Project Security Using Analytical Time Evaluation Techniques

Maksim Goman
LIT Secure and Correct Systems Lab
Johannes Kepler University Linz
Linz, Austria
maksim.goman@jku.at

***Abstract*—Time planning of IT Projects is difficult due to a large number of uncertainties in the software development process. Mistakes in software development planning open ways to architectural, functional, and integrative deficiencies of the product and surrounding processes. Using a simple synthetic project, we compare three analytic techniques to evaluate project duration. These three techniques can work with uncertain input parameters, represent projects as stochastic activity networks, and use simple computations to approximate distributions of the start time and end time of all tasks. Most importantly, they address the problem of merge event bias. We observe that a method with most relaxed assumptions performs better than others in comparison with a simulated ideal solution.**

Keywords-project management; software development; project time estimation; merge event bias

I. INTRODUCTION

Time planning in software development projects has always been problematic due to uncertainty of required resources, time, specifications, and unreliable human estimates. A recent review on the development of evaluation techniques of project time is given in [1]–[3]. Modern software projects contain uncertainty in virtually all possible parameters, the project environment is usually unstable and changeable (including requirements specification), and there are no established tools to help project managers with time analysis. Simple techniques that can work with roughly estimated input data seem useful for this situation in industry.

Program evaluation and review technique (PERT) [4, p. 269] is always mentioned in the literature for practitioners as an analytical probabilistic method for project time analysis [2,3]. This technique searches for only one critical path (CP), and have very restrictive assumptions for applicability. Another way of reasoning is based on approximations of the effect of merge event bias (MEB) [4, p.296] in each node of the stochastic network. MEB shifts the mean of the time distributions (TD) to the right, i.e. to a longer duration.

Although the issue of MEB is not well communicated in today's literature, a practical technique should include MEB into consideration, because it is a major model factor of underestimation of project time [1]–[4] beyond estimating *uncertain parameters*. As we are working on a new analytical method for project time analysis for very uncertain projects, our objective is to evaluate performance of simple techniques which address MEB problem. We have found no recent traces of such a comparison of any analytical techniques for project time analysis. We have identified three analytical project evaluation techniques that can operate under high uncertainty of task duration estimates, and they include a MEB correction procedure. These are modified PNET algorithm [4], a technique for estimating the distribution of a stochastic project makespan by Cohen and Zwikael [5] (hereinafter C-Z method), and the Pessimistic project evaluation and review technique (PPERT) [3].

The PNET method is an extension to classical PERT technique based on estimates and initial task TD of classical PERT. As in PERT, the normal distribution is assumed for eventual path duration. It considers paths in the stochastic network of the project but introduces a MEB correction step. The task durations are not necessarily statistically independent. A heuristic is given to decrease the number of paths involved in computations. Then, they select a set of representative paths determined using a proposed linear correlation coefficient between pairs of project paths and a given threshold 0.5. If the correlation coefficient of two paths is less than 0.5, then the two paths are assumed independent and representative. Otherwise, the "longer" path is selected to represent both paths and the shorter path is removed from consideration. The representative paths are assumed statistically independent and hence having few or no common tasks. It is required to generate the list of paths (starting with the CP). The cumulative distribution function (CDF) of the project completion time is estimated as a product of CDFs of the representative paths.

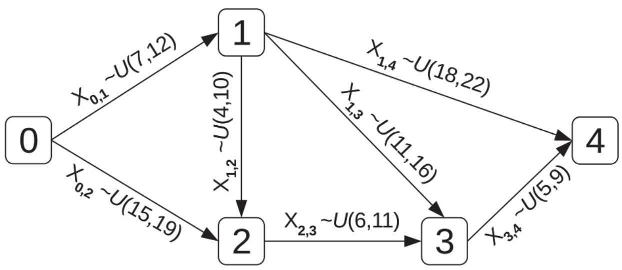
The C-Z method can work with diverse *symmetric* distribution types, and it approximates the CDF of project time in each network node using discretization, interpolation, and extrapolation. The main point is that the maximum distribution in nodes differs from the distribution of the longest expected time among the predecessor tasks. Assuming independence of predecessor tasks of a node, it is possible to estimate (bound) the CDF of the event in the node as a product of CDFs of end TD of its immediate predecessor tasks connecting the current node and preceding nodes. The TD of the immediate predecessor task is the TD of the preceding node plus the random duration of the predecessor task. Thus, this algorithm does not consider paths and needs only one traverse through the project network to compute all approximations of end TD of nodes. The meth adds quantiles of distributions of consecutive tasks, and multiplies quantiles of parallel collapsing tasks.

