Visualizing Uncertain Critical Paths in Schedule Management

Robert Gove* — Booz Allen Hamilton

Brandon Herzog† — Booz Allen Hamilton

Abstract

Organizations face complex decisions about schedule management. Project managers often employ probabilistic methods to analyze inherent uncertainty in project task execution. Current tools, such as Gantt charts and sensitivity metrics, do not adequately enable project managers to conduct in-depth analysis of the many probabilistic critical paths (the most time consuming paths through an uncertain schedule) that may exist in a schedule, making it difficult to identify problematic tasks. This paper contributes two methods to visualizing uncertainty in schedules: 1) Visualization of probabilistic critical paths, which creates a critical path tree that is visualized using the LifeFlow visualization developed by Wongsuphasawat et al. [3]. 2) Milestone folding, a method that reduces the visual complexity of critical path trees without sacrificing important schedule information. Using a government project, we illustrate that the critical path tree can highlight insights about uncertain schedules that can be difficult to ascertain using existing schedule analysis tools.

1 Introduction

Organizations often face complex decisions related to schedule risk management due to the uncertain duration of tasks in their schedule. Projects are often completed with cost and/or schedule overrun [2]. This could be caused by several reasons, for example project managers may not have understood how uncertainty affected their schedule estimates or the downstream effect of tasks with unexpectedly long durations.

When analyzing project duration estimates, we found a lack of visual tools to help project managers understand the critical paths in the schedule (the *critical path* is the longest sequence of tasks in which completing any task late will cause a delay in the project finish—see Figure 1 for an example). Schedule analysts need to understand the critical paths because the tasks on the critical paths usually benefit the most from intervention such as closely managing the tasks, breaking up long tasks into parallel tasks, or assigning more resource to finish the tasks faster.

Understanding the critical paths can be difficult because of the inherent uncertainty in predicting how long tasks will take to complete. Schedule analysts often use probabilistic methods, such as Monte Carlo simulation, to perform predictive modeling on the uncertainty in a project's schedule. Schedule analysts build a statistical model of their *project schedule* (a list of tasks and their predecessor tasks) by using probability distributions to model task durations. ([1] describes how Monte Carlo can be used in project management.) Schedule analysts then need to analyze the simulation data and predict project outcomes, identify problem areas in the project plan, and provide insight into how to save time, money, and reduce uncertainty.

The critical paths can be analyzed using Gantt charts, but Gantt charts do not incorporate uncertainty and cannot effectively visualize the hundreds of different critical paths from a Monte Carlo simulation. Tasks can be scored using sensitivity metrics (such as correlation to the schedule duration, or frequency the task is on the critical path), but sensitivity metrics do not give enough context to develop a course of action. However, we discovered that the probabilistic critical paths could be transformed and then visualized using the LifeFlow [3] visualization developed by Wongsuphasawat et al., showing each critical path's relative likelihood and the average duration of each task on a critical path. We also developed *milestone folding*, a technique that simplifies critical path trees, creating a simpler, easier-to-read LifeFlow without sacrificing important schedule information.

*e-mail: govejr_robert@bah.com

†e-mail:herzog_brandon@bah.com

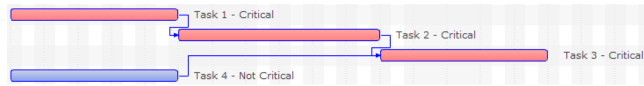

Figure 1: Task 1, Task 2, and Task 3 (colored red) must be completed sequentially with no delays between tasks, and thus they form the *critical path* because an increase in any of these tasks' duration will extend the finish date for the project.

