

MODELING UNCERTAINTY IN PROJECT SCHEDULING

Patrick Leach
Decision Strategies, Inc.
3902 Gallaher Court
Missouri City, TX 77459 USA

ABSTRACT

Projects frequently run late and over budget because of two probabilistic phenomena: the statistics of parallel tasks and the effect of the central limit theorem on right-skewed probability distributions. These factors combine to virtually assure failure to meet the predicted timeline and budget. The best way to avoid these problems is to model project schedules stochastically. Even so, the P10 – P90¹ uncertainty range for the total timeline is often underestimated due to failure to capture dependencies between inputs. However, determining the appropriate correlation factor between each pair of tasks on a major project is practically impossible (and implementing them is sometimes literally impossible). One solution is to use a “global correlation factor.” Although scientifically impure, such a factor is a practical way to generate overall project schedule ranges that reflect reality. Even with this approach, cost overruns are further exacerbated by the MAIMS principle: Money Allocated Is Money Spent.

1 INTRODUCTION

Projects that take longer than expected and run over budget (sometimes way over budget) are endemic in the business world. Construction of new retail stores, upgrading of highway interchanges, drilling of deepwater oil wells – all are plagued by an apparent failure to meet the predicted timetable more often than not. It’s almost a cliché that any large-scale project will miss its target deadline and exceed its budget.

As a result, fudging is common. An expanded time estimate is entered for most major tasks just in case problems are encountered, and the all-purpose “contingency” item makes its appearance on many a Gantt chart. There is a reason for this: lots of things can go wrong on a major project. Weather alone is a wild card that affects almost all outdoor projects. Yet even with contingency tacked on and major tasks fudged to the high side, many projects still exceed the predicted time (and budget, which generally goes hand-in-hand with time). Moreover, we usually have no way of predicting how likely it is that a given project will meet its milestones and/or overall timetable.

Why should this be? Are project managers and crews really this incompetent? Does Murphy’s Law have something to do with it?

The short answer is “No” (although I wouldn’t rule out Murphy’s Law entirely). Project teams are generally staffed with highly trained professionals and skilled laborers who know their jobs as well as anybody. They’re just fighting a couple of forces against which they will very rarely win.

2 NEAR-CRITICAL-PATH ACTIVITIES

Most major projects run a number of tasks in parallel. Often, a subsequent task cannot begin until all of these parallel tasks have been completed. As a simple example, imagine building a house and having plumbers running pipes, electricians running wires, and air conditioning specialists running duct work in the walls. Theoretically, all of these workers can be active on the house simultaneously (especially if it’s a big house, so they can work on different parts at different times and not get in each other’s way). However, the dry wall cannot be installed until all of these tasks have been completed.

In situations like this, the longest of the parallel tasks becomes part of the critical path for the project (the string of tasks that determines the overall time it will take to complete the project). Some amount of slippage on non-critical-path tasks can be tolerated, but any slippage on critical-path tasks directly results in slippage on the project as a whole.

Now consider Figure 1 – a portion of a Gantt chart for a major project. Five tasks are running in parallel and the longest of these – task C – is on the critical path. The other four tasks (B, D, E, and F) are all nearly as long as C, but not quite.

¹ For the purposes of this paper, “PX” shall represent the Xth percentile of a probability distribution; i.e., there is a 10% probability that the actual value will be less than the P10, a 50% probability that it will be less than the P50, etc.

the length of each task is the estimated P50 in each case, what is the probability that all five tasks will be completed in the time allotted?

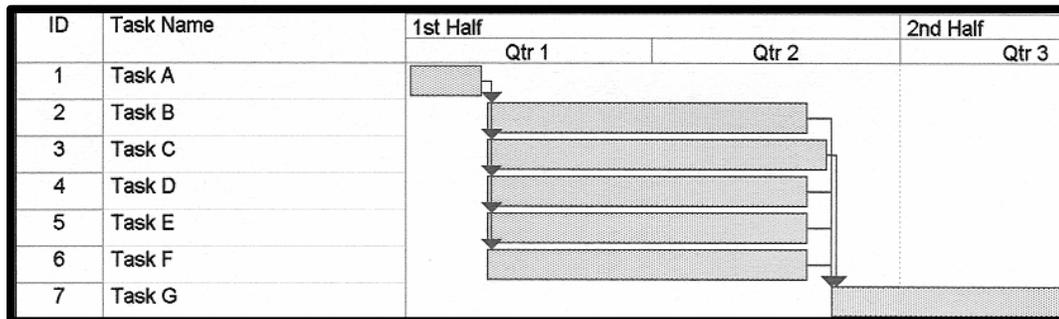


Figure 1: Gantt chart with parallel tasks

Let’s break it down and start with the easiest one: what is the probability that task C will be completed in the time allotted? Fifty percent, of course – we just said that the allocated time represents our best estimate of the P50. So half the time, task C will take less time than predicted, and half the time it will take more (for simplicity’s sake, tasks that are completed exactly on time will be considered to have been completed in less than the time allotted).

