum p_{jk})$ and obtained the (sorted) values 0.1931 and 0.5783. Then we generated cdfs for the seven remaining projects, but in each case we removed the data obtained from the project itself before simulating its cdf. Thus each project simulation used the activity estimates for its own activities along with “historical” information from *other* projects. The $F_k(\sum p_{jk})$ results of these projects were 0.0537, 0.1449, 0.3978, 0.5000, 0.7130, 0.8348 and 0.9196. Together with the previous two outcomes, we obtain the following set:

{0.0537, 0.1449, 0.1931, 0.3978, 0.5000, 0.5783, 0.7130, 0.8728 and 0.9196}

A K-S test does not reject the null hypothesis (that these observations come from a uniform distribution on the unit interval). (The complementary test using the sum of the two minimal distances—namely 0.0537 + [0.9196 − 0.8728] = 0.1005—leads to the same conclusion.) Figure 6 depicts the resulting P-P plot. Although our results are apparently excellent, we make no claim that it is always safe to use such a small set of active equations.

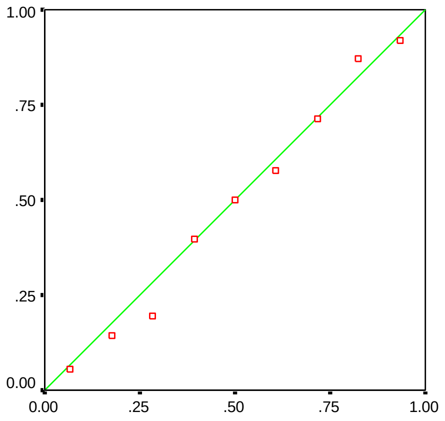
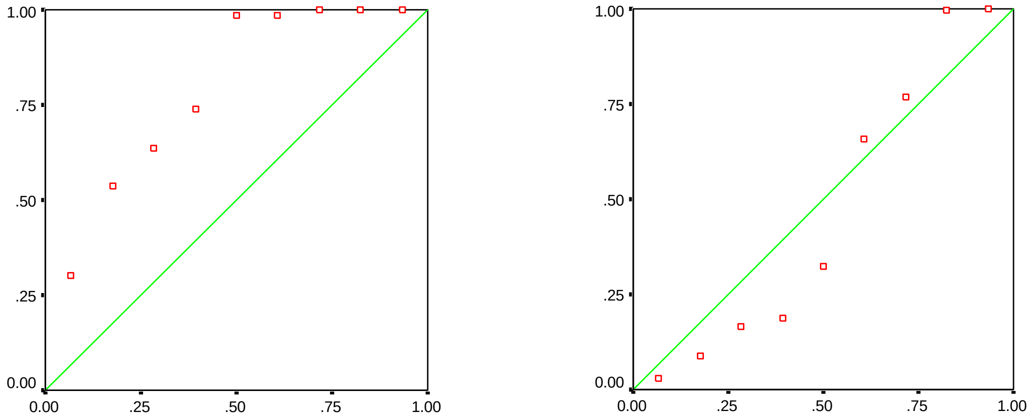


FIGURE 6

The left-hand side of Figure 7 depicts the results of regular PERT analysis for this case. Here, the K-S test rejected the null hypothesis resoundingly (as did the complementary test), but there is no need for a formal test. We observe that three points are practically on the edge of the chart—using four-digit precision, their $F_k(\sum p_{jk})$ values are 1.0000—and two more are very close. This configuration demonstrates the inadequacy of the model. Suppose we were to set safety times with a service level (probability of on-time completion) of 99%, then those three projects would be tardy and two more projects would be close. Safety times of 98% would fail for five projects. A well-known heuristic method for buffer setting, the “Critical Chain” (see Leach 2000), assigns arbitrary safety times of 50% of the estimate. In our sample of nine projects, such a safety time would fail in eight projects.