The method PPERT needs uniform distributions of task times as the input. Additionally to assumed task dependencies, other non-statistical dependencies (logical, technical, procedural, etc.) may be known as background information. Therefore, accepting the high uncertainty of the outcome of time of each task, a symmetric triangular distribution is used for the approximation of TD in nodes of the stochastic network as a MEB correction procedure. The chance constraint is used to compare two distributions to determine if MEB correction is required or the "smaller" path can be dropped, and the distribution of the longer "dominating" path can be set as the TD in the node. One traverse through the network is required. A simple correction of the upper bound of the final TD of the end node is required if MEB corrections were applied.

We evaluate the application of the methods using an artificial generic network taken from [2]. We will compare CDFs of the project duration obtained with the three methods and classical PERT to CDF of a simulated makespan. The network includes two paths of relatively equal probabilistic duration and generates MEB in three nodes.

II. Analysis of example project

The project network is given in Fig. 1 in activity-on-arc notation, and input data are given in Table I. Random time variables of nodes i are denoted as X_i, $i = 0,1,2,3,4$ and random time of tasks $X_{i,j}$,is indicated as $X_{i,j}$, where i and j are their begin and end nodes. Their respective CDF are F_X_i and $F\_X_{i,j}$. For paths, the indices of the origin, intermediary nodes, and the end node are used with hyphens, e.g. random time of path X_{0-1-3}.



The network of the example project

Initial time estimates of tasks are uniformly distributed. These data are used directly in C-Z and PPERT techniques. Beta distributions for classical PERT and, respectively, PNET algorithm, were obtained with simple approximations from PERT theory: the mean time is *(o+4m+p)/6* and standard deviation is *[p-o]/6* [5], where optimistic (o), pessimistic (p) and most likely (m) values coincide with the minimum, maximum and mean parameters of given uniform distributions.

We are searching for an approximated maximum time distribution in nodes 2, 3, and 4. The distribution in node 4 is also project makespan.

TABLE I. Input data

Task	Min	Max	Mean	Variance	Std. dev.
$X_{0,1}$	7	12	9.5	0.6944	0.8333
$X_{0,2}$	**15**	**19**	**17**	**0.4444**	**0.6667**
$X_{1,2}$	4	10	7	1	1
$X_{2,3}$	**6**	**11**	**8.5**	**0.6944**	**0.8333**
$X_{1,3}$	11	16	13.5	0.6944	0.8333
$X_{1,4}$	18	22	20	0.4444	0.6667
$X_{3,4}$	**5**	**9**	**9**	**0.4444**	**0.6667**

Arcs related to CP of the classical PERT are marked in bold in Table I. The solution of the classical PERT for all nodes is given in Table II. The outcome is obvious. The PNET solution is shown in Table III, the outcome of the C-Z method is given in Table IV, and PPERT solution is in Table V.

A. PNET solution

As paths 0-2 and 0-1-2 are not correlated using the correlation coefficient, the PNET solution for node 2 is trivial: CDF $F\_X_2 \sim N$(mean, std. dev.) $= N(17,\ 0.6667) \cdot N(16.5,\ 1.3017)$. Solutions for nodes 3 and 4 are presented in Table III. Paths from the origin to node 3, respectively, node 4, are ranked in descending order of the mean path duration in the left part of the table. In the right part of the table are correlation coefficients for each pair of significant paths. Following the given heuristic rule for node 3, path P13 is not considered as its mean differs from the first (critical) path by more than twice the larger of its standard deviations, i.e. $25.5 - 23 = 2.5 > 2.3570 = 1.1785 \cdot 2$. In the same way, for node 4, path P4 is not considered as its mean differs from the first (critical) path by more than twice the larger of its standard deviations, i.e. $32.5 - 29.5 = 3 > 2.5166 = 1.2583 \cdot 2$. The respective correlation coefficients appear in italics. For node 4, the correlation coefficient between paths P1 and P2 is 0.5377>0.5, therefore the path P1 represents both paths and the path P2 is removed. PNET solution for node 3 is $F\_X_3 \sim N(25.5,\ 1.0672) \cdot N(25,\ 1.5456)$, and for node 4 is $F\_X_4 \sim N(32.5,\ 1.2583) \cdot N(30,\ 1.3540)$.