Now, what about tasks B, D, E, and F? In each case, the allotted time (which is the P50, remember) is only slightly less than the allotted time for task C. This means a little bit of slippage can be tolerated, but not much; if any one of these tasks ends up taking longer than task C, it replaces C on the critical path and delays the project.

So what is the probability that, say, task B will be completed in less time than the amount allocated for task C? We cannot tell the exact probability looking at this Gantt chart alone, but it is almost certainly only slightly higher than 50% (since B has a 50% chance of taking less than its allotted time, and its allotted time is only slightly less than that for task C). This would also be the case for tasks D, E, and F.

A subtle problem begins to reveal itself. With sequential tasks, delay on one task can sometimes be compensated for by speeding up a later task. Not so with parallel tasks: in order for the overall project to avoid delay, all five of these tasks must be completed in less than the time allotted for task C. What is the probability of this happening?

Assuming the tasks are independent of each other – i.e., delay on one says nothing about the probability of delay on any other – the probability of all five tasks being completed within the time allocated for task C is simply the product of the five individual probabilities. For tasks B, D, E, and F, this probability is slightly higher than 50% - let’s say 55%. For task C, the probability is 50% exactly. $50\% \times 55\% \times 55\% \times 55\% \times 55\% = 4.6\%$.

In other words, this portion of the project has less than a 5% chance of meeting its deadline, despite the fact that each individual task has a 50% chance of coming in on time. The project team is probably doomed to failure before the first two pieces of wood have been nailed together.

This problem is exacerbated by most project managers’ tendency to focus almost exclusively on critical-path tasks (in this case, Task C). We’ve said that the probability that Task B, for instance, will take less than the time allotted for Task C is about 55%; conversely, the probability that Task B will exceed the time allotted for Task C is about 45%. However, this assumes that Task B is given the proper attention. If Task B goes largely ignored, this probability can easily jump from 45% to 50% or more – a higher probability than that for Task C.

Even if this doesn’t happen – even if the project manager keeps her eye on all five tasks with equal concern – the probability is still far higher that one or more of the four non-critical tasks (B, D, E, and F) will end up exceeding the time allotted for Task C than it is for Task C exceeding its time. The former probability is about 91% ($1 - 0.55^4$); the latter probability is only 50%. This is simply because there are four of the near-critical tasks, and only one critical one; it’s far more likely that one (or more) of the four will have problems than it is that Task C, specifically, will have problems.

This is an admittedly extreme example; it’s not very common to have five parallel tasks that are so close in length. However, even with only two parallel tasks – let’s say B and C – the probability of both coming in without delay is only about 28% ($55\% \times 50\%$). Moreover, we’re only looking at one set of parallel tasks here; any real project will have dozens of sets of parallel tasks, some with five or more tasks, some with only two or three, some that are nearly uniform in length, some that are of vastly different lengths. The overall effect gets multiplied, and the result is near-certain delay to the overall project.

3 THE PORTFOLIO EFFECT AND RIGHT-SKEWED PROBABILITY DISTRIBUTIONS

Even without any parallel tasks, most projects are extremely unlikely to finish on time if a “most likely” or even a P50 estimate is used for the allotted time for each individual task and these are then summed to estimate the time needed for the overall project. The reason is an insidious interplay between the central limit theorem and asymmetric uncertainties.

There is uncertainty surrounding any estimate of how long a task will take. If it’s a short, simple task we’ve done a thousand times before, the uncertainty range may be so small that it can safely be ignored. With large, complex tasks, however, the uncertainty becomes very significant, indeed (especially if we’re depending on factors beyond our control, like weather).

For the vast majority of the tasks in most projects, this uncertainty distribution is skewed to the right – that is, the distribution will show a much longer tail extending off to the right than it will show on the left (see Figure 2). This is because in most cases, if everything goes well, the Gods smile on us, and our good-luck charms are in high gear, we might finish a task, say, 30% ahead of schedule; but if a gale hits, key parts never show up at the work site, and everything goes to hell in a hand basket, we can easily find ourselves going over time by 200%, 300%, or more. More basically, there is simply a physical limit to how quickly a task can be done; there’s no upper limit to how long it might take. Thus, the range of possible values for the time to complete this task runs from only a little bit below our predicted value to way above our predicted value. This is what the probability distribution in Figure 2 demonstrates; the most likely time needed for this task is in the range of 8 to 16 weeks, and it could be as low as 5 or 6 weeks – but it could also be as high as 40 or 45 weeks.