PERT results

PERT with mean bias correction

FIGURE 7

Malcolm et al. (1959) recommended correcting for the average historical bias (but not for its variance). If we were to do so here, the results would be as in the right-hand side of Figure 7. They are much improved relative to the left-hand side, but two projects would still be tardy if we set safety time to achieve service levels of 99% by this approach. (Of these two points, one is practically on the edge and the other is at 99.79%). Although the K-S test accepts this sample, the probability that two points out of nine would be so far out is negligible (and the complementary test rejects the null hypothesis with a *p*-value of 0.65%). We conclude that correcting for the mean bias is beneficial but not sufficient.

We reiterate that the data in Figure 7 are based on the exact *cv* estimates obtained in hindsight. Such precise estimates would not be available to PERT practitioners in reality. In addition, our model does not involve a Jensen gap. Thus the PERT failure that we demonstrate here is solely due to the independence assumption. In practice, with imperfect variance estimates and the Jensen gap to consider, PERT should perform even more poorly. By contrast, the results in Figure 6 are excellent. Thus, the systemic error model is supported by our two data sets. Furthermore, by rejecting the PERT model, with or without a correction for average bias, the statistical analyses indicate that a model such as ours is necessary.

The Parkinson Distribution

By observation of the five projects in the first set (Figure 2), it is apparent that in all projects a fraction of activity times was reported as precisely on time. Some of those can be explained by rounding: the data were given in crude units and our ratios are all members of a small subset of the rational numbers. Nonetheless, at least in some instances it is clear that the proportion of on-time points (with logarithms of 0) is relatively high. Furthermore, recall that we could not include project 1 in our analysis so far because it exhibited too many on-time points. Therefore, in the second stage of our research, we defined the Parkinson distribution to fit the first family of projects. When we ran simulations for these projects we obtained the following $F_k(C)$ values: 0.015, 0.206, 0.555, 0.665, and 0.780. Figure 8 depicts the P-P plot for this case. Although the value 0.015 is quite low, this P-P plot passes the K-S test and the complementary test. It is also higher than the minimal value of the regular PERT case (0.004). Thus we can say that the Parkinson distribution provides reasonable prediction for all the projects in the set, each based on the others.

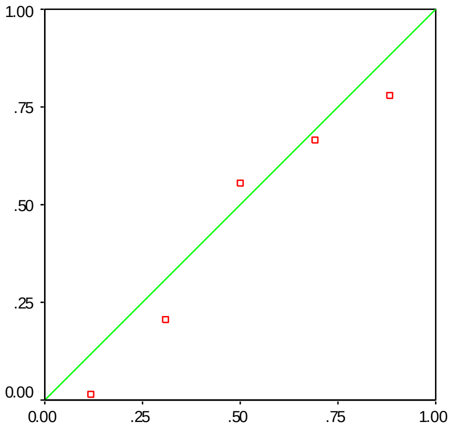


FIGURE 8

Summary and Conclusion

This research set out to validate the systemic error model of Trietsch (2005) and ultimately confirmed the applicability of the lognormal and the Parkinson distributions to project activities. Since the first use of the beta distribution in the PERT method, other distributions have been proposed, including the lognormal. The Parkinson effect is also well known. But apparently none of these ideas has been tested empirically. We focused on the lognormal distribution because of its several appealing properties and formulated a distribution to model the Parkinson effect. We reported that the lognormal distribution provided a surprisingly good fit for the activity times in one set of five projects, whereas one of these projects and another set of nine projects seemed to exhibit the Parkinson distribution. We then identified a partial Parkinson effect in the former set as well, and defined the Parkinson distribution to model such partial effects. Normatively, it is desirable to avoid the Parkinson effect, but that is easier said than done. Therefore, it is important to be able to analyze the performance of projects that exhibit the effect. When we analyzed those projects based on the assumption that the internal random variable is lognormal, we obtained useful results.