TABLE II. PERT solution

Node	Path	Distribution
1	0-1	N (9.5, 0.8333)
2	0-2	N (17, 0.6667)
3	0-2-3	N (25.5, 1.0672)
4	0-2-3-4	N (32.5, 1.2583)

B. C-Z solution

According to the algorithm, we begin with tasks $X_{0,1}$ and $X_{0,2}$ as those without predecessors and set their start-time to zero. The maximum time in node 1, and start time of

Table III. PNET SOLUTION

Node 3									
Path id.	***Path***	***Mean***	***Variance***	***Std. dev.***	***/***	***P11***	***P12***	***P13***	
P11	0-2-3	25.5	1.3389	1.0672	P11	1	0.4210	0	
P12	0-1-2-3	25	2.3889	1.5456	P12	–	1	*0.3812*	
P13	*0-1-3*	*23*	*1.3889*	*1.1785*	P13	–	–	1	
Node 4									
Path id.	***Path***	***Mean***	***Variance***	***Std. dev.***	***/***	***P1***	***P2***	***P3***	***P4***
P1	0-2-3-4	32.5	1.5833	1.2583	P1	1	**0.5377**	0.2609	0
P2	0-1-2-3-4	32	2.8333	1.6833	P2	–	1	0.4997	0.3866
P3	0-1-3-4	30	1.8333	1.3540	P3	–	–	1	*0.4806*
P4	*0-1-4*	*29.5*	*1.1389*	*1.0672*	P4	–	–	–	1

TABLE IV. SOLUTION WITH C-Z

CDF/Days	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**	**33**
F_X_4Tc	0	0	0	0.002	0.004	0.112	0.214	0.329	0.443
	34	**35**	**36**	**37**	**38**	**39**	**40**	**41**	**42**
	0.543	0.631	0.715	0.793	0.855	0.911	0.955	0.999	1
F_X_4Ti	0	0	0	0	0	0.050	0.131	0.230	0.351
	34	**35**	**36**	**37**	**38**	**39**	**40**	**41**	**42**
	0.495	0.594	0.676	0.741	0.799	0.850	0.900	0.950	1

immediately following tasks $X_{1,2}$ and $X_{1,3}$ is CDF of the respective predecessor task s $X_{0,1}$.

For node 2, start-time CDFs are discretized, and sums or convolution are computed as completion TD for the task $X_{1,2}$ is TD of path 0-1-2. CDF of maximum time in node 2 (F_X_2Tc, F_X_2Ti) and start time of immediately following task $X_{2,3}$ are computed, and required quantiles are extrapolated. Computations for nodes 3 and 4 are similar. The approximate discretized distributions of makespan are in Table IV.

C. PPERT solution

Initially, the end time of tasks $X_{0,1}$ and $X_{0,2}$ are known. The start and end time of all other tasks are set to 0, and TD in all nodes, start and end time of the rest of the tasks are assigned "unset". The steps are given in Table V. We go from the source to the sink of the network computing approximate distributions in nodes searching the distribution of maximum project time (DMPT) in each node. Node distributions depend on end time distributions of incoming tasks and determine start time distributions of outgoing tasks.

As the end TD of task $X_{0,1}$ is known. It is equal to TD of the node 1 and start time of tasks $X_{1,2}$ and $X_{1,3}$. End time of task $X_{1,2}$ is a convolution of TD in node 1 and the duration of the task. Now, knowing the end TD of tasks $X_{1,2}$ and $X_{0,2}$, we can determine the TD in node 2 where two sub-paths are met in step 1 (see Table V).

In node 2, the domination of one sub-path A measured with probabilistic comparison operation (PCO) A>B (and B>A) is less than chosen reliability coefficient $\alpha = 0.9$. Therefore, we perform MEB correction with a symmetric triangular distribution with $P_1 = h(A, B) \sim Tri(15, 18.5, 22)$ and continue with this triangular distribution as the distribution P1 of the node 2 in the next step.

In step 2 we compare distributions of two collapsing subpaths A: $Tri(15, 18.5, 22)$ and $U(6, 11)$, and B: $U(7, 12) * U(11, 16)$ in the next node 3. The former distribution is larger than the latter compared with PCO with the given α. Therefore, we continue with $Tri(15, 18.5, 22) * U(6, 11)$ under the name P2 as the distribution of the node 3 in step 3.

Step 3 is performed by analogy. P3 with support [26, 42] is the approximation of the makespan. Because there was MEB correction in node 2, the upper-bound is adjusted by (1−α)%: (42-26)*0.9 = 14.4, so, the new support is [26, 40.4]. The distribution in node 4 is symmetric triangular $Tri(26, 33.2, 40.4)$.

III. Conclusion

A summary of the analysis is given in Fig. 2. Approximate CDF of makespan depends on the method. Although the mean is almost the same with all methods, of interest is naturally the probability *above the mean.* This is a manifestation of a longer possible delay. A simulated makespan is a "perfect" time to which the methods should be close to.