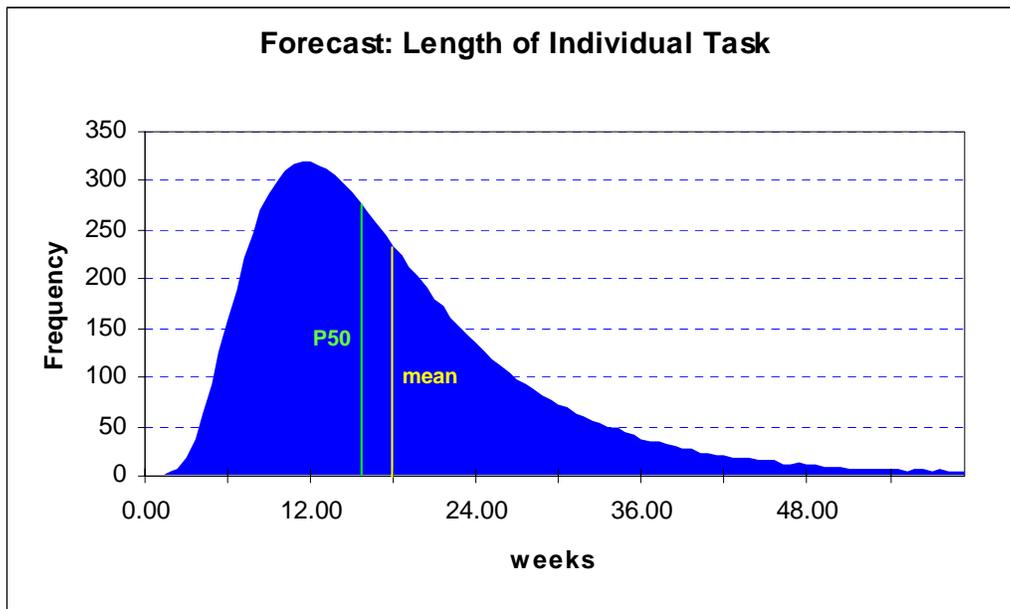


Figure 2: Probability distribution for length of an individual task

So why should right-skewed distributions spell doom for projects? The answer can be found by using a stochastic simulator to sum a large number of such distributions.

Stochastic simulation (often called Monte Carlo simulation) allows you to capture and understand the uncertainty inherent in your project. Instead of entering a best guess for the time needed for each task, you enter a range of possible values and assign a probability distribution to that range (like the one in Figure 2). When the stochastic simulation is started, the computer randomly selects a time value for each task – honoring the range and probability distribution for each one – and calculates the overall time needed for the project. It stores this value. Then the simulator goes through the whole process again, randomly selecting new lengths for each task, summing them up, getting a new value for the overall project time, and storing it.

It repeats the process however many times you like – at least a few hundred, often a few thousand – and ultimately ends up with a long list of possible Overall Time Needed values. The computer then rank orders this list from smallest to largest, calculates statistics based on this list, and displays a cumulative probability chart of the values – i.e., a probability distribu-

tion for the amount of time the overall project is likely to take. If you've set up your Gantt chart software to estimate costs based on project time, you'll get a probability distribution for overall cost, too.

So let's go back to the probability distribution in Figure 2. For simplicity's sake, we'll look at a project with twenty such tasks (each with an identical uncertainty distribution) running sequentially – i.e., no parallel tasks (Figure 3). Let's also assume that if we were estimating the time for this project deterministically (the way it's usually done, with single values for each input rather than a probability distribution), the “best estimate” that would be entered into the Gantt chart for each task is equal to the P50 of that task's uncertainty distribution (in this case, 16 weeks). And there's one final assumption: the tasks are independent of each other, or nearly so – i.e., how long one task takes has no effect on how long other tasks will take.