With these distributions in place, we were also able to validate the systemic error model. For each project, our validation involved using the other projects in the set as the basis for estimating distributional parameters. Using single activity estimates for each activity and the parameters obtained from the other projects, we generated a distribution for each project's completion time and estimated the completion probabilities, $F_k(\sum p_{jk})$, associated with the project's actual sum of activity completion times. If we do that for K projects we should obtain probabilities that correspond to a random sample of K independent realizations from a uniform distribution. We can then test the hypothesis that they form such a sample, and we can measure the probabilities that would hold under the PERT assumption.

The first set of projects provided statistical evidence of differences between project distribution times that can be explained by systemic error. Initially, we could handle only four out of the five projects in the family using the lognormal distribution, but with the general Parkinson model in place we could include them all reasonably well. There were not enough projects to reject the PERT model for that set using a traditional test. However, we were able to reject it using a new test. In the second set, we had less useful information per project but more projects. Here, we were able to show by classical tests that our results are statistically plausible, whereas similar analysis based on PERT would have failed. Stated differently, if planners were to use our approach for these projects and set safety times according to our estimated distributions, the safety times would perform approximately as planned. By contrast, if they were to use PERT, most projects would be tardy. For instance, if we had set safety times for 90% service levels under the PERT model, five out of nine projects would have failed, but only one would have failed under the systemic error model. Similar safety times based on 98% service would still fail in five projects under PERT but would be sufficient for *all* projects under our approach.

Technically, we advocate regression analysis instead of the PERT triplet method. To some extent, the use of regression for estimating activity times is an established technique. Furthermore, estimates that are based on solid regression analysis are likely to be more accurate and precise than those obtained by other methods (Shtub et al., 2004).

However, our use of regression extends conventional approaches, because we can use historical estimates that might have originally been developed with regression. Our approach is also special because when we simulate a new project we use our regression results not only for means but also for variances.

Our field data came from two NGOs, both of which use subcontractors to carry out most or all of the individual activities. Due to subcontracting, the processing time visible to the project manager likely includes queueing time within the supplier organization. That may explain the high coefficients of variation that we obtained. We suspect that similar queueing may also occur under a matrix organization structure, where projects compete for resources. One could hypothesize that when resources are dedicated, typical *cv* values will be lower because the project manager controls the queue and processing times do not include waiting. Checking that hypothesis requires further research. Additional research is required to quantify the effect of the particular industry on *cv*. For example, we might expect lower *cv* values in construction than in software engineering.

When the new engine is fitted in PERT, we can implement what we call PERT 21. PERT 21 merges sequencing and crashing techniques that were historically developed for CPM with the stochastic approach that is vital for practicability in project scheduling, as in PERT. Such a merger has been conspicuously absent until now. Baker and Trietsch (2009) show how this merger can be achieved when valid activity time distributions are available.

Finally, the systemic error model has implications for models other than PERT 21. In that sense, our research opens the way to a new, empirically-based approach to stochastic scheduling in other environments than project scheduling. It remains for further research to explore the application of similar principles to other problems in stochastic scheduling.

Acknowledgements: We are grateful to Kristine Sargsyan, Development Project Coordinator, and to Arman Navasardyan, Charity and Development Project Coordination Specialist, Action Against Hunger (ACF-USA).

References

Baker, K.R. and D. Trietsch (2009) *Principles of Sequencing and Scheduling*, John Wiley & Sons, New York.

Baker, K.R. and D. Trietsch (2010) Research Notes for Appendix A in *Principles of Sequencing and Scheduling* (Wiley, 2009).
URL: http://mba.tuck.dartmouth.edu/pss/.

Clark, C.E. (1961) The Greatest of a Finite Set of Random Variables, *Operations Research* 9, 145-162.

Clark, C.E. (1962) The PERT Model for the Distribution of an Activity Time, *Operations Research* 10, 405–406.

Esary, J.D., F.Proschan and D.W.Walkup (1967) Association of Random Variables, with Applications, *Annals of Mathematical Statistics* 38, 1466-1474.