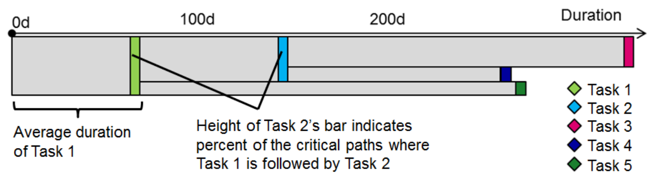

Figure 2: Example of a critical path tree visualized as a LifeFlow. The visualization is a tree showing that a task may be followed by one of several tasks. Each gray box is a task; width encodes average duration. The height of the box indicates the likelihood the preceding task was followed by the given task. For example, Task 1's average duration is 70 days, and 75% of the time it is followed by Task 2 but 25% of the time it is followed by Task 5.

2 System Design and Analysis Use Case

We expanded Polaris, our commercial Monte Carlo simulation project analysis tool, to include the new critical path tree visualization. The Monte Carlo simulation produces a thousand or more different possible durations for each task, which can result in many possible critical paths. We treat each critical path as a sequence of events, where the end of each task is an event. Because of task duration uncertainty and branches in the schedule, a task on the critical path could be followed by one task in one simulation but be followed by a different task in a different simulation. This branching effect can be seen in the critical path tree, which visualizes tasks on the critical path by transforming the critical paths into a LifeFlow [3]. Task sequences are laid out from left to right. Box width

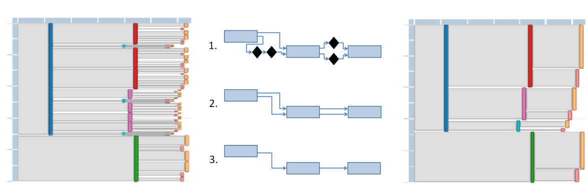

Figure 3: Without milestone folding (left) critical path trees are visually complicated and difficult to compute. Milestone folding removes milestones and collapses redundant task dependencies (center), reducing computation and simplifying visual representation (right).

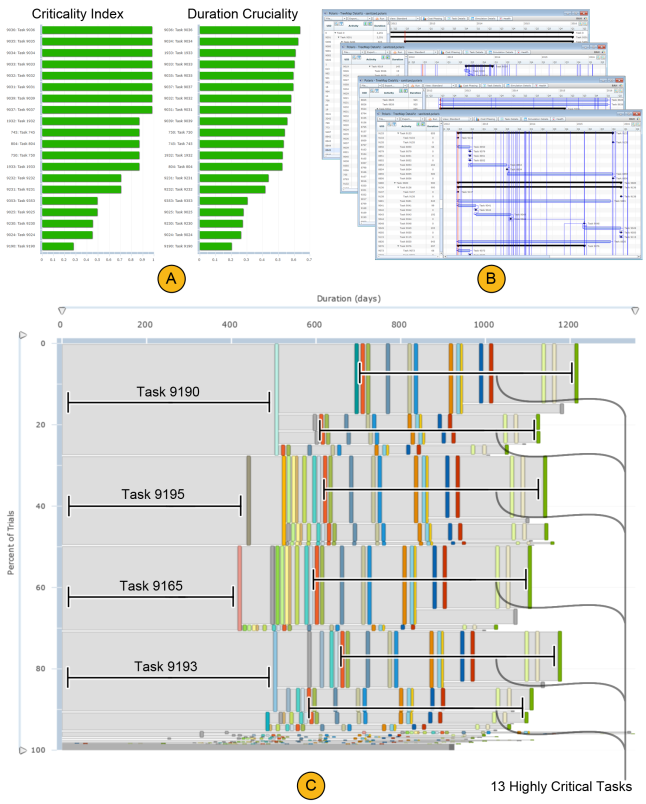

Figure 4: Views of a government agency project schedule. (A) Bar charts are commonly used to show tasks ranked by sensitivity metrics. (B) Using a Gantt chart, the schedule's 380 tasks fill 10 screens at 1920x1280 resolution. A Gantt chart showing only tasks on the critical path still fills 4 screens. (C) The same schedule's critical paths shown as a critical path tree.

encodes task duration, and box height encodes the likelihood of that task occurring in that path (see Figure 2). The box's right edge color uniquely identifies the task and may be repeated as the same task may belong to different paths. Tasks are scored by duration crucialality, a sensitivity metric combining criticality and duration, and the top 28 tasks are colored with a unique color from Polaris's default 28-color palette. All other tasks use a dark gray color.