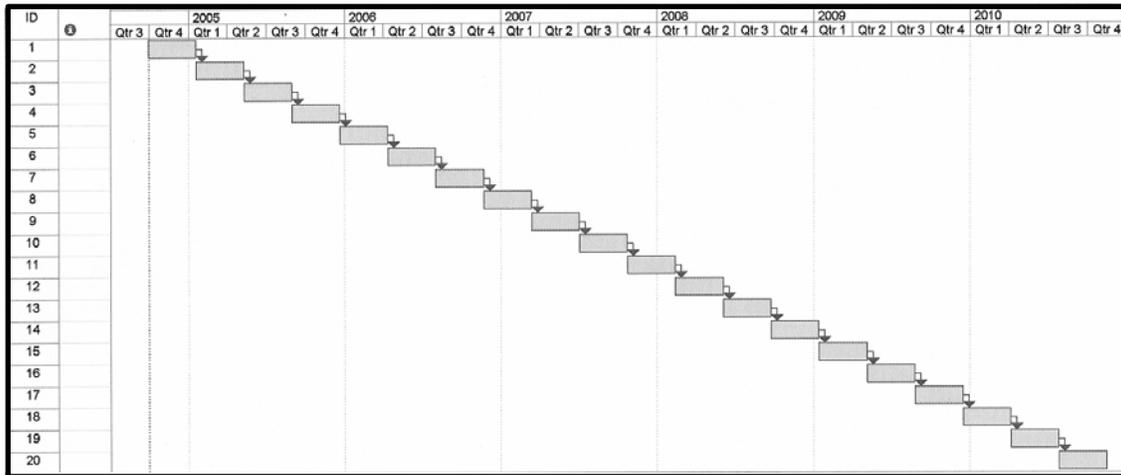


Figure 3: Gantt chart of twenty sequential tasks

If we run a stochastic simulation on this project, the output cumulative distribution curve looks like Figure 4. What does this tell us about the probability that we'll complete this project in less time than the deterministically predicted number of weeks? Where does the sum of all the P50 estimates – which would be our “best guess” estimate – fall on this curve?

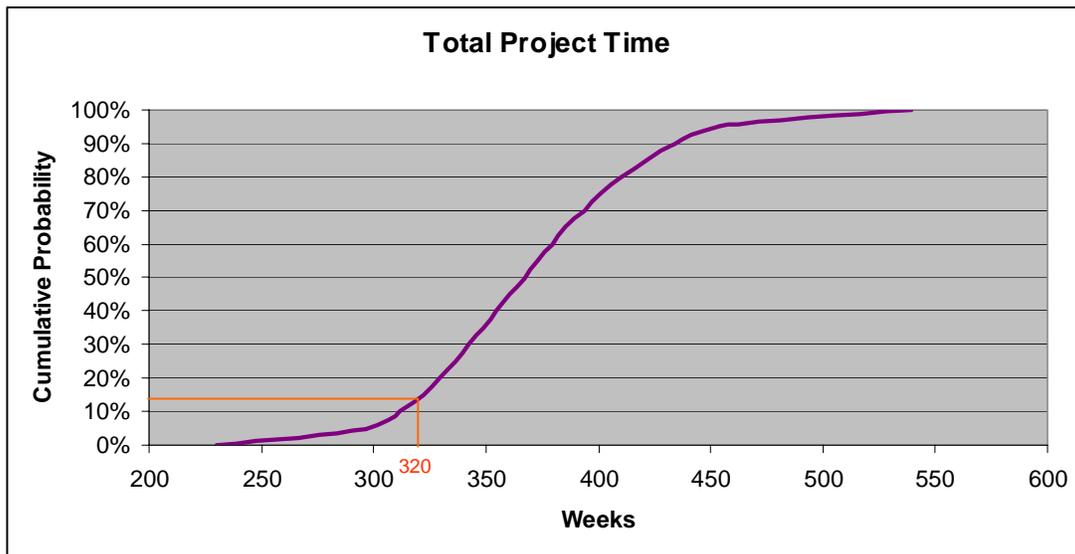


Figure 4: Total project time cumulative probability distribution, 20 tasks

The sum of the P50s is 320 weeks, which plots on the curve with a probability of 14%. In other words, if we use estimates for individual tasks that we think we have a 50/50 chance of achieving in each case, our overall probability of achieving this timetable is only 14%. We are highly likely to miss our deadline.

This happens because of two facts:

1. If you sum a number of uncertain items stochastically, the relative range of the output distribution tends to tighten up around the sum of the means of the inputs (this is the portfolio effect).
2. The mean of a right-skewed distribution is larger than the P50 (see Figure 2 again).

What does this mean in plain English? Better yet, let's look at a picture. In Figure 5, we start with a single task, and then add more and more tasks to the project and watch what happens to the overall time needed. These plots have been normalized so they can be compared with each other directly; the x-axis shows the average time taken for each task, rather than the total time for the whole project. Note that as more tasks are added, the curve becomes steeper – i.e., the range of probable values for average time per task becomes narrower. This is the central limit theorem, or the portfolio effect; this is why mutual funds are less volatile than individual stocks.