Gevorgyan, L. (2008) Project Duration Estimation with Corrections for Systemic Error, Master Thesis, Industrial Engineering & Systems Management, College of Engineering, American University of Armenia, Yerevan, Armenia.

Grubbs, Frank E. (1962) "Attempts to Validate Certain PERT Statistics or Picking on PERT," *Operations Research* 10, 912-915.

Gutierrez, G.J. and P. Kouvelis (1991) Parkinson's Law and its Implications for Project Management, *Management Science* 37, 990-1001.

Hillier, F. and M. Hillier (2007) Introduction to Management Science, Irwin/McGraw-Hill, Boston.

Klingel, A.R. (1966) Bias in PERT Project Completion Time Calculations for a Real Network, *Management Science* (Series B) 13, B-194-201.

Leach, L.P. (2000) *Critical Chain Project Management*, Artech House.

Malcolm, D.G., J.H. Roseboom, C.E. Clark and W. Fazar (1959) Application of a Technique for a Research and Development Program Evaluation, *Operations Research* 7, 646-69.

Mazmanyan, L., V. Ohanian and D. Trietsch (2008) The Lognormal Central Limit Theorem for Positive Random Variables (Working Paper; see also Baker and Trietsch, 2010).

Meredith, J.R. and S.J. Mantel, Jr. (2009) *Project Management: A Managerial Approach*, Seventh Edition, John Wiley & Sons, New York.

Parkinson, C.N. (1957) *Parkinson's Law and Other Studies in Administration*. Houghton-Mifflin.

Ragsdale, C. (2008) *Spreadsheet Modeling and Decision Analysis*, South-Western.

Schonberger, R.J. (1981) Why Projects are "Always" Late: A Rationale Based on Manual Simulation of a PERT/CPM Network, *Interfaces* 11, 66-70.

Shtub, A., J.F. Bard and S. Globerson (2004) *Project Management: Processes, Methodologies, and Economics*, 2nd edition, Pearson Prentice Hall.

SPSS Help File, SPSS for Windows, Release 11.0.1.

Trietsch, D. (2005) The Effect of Systemic Errors on Optimal Project Buffers, *International Journal of Project Management* 23, 267–274.

Tversky A. and D. Kahneman (1974) Judgment under Uncertainty: Heuristics and Biases, *Science* 185, 1124-1131.

Woolsey R.E. (1992) The Fifth Column: The PERT that Never Was or Data Collection as an Optimizer, *Interfaces* 22, 112-114.

Appendix A. Brief Summary of the Basic PERT Method

1. For each activity obtain an estimate of its optimistic duration (*a*), its mode, or most likely duration (*m*), and its pessimistic duration (*b*). Assume that if these estimates are obtained by querying a third party, "optimistic" and "pessimistic" are understood to mean the literal minimum and maximum, rather than, say, the 10th and 90th percentiles of the distribution.

2. Calculate the mean and variance of the activity duration from the formulas:

   $\mu = (a + 4m + b) / 6$
   $\sigma = (b – a) / 6$

   Assume that these formulas are justified by a relationship (often unspecified) to the beta distribution model (see also Appendix B).

3. Treating the mean values as deterministic activity durations, identify the critical path. (This is called the *nominal* critical path.)

4. To find the mean length of the critical path, sum the mean durations of the activities along the nominal critical path. To find the variance of the length of the critical path, sum the variances of the activities along the nominal critical path. Assume that the activity times are statistically independent and that the properties of the nominal critical path can serve as a model for project length.

5. Using the mean and variance computed in Step 4 as parameters, apply the normal distribution to estimate probabilities associated with the length of the project. Assume that the Central Limit Theorem applies.

