Deterministic versus Probabilistic Schedule

There are two general types of schedules in the planning domain.

1. Deterministic schedules - are networks of tasks connected to each other with dependencies that describe the work to be performed, that work's duration and the planned completion of the project.
   * Each task has a planned duration
   * Each task has a predecessor and a successor. The only task that should not have a predecessor is the start of the project and the only task without a successor is the completion of the project. This forms a closed network with no widows or orphans
   * The longest path through the network is the critical path
   * The total duration of the project is a fixed value - it is deterministic
2. Probabilistic schedules - are networks with all the elements of a deterministic plan, but the durations of the tasks are random variables
   * The duration are not random, but they are random variables drawn from a probability distribution.
   * Three point estimates "can" be used to describe these random variables. It is not necessary to use the three point estimate approach, but it is an easy starting point
   * The total duration of the project is a random number

With that simple definition, comes several problems:

1. How to capture the three point estimates?
2. What probability density function should be used to model the values in between the three point estimates?
3. What values do the three point estimates actually represent?

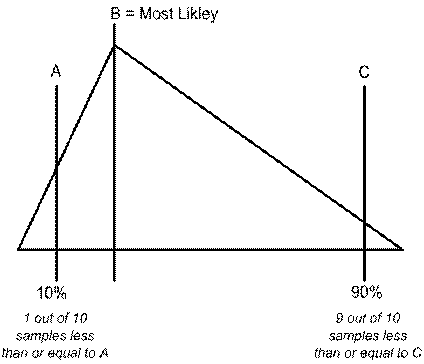
**What Values do the Three Point Estimates Represent?**

This is a critical question. The first naive approach is to say they are the "best case" and "worst case" values. This means they are the 0% and 100% end points of a probability distribution. This forces the probability distribution to be from the family of finite distributions. This is not really the problem. The problem is how these estimates are gathered.

If you ask some "tell the worst case estimate" they will bias this estimate by a personal risk avoidance factor. Field research has shown this is the case. A better approach works like this.

1. Tell me the Most Likely duration of this task. This means that if I perform this same work in the same project 10 times, this is the duration I will see "most often." It is the Mode of all the possible durations for the task. Notice this is NOT the Mean duration. That is the average duration. It is the "most likely" duration.
2. Tell me the duration of the task that will observed 1 time out of 10 if you performed this work 10 times. This is the 10% estimate. This says *for 10 executions of this task, 1 out of those 10 times I'll see this task complete in this duration or less.*
3. Tell me duration of the of the task that will be observed 9 times out of 10 if you perform this work 10 times. This is the 90% estimate. This says *for 10 execution of this task, 9 out of 10 times the task duration will be this value or less*.

The picture below describes these cases for a Triangle distribution. The really nice thing about Triangle distributions is the Mean and the Mode are the same. This is NOT the case for other distributions, like Beta and BetaPERT.

[](http://herdingcats.typepad.com/photos/uncategorized/triangle.gif)

With this information in hand (the three point estimates) the first inclination is to run the PERT tool in your favorite project management [application](http://herdingcats.typepad.com/my_weblog/2006/02/deterministic_v.html). DO NOT do this. PERT has several significant problems. The first is it assumes the random variables are distributed under the Beta distribution. This may or may not be the case. Second is that PERT assumes a Standard Deviation has a single formula along with the Mean having a standard formula. The Mean is A+4B+C. This is almost NEVER the case. It gets worse from there.

One very wicked problem is called *merge bias*. This is an unfavorable bias introduced into the duration calculation when parallel tasks are "joined" (usually at a milestone). A Google search for "PERT Merge Bias" will find all the information you can probably stomach for some time.

**Use Monte Carlo**

Once you have the three point estimates use a Monte Carlo Simulation (MCS) tool on the network. There are many of these Risk+, @Risk for Project, Crystal Ball are ones I've used. There are others but make sure you know how to "trust" them by running a null network - Lo/Most Likely/Hi all set to the same value for 10,000 iterations and see if there is any difference from the statistical network and the deterministic network.

Once you get some experience with MCS you can start assessing the schedule in terms of probabilistic confidence. You're now doing probabilistic scheduling.