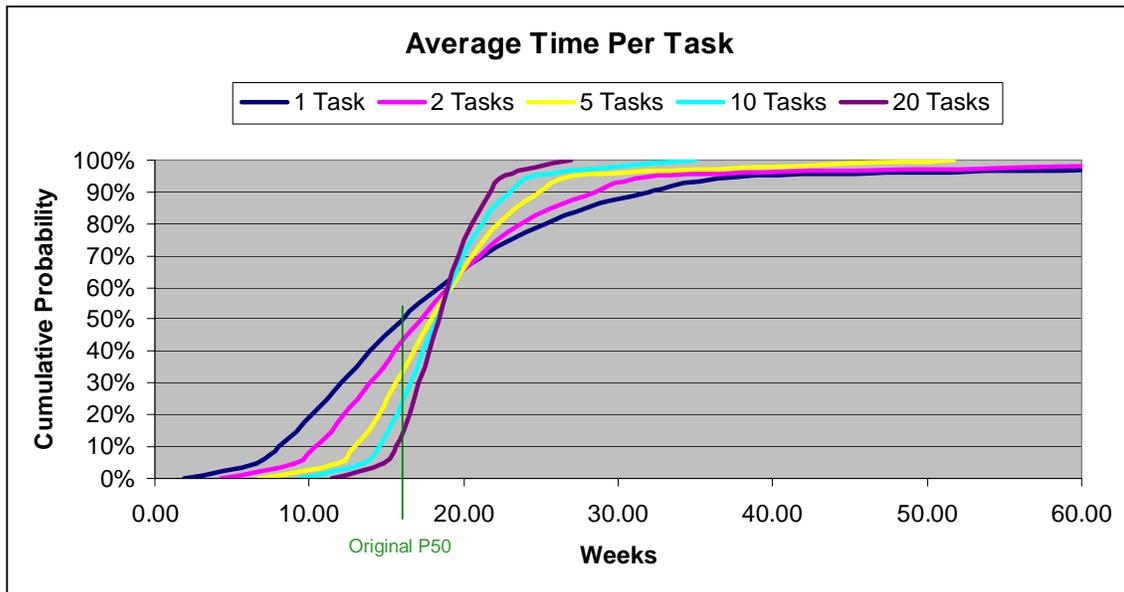


Figure 5: Average time per task, different numbers of tasks summed

But note also that the pivot point about which the curves become steeper is the mean, not the original P50. As a result, as more and more tasks are added, the probability that the average time taken per task will be less than or equal to the original P50 drops from 50% to 44%, to 35%, to 25%, to 14%. Eventually, the probability of averaging the original P50 value becomes negligible. We're almost certain to exceed our predicted timetable.

This example uses twenty identical tasks in order to simplify the demonstration, but the phenomenon holds true regardless of whether the tasks are of varying length. As long as the uncertainty distributions are right-skewed and the tasks are at least partially independent of each other, as more tasks are added, the probability of staying on the deterministic schedule shrinks to the point of becoming extremely unlikely.

4 CORRELATIONS: THE GOOD NEWS AND THE BAD NEWS

Fortunately, there's an unrealistic assumption in Figure 5; in a real case like this, you would probably have higher than a 14% chance to meet your deterministic estimate. The unrealistic assumption is that all twenty of the tasks in the schedule are completely independent of each other. On a typical project, one crew performs multiple tasks, and that crew may be better (or worse) than the average crew; the equipment they use on these tasks may be old and obsolete or new and efficient; the weather for the tasks on one day is likely to be similar to the weather on the following day. Some degree of dependency exists between a large number of the tasks on a project, and this dependency tends to flatten the Total Project Time curve.