Appendix B. Three Building Blocks for the Systemic Error Model

1. A mathematical feature of variance estimates in PERT. The traditional PERT approach relies on a specialized beta distribution as a model for the randomness in activity durations. Specifically, the method assumes that a beta distribution exists that satisfies two conditions: (1) $\mu = (a + 4m + b)/6$ and (2) $\sigma = (b - a)/6$ (see Appendix A for notation). Clark (1962) discusses the reasoning behind the triplet method and clarifies that it was meant to serve merely as an approximation. However, that approximation was also supposed to cover all reasonable cases that might arise in practice.

The beta distribution has four parameters, *min*, *max* (corresponding to *a* and *b*), and a pair of shape parameters, α and β. We can scale the distribution by setting *a* to 0 and *b* to 1. The scaled mean and standard deviation in the PERT model are: (1) $\mu = (4m + 1)/6$ and (2) $\sigma = 1/6$. In general, for any triplet (a, b, m), precise beta-distribution parameters that comply with the PERT conditions rarely exist (Grubbs, 1962), so we must settle for approximate results. (One exception is the symmetric case where setting $\alpha = \beta = 4$ provides a perfect fit. Grubbs identified two other exceptions.) Suppose that we are willing to tolerate a relative deviation of at most *d* between the mean, mode, or standard deviation estimated in the triplet method and the corresponding values of the approximating beta distribution. In other words, we look for parameters α and β for which the scaled mode, mean and standard deviation are all within *d* of the prescribed values and for which the beta distribution is as close as possible to the estimates of these three values. We can establish numerically that a tolerance of $d = 5\%$ approximately requires a scaled *mode* just below 0.05. The corresponding *cv* as calculated by PERT is thus no larger than 0.8333. Hence, adopting the triplet method leads us to a distribution with $cv \le 0.8333$, and the value is likely to be much lower if we require tighter tolerance levels. Furthermore, even if we were to ignore the quality of the fit completely, the triplet method implies $cv \le 1$, because after scaling we get $cv = 1/(1 + 4m)$. If $a/b > 0$ the *cv* is even lower; for instance, if $a = 1$ and $b = 4$ we get $cv < 1/2$ whereas if $a = 2$ and $b = 5$ we get $cv < 1/4$.

Typical practitioners are not aware of this technical limit on the location of the mode (*m*) relative to *min* and *max*, but they can't avoid underestimating σ when the true range between min and max exceeds 6σ. Therefore, we suspect that the triplet method is likely to lead to small *cv* values in practice as well as in theory. Systematic underestimation of variability would also be consistent with the observations of Tversky and Kahneman (1974). As our field data demonstrate, however, small *cv* values may well be inappropriate.

2. ***The lognormal distribution.*** By definition, the lognormal random variable is the exponent of a normal random variable with mean *m* and variance s^2. (In other words, the logarithm of the lognormal is normal.) Let *X* denote a lognormal random variable with mean μ and variance σ^2. To evaluate *m* and *s*, let $cv = \sigma/\mu$. Then

$$s^2 = \ln(1 + cv^2) \quad \text{and} \quad m = \ln\mu - s^2/2 \qquad \text{(A1)}$$

Baker and Trietsch (2009) present four arguments why the lognormal random variable is especially attractive for modeling stochastic processing times: (1) it is strictly positive,

(2) its *cv* is not restricted, (3) it is convenient for approximating sums of positive random variables, and (4) it is suitable for modeling the relationship between capacity and processing time. Among the distributions typically employed to model processing times, only the lognormal exhibits all four traits. We next elaborate on the last two points.