BTW probabilistic network have little use for Critical Paths. Since the duration of a task is a random variable following a known probability distribution the critical path changes depending on the sampled value of this random variable. The MCS tools have ways of showing sensitivity, critically, and crucially of the various paths through the network. But a single Critical Path is not there.

This is about as detailed as it should get in a Blog level presentation. There is loads of detail that needs to be understood before probabilistic scheduling is used as a decision making tool. But this is the way large aerospace and construction jobs are planned. This is the way modern projects are planned. There are pit falls in this method, but the benefits far out way the little diversions along the way

Ref: <http://herdingcats.typepad.com/my_weblog/2006/02/deterministic_v.html>

Listed below are links to weblogs that reference [Deterministic versus Probabilistic Schedule](http://herdingcats.typepad.com/my_weblog/2006/02/deterministic_v.html):

### Comments

1

[Glen B. Alleman](http://www.niwotridge.com/) said...

Saso,  
The Risk+ tool uses 0% / 100% boundary points for the Triangle. The @Risk for Project tools uses the more proper 10% / 90% boundary points.   
The statistical correlation between the outputs of the two products is "close enough" for use in estimating the needed schedule margin or potential overrun risk.  
But this becomes more important as the number of activities grows and the correlation between the activities increases.

[August 16, 2008 at 12:40 PM](http://herdingcats.typepad.com/my_weblog/2006/02/deterministic_v.html#comment-6a00d8341ca4d953ef00e553eac7da8833)

[Sašo](http://rule-of-thumb.net/) said...

Glen, do you know how to get true A (min) and C (max) from 10th and 90th percentile estimates in triangular distribution?

[August 16, 2008 at 08:49 AM](http://herdingcats.typepad.com/my_weblog/2006/02/deterministic_v.html#comment-6a00d8341ca4d953ef00e5540629278834)

[Huybert](http://www.risk-modelling.com/) said...

Great subject for a blog-discussion. I have a two comments related to the original posting.

My first comment is related to preference of the Triangular distribution above a Pert distribution.   
1. The argument for the Triangular is that Mode = Mean is not true, unless you are dealing with a symmetrical distribution. However, in that case this would also hold for the Pert distribution.   
1. “PERT Merge Bias” is the difference between (1) the deterministic time that more than 1 parallel tasks take and (2) the probabilistic duration of these parallel tasks. The reason it is called “PERT Merge Bias” is that the Project Evaluation and Review Technique (PERT) used a Methods of Moments (MoM) analysis, and therefore had a bias (it didn’t take into account that the duration is longer than the maximum of the individual task durations). This bias is however not present when using PERT distribution in MC simulation – in fact, when using MC simulation we prevent this error from occurring!

There is however a reason why we normally prefer using the Pert distribution instead of the Triangular. The Triangular distribution has a mean equal to the average of its three parameters, i.e. (Min+Mode+Max)/3. The PERT distribution has a mean equal to (Min+4\*Mode+Max)/6, in other words it is the average of all three parameters but with four time the weighting on the Mode. In real-life situations, we are usually capable of giving a more confident guess at the Mode than the extreme values. For example, if you’re asked “What is the maximum cost of this project?” your mind can race with all the possibilities of what could go wrong, making it extremely difficult to give a definitive answer. The PERT distribution makes it less important to get the maximum (or minimum) value precisely right. You can read more about the difference between the Triangular and the Pert at our posting at<http://www.crystalball.com/support/risktips/risktip-3.html.>

My second comment is that Critical Path are still VERY relevant in a probabilistic schedule risk analysis. Within a probabilistic schedule risk analysis, one however looks at the Criticality Index (a measure that expresses how likely a task will be on the critical path). This Index can greatly help to focus on the most important individual tasks for the overall schedule risk.

[September 15, 2006 at 04:02 PM](http://herdingcats.typepad.com/my_weblog/2006/02/deterministic_v.html#comment-6a00d8341ca4d953ef00e55070a7ae8834)

Wander Gomes said...

Glen,

I found four post very helpful, but in a triangular distribution the mean (expected value)is really always the mode (most likely)? Isn't the mean given by the formula m = (a+b+c)/3?