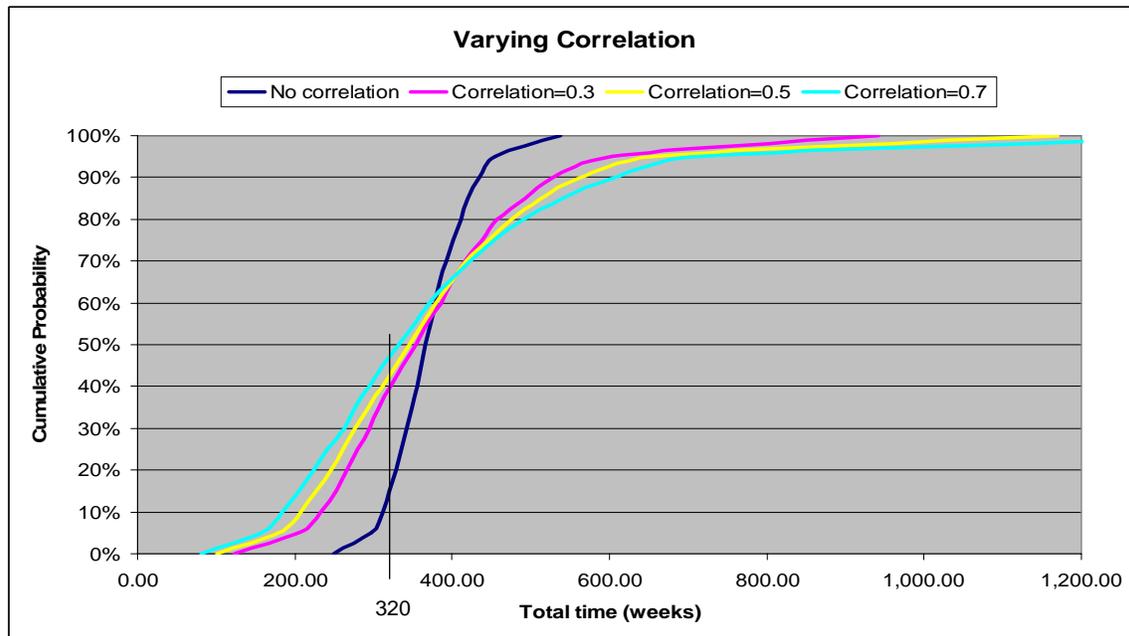


Figure 6: The Effect of Dependency

Figure 6 shows this effect. Correlations of various strengths have been applied between the distributions for the individual tasks in order to reflect the underlying dependencies between these tasks. The stronger the correlation, the flatter the curve – and the higher the probability of achieving the P50 sum of 320 weeks. That’s the good news.

The bad news is twofold. First, although your probability of meeting your deterministic target has increased, the overall uncertainty regarding the amount of time this project will take has increased dramatically. With no correlation between the individual tasks (the dark blue curve), you can be 90% certain of completing the project in less than 435 weeks. But with a correlation of 0.7 applied between all of the tasks, your 90% confidence figure grows to 609 weeks. This is not the type of news that managers welcome.

However, it’s the truth. Summing a large number of tasks stochastically without considering the dependencies between them invariably yields a Total Time curve that is far too steep, and has far too narrow an uncertainty range. A correlation factor must be applied.

This brings the second piece of bad news: applying correlations between dozens of interdependent tasks is nearly impossible to do. Even if you could somehow know what the appropriate correlation factor was between each pair of tasks (which is unlikely), applying them would be daunting at best. Our simple schedule with twenty tasks would have 190 correlations to apply between pairs of tasks; a project with 100 tasks would need 4950 correlations.

Fortunately, there’s a relatively easy (albeit scientifically impure) way around this. Use a global correlation factor. The easiest way to do this involves the use of a dummy variable, and takes advantage of the fact that if A is correlated to B, and A is also correlated to C, B and C will usually show a correlation. If you’re using Crystal Ball to run the stochastic simulation, and if the A-B correlation factor and the A-C correlation factor have the same value, the correlation factor between B and C will be the square of that value (because of the way Crystal Ball handles correlations).

So provided we’re content to have the same correlation applied between every pair of tasks in the project, the problem is simplified immensely in Crystal Ball. Just create a dummy variable (I usually use a uniform distribution between 0 and 1 for simplicity’s sake), and correlate every task in the project to this variable, using a correlation factor equal to the square root of the correlation you desire to have between each individual task. For example, if we want a correlation factor of 0.5 between all of the tasks in our project, we create a dummy variable and correlate each of our real tasks to the dummy with a factor of 0.707 (the square root of 0.5). This requires 20 correlations instead of 190 – far easier on the model and the modeler.

If applying one correlation factor across all of the tasks in a project is too crude, you can always apply several factors at different levels – say, one weak correlation across all tasks, and then a stronger factor between tasks that are in the same sub-group, or will be performed by the same sub-team (Kujawski and Alvaro 3).

5 THE MAIMS EFFECT

Even with stochastic scheduling, project costs are affected by an additional factor: the dreaded MAIMS effect. “Money Allocated Is Money Spent.” This refers to the strong tendency of project teams (and sub-teams) to spend whatever money is allocated to them, regardless of the amount. Even when things go well, at the end of the day it is rare for a team of people to give back a significant portion of whatever money was allocated to them (Kujawski and Alvaro 2). They may choose higher quality materials; they may double- and triple-check some items; they may decide to ship parts overnight to increase the probability of an early finish to the project. But they will find a use for whatever money they have been given.

Figures 7 and 8 show why this is a problem. Suppose you’ve been a conscientious project manager and have modeled your schedule and costs with Monte Carlo simulation. The cumulative probability curve for the cost of some group of sub-tasks in your project – call this Sub-Group A – is shown in Figure 7, along with the mean cost (the expected value).