As indicated previously (e.g. [2], [4]), classical PERT underestimates probabilities of longer duration. Although PNET corrects it, the improvement is small due to reliance on paths instead of analysis of nodes where MEB emerges and accumulates. On the opposite, the C-Z method overestimates the duration. Moreover, overestimation is larger if one uses by design only discretized TD and interpolation (extrapolation) to calculate distributions of successor tasks without application of sometimes available original CDF or their convolutions. It follows from Fig. 2 where lines C-Z1 and C-Z2 show the results with Tc and Ti suffixes respectively.

The approximation of PPERT is *closer* to the simulated result despite its simple type of approximating function. This simple example shows that under- and overestimation can be visible even in a small network. There is no data on the behavior of unsophisticated heuristics in larger stochastic networks. However, the error can become larger. Although comparison to simulated distributions is a usual way of performance comparison, a better way were to apply the methods to data from real projects and compare results to recorded end times of each task and makespan. The latter is our incentive for future research in this area.

Table V. Processing with PPERT, α= 0.9

Step	A	B	P(A>α B)	DMPT	MEB Corr.	Next step
1	$X_{0,2}$	$X_{0,1}*X_{1,2}$	0.43	$P_1 = h(A,B)$	Yes	2
2	$P_1*X_{2,3}$	$X_{0,1}*X_{1,3}$	0.91	$P_2 = A$	No	3
3	$P_2*X_{3,4}$	$X_{0,1}*X_{1,4}$	0.93	$P_3 = A$	No	–

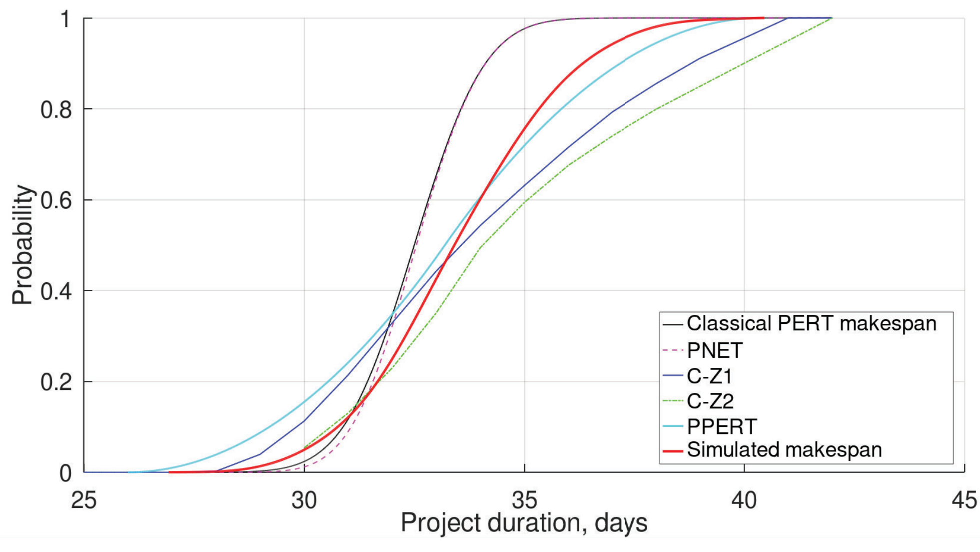


Figure 2. Approximated CDF of the project duration obtained with three methods

References

[1] P. Ballesteros-Pérez, G. D. Larsen, and M. C. González-Cruz, "Do Projects really end late? On the shortcomings of the classical scheduling techniques," J. Technol. Sci. Educ., vol. 8, no. 1, p. 17, Mar. 2018.

[2] M. Goman and J. Pecerska, "Merge Event Bias in Project Evaluation Techniques – Problems and Directions," in 2020 61st International Scientific Conference on Information Technology and Management Science of Riga Technical University (ITMS), Oct. 2020, pp. 1–6.

[3] M. Goman, "A Heuristic Technique for Project Time Analysis in Conditions with High Uncertainty," in Business Modeling and Software Design, B. Shishkov, Ed., Cham: Springer International Publishing, 2021, pp. 363–373.

[4] J. J. Moder, C. R. Phillips, and E. W. Davis, Project management with CPM, PERT, and precedence diagramming, 3rd ed. New York: Van Nostrand Reinhold, 1983.

[5] Y. Cohen and O. Zwikael, "A New Technique for Estimating the Distribution of a Stochastic Project Makespan:," International Journal of Information Technology Project Management, vol. 1, no. 3, pp. 14–27, Jul. 2010.