[]'s  
Wander

[March 02, 2006 at 09:21 PM](http://herdingcats.typepad.com/my_weblog/2006/02/deterministic_v.html#comment-6a00d8341ca4d953ef00e5505cd7b28833)

[Glen B Alleman](http://www.niwotridge.com/) said...

Jack,

The Null test is to exercise the floating point number generator to assure that no "leaks" are occurring. You want to test that 10,000 sample can be drawn from the population that fit in the Delta function (it’s not a point) for the duration (or whatever other variable you're modeling).

Since the underlying engine uses floating point calculations there are potential instabilities in this library. This is where most of the hard work for MCS comes from - generating the random numbers in a repeatable manner.

10,000 assures convergence of the sample spaces variance.

[February 02, 2006 at 09:23 AM](http://herdingcats.typepad.com/my_weblog/2006/02/deterministic_v.html#comment-6a00d8341ca4d953ef00e5505cd99b8833)

[Jack](http://zo-d.com/blog/) said...

Glen,

If High = Low = Most Likely, then the probability density function is a single point. One sample is enough.

You say do this and run it 10000 times: "but make sure you know how to "trust" them by running a null network - Lo/Most Likely/Hi all set to the same value for 10,000 iterations"

I fail to see what this gives you over a couple of iterations unless you are concerned that there is some intermittent problem which might crop up occasionally after an extended number of trials. In that case why not pick a million or two?

Now I can see 10000 iterations for arriving at a combined PDF for the whole schedule, but as a smoke test for the software it just seems arbitrary.

[February 01, 2006 at 11:32 PM](http://herdingcats.typepad.com/my_weblog/2006/02/deterministic_v.html#comment-6a00d8341ca4d953ef00e55070a8a38834)

[Glen B. Alleman](http://www.niwotridge.com/) said...

Jack,

The null test shows that "nothing" produces "nothing." Our work-a-day run is usually 1,000 or 1,500. For production runs - submittal for proposal or for submittal to Contract Performance Report (Monthly CPR), the number of samples from the Monte Carlo distribution(s) for each task needs to "visit" all possible areas under the curve, bounded by the upper and lower limits of the distribution - usually the 10/90 limits.

You can look at the Probability Density Function as a "histogram" of the probability of returning a sample of that value from the population of all possible samples.

10,000 is a number that puts us past the boundary of having to ask - "did the MCS sampler (Latin Hypercube algorithm in most products - ours is @Risk for Project and Risk+) get all the near-end-point values to run against all the other near-end-point values for the durations of all the tasks.

If you run 10,000 passes for a schedule with 5,000 activities (a medium size for use) with a distribution for each activity you get a lot of sample 10,000 \* 5,000. This covers the sample space for all possible durations, each with the right probability of occurrence.

[February 01, 2006 at 07:07 PM](http://herdingcats.typepad.com/my_weblog/2006/02/deterministic_v.html#comment-6a00d8341ca4d953ef00e55070a65e8834)

[Jack Dahlgren](http://zo-d.com/blog/) said...

Glen,

Nice post.

The null test is an interesting one, but what is magic about the # 10,000? Wouldn't 3 be as effective? And if 10,000 are really required for some reason wouldn't 10,003 be even better? :-)

Second, there is definitely a possibility that several tasks are going to ALWAYS be in the critical path. That said, if a task turns up on the critical path for a significant fraction of iterations you definitely need to keep an eye on it and minimize it if possible.

For those who want to play around with a free (minimally featured) Monte Carlo simulator you can download mine here:  
<http://masamiki.com/project/blackjack.htm>

Enjoyed your post about how the variance should ideally converge as you get closer to the goal. If one is not seeing this, then it is a sign that something is wrong.

[February 01, 2006 at 06:42 PM](http://herdingcats.typepad.com/my_weblog/2006/02/deterministic_v.html#comment-6a00d8341ca4d953ef00e5505cd8998833)

Denis Haskin said...

Thanks for the detailed explanation. That's what I was expecting, but I didn't know the details.

[February 01, 2006 at 02:16 PM](http://herdingcats.typepad.com/my_weblog/2006/02/deterministic_v.html#comment-6a00d8341ca4d953ef00e55070a95e8834)