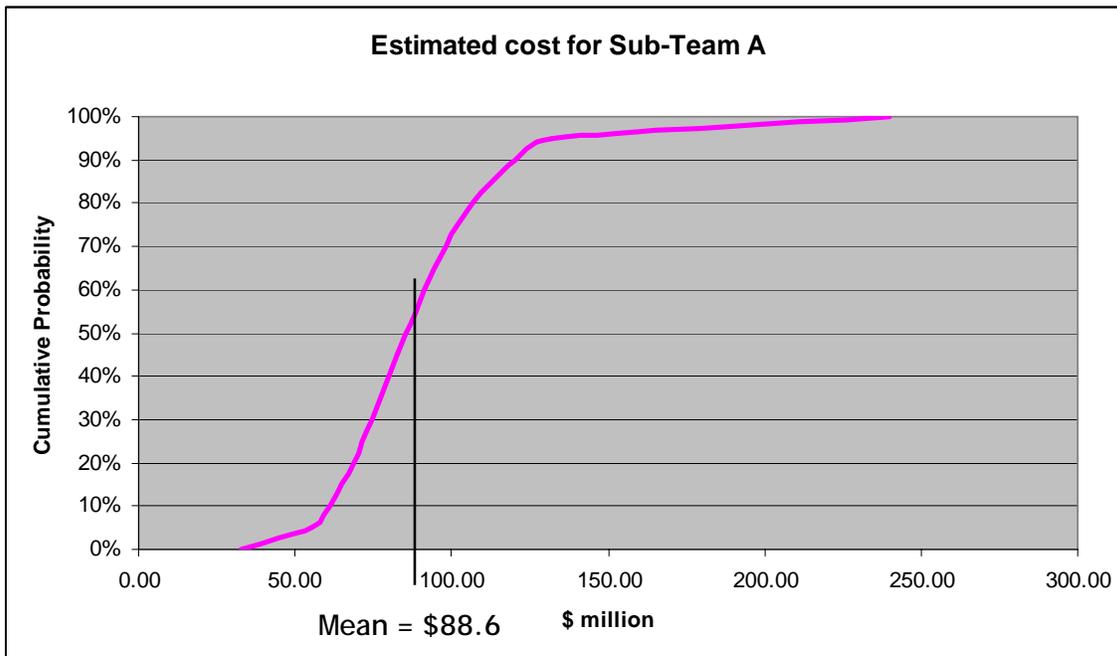


Figure 7: Cumulative Cost Curve for Sub-Team A

Say you decide to allocate \$80 million to the team (the P40 amount). You fully realize that there’s a 60% probability that they will exceed this amount, but that’s okay – you are holding a contingency fund for the overall project, and those sub-teams that need extra funds will be able to come to you for additional capital.

But remember the MAIMS effect: Money Allocated Is Money Spent. If you allocate \$80 million to this team, you can essentially forget about the probability that the team will spend less than that. In those instances in which fortune smiles on them and they would be on the lower part of the curve, they’ll find a way to spend something close to the full amount you allocated (otherwise, you probably won’t give them as much next time!). This truncates the curve on the downside at the allocated amount (Figure 8).

However, look what happens to the mean in this case: it increases from \$88.6 million to \$93.8 million. *The very act of allocating funds has decreased the probability that you will stay within your budget.* This is more than just an ironic twist. It’s a very real characteristic of the dynamic, interactive nature of estimating costs, budgeting, and allocating funds (and if you’re a contractor, bidding on work).