The lognormal distribution is convenient for representing sums of positive random variables (convolutions) because it satisfies the *lognormal central limit theorem* (Mazmanyan, Ohanian and Trietsch, 2008): as *n* grows large, the sum of *n* independent positive random variables tends to lognormal. (This property holds subject to mild regularity conditions similar to those that apply to the classical central limit theorem.) Practitioners often cite the classical central limit theorem as justification for using the normal distribution as a model for the convolution of a small number of random variables, but the normal may lead to negative values. This is especially true when the *cv* values are high, which our data show to be an important case. Instead, we use the lognormal central limit theorem as a basis for approximate convolutions of positive random variables. An important special case has components that are also lognormal. Let Y denote the sum of n independent lognormal random variables with parameters m_j and s_j^2. To approximate the distribution of Y by another lognormal distribution, we first evaluate the components' means and variances from the following formulas:

$$\mu_j = \exp(m_j + s_j^2/2) \qquad \sigma_j^2 = \mu_j^2[\exp(s_j^2) - 1] \qquad \text{(A2)}$$

Given these parameters, we can add means to obtain μ_Y and add the variances to obtain σ_Y^2. Then we can calculate m_Y and s_Y from (A1). Such calculations are easy to program and should cause no difficulty in practice.

The reciprocal of the lognormal distribution is also lognormal (with the same *cv*). In some project environments, the processing time is inversely proportional to the effective capacity allocated to an activity. If the total work requirement is a constant but the effective capacity is lognormal, then the processing time will be lognormal. Indeed, the ratio of two lognormal random variables is also lognormal, so even if the total work requirement is lognormal, the ratio of work requirement to capacity—in other words, the processing time—will be lognormal as well.

3. A Model for the Parkinson effect. Baker and Trietsch (2009) introduced another distribution for processing times, the [pure] *Parkinson distribution*. Parkinson's Law (Parkinson, 1957), states that "work expands so as to fill the time available for its completion." In that spirit, suppose that work is allotted q units of time, but it really requires X, where X is a random variable. Then the activity duration we observe, Y, is given by

$$Y = \max\{q, X\}$$

and we say that Y has a pure Parkinson distribution. The relevance of Parkinson's Law to projects has been documented (e.g., Schonberger, 1983; Gutierrez and Kouvelis, 1991), but the Parkinson distribution has apparently not been defined or validated previously. In this research we apply the pure Parkinson distribution to projects that exhibited a tendency for estimates to serve as a floor for actual realizations. In those cases, we model X as lognormal.

If some activities are reported early but a high proportion is recorded precisely on time, we can employ the Parkinson distribution in its general form. Let p_P denote the probability that an early activity is falsely recorded as precisely on time. We assume that this probability applies to each early activity independently. That is, early activities are recorded correctly with a probability of $(1-p_P)$, and precisely on time otherwise; tardy activities are always recorded correctly. Notice that for $p_P = 1$ we obtain the pure Parkinson distribution whereas for $p_P = 0$ we obtain a conventional distribution. The Parkinson distribution can be defined for any internal random variable, but here we assume the lognormal.

To estimate parameters from history, we adapt Equation (1) for correctly recorded early activities. Recall that for tardy activities we defined $z_{jk} = \Phi^{-1}((j-0.5)/n_k)$, and this remains valid for strictly tardy activities when we estimate parameters for the Parkinson distribution. But because a proportion of early activities must be ignored for estimation purposes, we must replace n_k by $n_k(1-p_P)$ for the strictly early activities; that is, for correctly recorded early activities we use $z_{jk} = \Phi^{-1}((j-0.5)/[n_k(1-p_P)])$.

To simulate the Parkinson, we estimate the necessary parameters, including p_P, from history (as is also the case for the pure Parkinson distribution). These parameters are used to simulate initial processing time values. Then, each early activity is subjected to a side lottery to decide how to record it. When the data set is not rich—as was the case here—we can use empirical distributions to represent history. For instance, working in logarithmic space, if the mean bias, slope and p_P parameters of project k are denoted by m_k, s_k, and p_{Pk}, and we wish to simulate for this project, then we use the parameters m_i, s_i, and p_{Pi} in a proportion of $[n_i / \sum_{i \neq k} n_i]$ of the rows of our sample. By adopting the parameters of each project $i \neq k$ in the "history" together, we automatically take account of any empirical correlation between parameters.