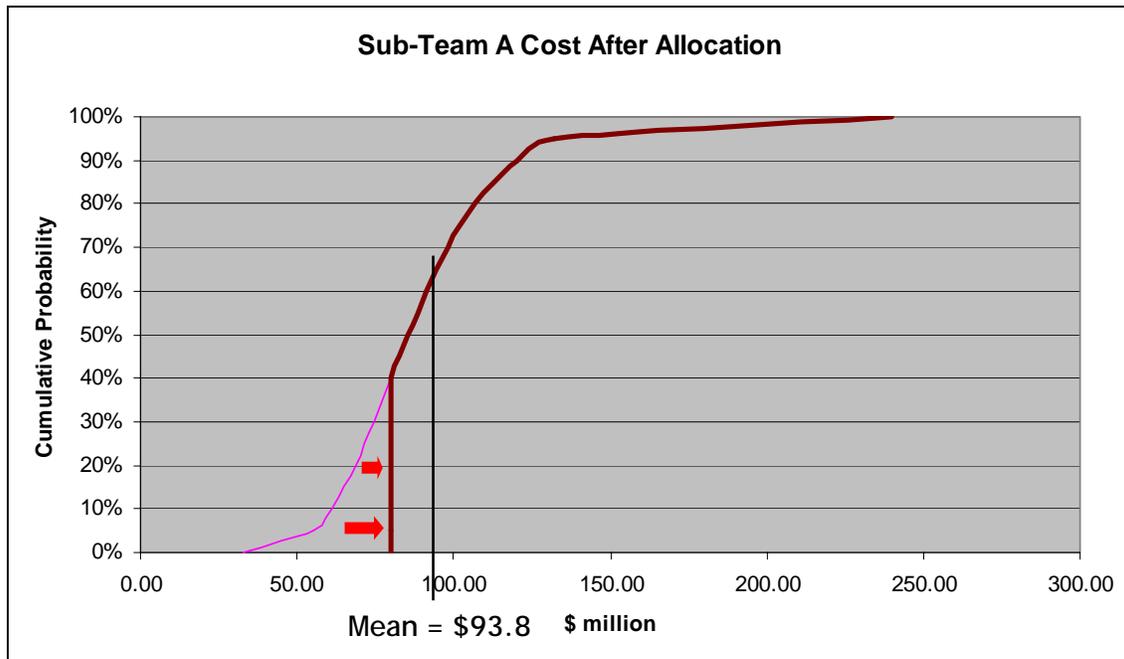


Figure 8: Truncated Cost Curve After Funds Are Allocated

There is no magical way to eliminate the MAIMS effect, but the paper by Kujawski and Alvaro in the bibliography describes a good approach to understanding and managing it. And again, you cannot even begin to cope with these issues if you are creating your project schedules deterministically.

6 CONCLUSION

Deterministic scheduling for major projects simply isn't good enough. Two phenomena – the effect of parallel tasks with nearly the same length and the central limit theorem as applied to right-skewed distributions – combine to virtually ensure failure to meet a deterministic timetable. We've examined each phenomenon individually here, but these effects combine and multiply on large, complex projects to the point where it is not uncommon to have less than a 1% chance of meeting the deterministic forecast for the overall project. The MAIMS effect then compounds the difficulties associated with cost management on such projects.

Historically, companies have either fudged the numbers upward or included contingency items in order to accommodate the possibility that unforeseen problems will delay the project. This is better than nothing, but not by much. We are still left with no idea about the *probabilities* that we'll achieve our bonus milestones, or how likely it is that we'll finish the new factory in time to start production before the fall season, or whether we'll avoid contract penalties for failing to meet deadlines. In order to manage our projects competently, we need stochastic scheduling.

With stochastic scheduling, curves like the one in Figure 4 give us far more realistic estimates of overall project length, how much it's likely to cost, and the probability that we will meet our deadlines. Schedules are created and analyzed using Monte Carlo simulation, rather than simply summing the best estimates for each task. Further, modern stochastic software enables us to identify key tasks – those that are most likely to cause delay and/or cost overruns, whether on the critical path or not – so we can focus precious resources on ensuring that those tasks don't slip.

Anyone who is serious about realistically forecasting project schedules, anticipating potential trouble spots, and taking action to mitigate against likely problems – in other words, truly *managing* major projects, rather than just monitoring them – should be using Monte Carlo simulation software to plan and analyze projects stochastically. It's the best way to avoid "late and over budget" syndrome.

REFERENCE

Kujawski, Edouard, and Alvaro, Mariana L., 2004. *Quantifying the Effects of Budget Management on Project Cost and Success*. Presented at the Conference on Systems Engineering Research, University of Southern California, Los Angeles, 15-16 April, 2004.

BIOGRAPHY

Pat Leach has 25 years' international experience in the energy industry. As a consultant, research lab portfolio manager and subsurface team leader, he has been involved in multiple aspects of the upstream oil and gas business around the world. He has conducted numerous customized workshops in decision-focused risk management, value-of-information techniques, Monte Carlo simulation using Crystal Ball, and decision analysis. While working for a large oil and gas company, Pat developed a process for integrating probabilistic analyses with deterministic reserves estimates and then implemented the process worldwide through hands-on training and workshops. He has also served as an outside technical expert during a reserves peer review at a major oil company.

Pat was graduated from the University of Rochester with a B. Sc. in Geophysics, and has an MBA from the University of Houston Executive Program. He is a member of the American Association of Petroleum Geologists, the Society of Petroleum Engineers, and the Society of Exploration Geophysicists, as well as the honor societies Beta Gamma Sigma, Phi Beta Kappa, and Tau Beta Pi. He has authored a book entitled *Why Can't You Just Give Me a Number? An executive's guide to using the dark arts of probability theory, simulation, and other statistical voodoo to make better decisions*. It is due to be published later this year. Pat joined Decision Strategies in 2004, and can be reached at peleach@decisionstrategies.com, or by phone at (281) 778-7908.