On some schedule models, the critical path tree encountered runtime, memory, and visualization issues due to the potential for an exponential explosion in the number of critical paths present in a project schedule. We found that the issue was usually caused by parallel milestone tasks (zero duration tasks that do not represent work but instead mark events in the schedule). A critical task may be linked through two milestones to another critical task resulting in two functionally equivalent critical paths which both represent the same work completed and have equal duration (see Figure 3).

Repeating equivalent critical paths multiple times creates a complicated visualization of a simple path and implies that each equivalent instance is different, so we developed and implemented a process called *milestone folding* to remove milestone tasks from the project schedule. Milestone folding is a process where (1) the schedule is loaded, (2) all milestones are removed and their dependencies with predecessor tasks are rerouted to the successor tasks, and (3) redundant dependencies are removed. This prunes the potential number of paths through the schedule without altering the schedule network logic or critical path analysis. See Figure 3 for a step by step example of milestone folding and an example critical path tree visualization before and after applying milestone folding. Milestone folding reduced the number of critical paths in our real project schedules from millions to hundreds, allowing the critical path tree visualization to be applied where it was otherwise impossible. This created a simpler visualization that we found easier to read and interpret without losing information.

We demonstrate an example analysis on an anonymized, real schedule from a large government infrastructure project using the critical path tree. Tasks have been renamed "Task 1," "Task 2," etc. and have probability distributions defined for their durations. The schedule contained 380 tasks and previous analysis did not reveal a dominant critical path, which made it ideal for analysis with the new critical path tree visualization.

We ranked the tasks using standard sensitivity metrics: criticality, duration sensitivity, and duration cruciality (Figure 4 (A)). We identified 13 tasks that were likely to be critical, but the bar charts do not provide enough context to see when the tasks occur in the schedule, if any of these tasks are on the same critical path, or what is the expected duration of the tasks. Through lots of memorization and manually searching the 10 screens of the Gantt chart we found that these 13 tasks occur in the second half of the schedule. We failed to find any additional tasks that might benefit from management oversight or intervention, so next we filtered the Gantt chart to show only tasks that occur on the critical paths (see Figure 4 (B)). This reduced the size of the Gantt chart from 10 screens to 4, but we could not easily make sense of the many potential critical paths and the likelihood of tasks appearing on the different paths.

However, when we used the critical path tree visualization with milestone pruning we discovered insights easier and faster (see Figure 4 (C)). All critical paths are visualized simultaneously, and we easily identified the branches and likelihood of each branch. We quickly saw there are four tasks with long duration that are likely to begin the critical path (Tasks 9165, 9190, 9193, and 9195). Sensitivity metrics are supposed to automatically discover such tasks as these, but existing metrics did not discover them, and writing a customized metric for every possible problem is not as simple as using the critical path tree. We also discovered that the final 13 tasks on almost every branch of the critical path tree are the same, revealing that most critical paths converge near the end of the schedule. These are the same 13 tasks identified with the sensitivity metrics, but now we have the context to know where they occur in the schedule, why they are all very likely to be critical, and their average duration.

3 Future Work and Conclusions

We found LifeFlow visualizations of critical path trees to be more effective than existing analysis methods. The critical path tree visualization is deployed commercially in the Polaris project analysis tool, which is in use by over 30 projects across government and industry. Our collaborators (schedule analysts and project managers) enthusiastically received the new visualization, and commented that they are "not aware of any other tool that can perform this kind of analysis." Milestone folding played a key role in simplifying the schedule data and creating a simpler visual representation. Future versions of the visualization will provide options to size the width of the boxes using a different metric (e.g. duration variance).

Acknowledgements

We thank the RealTime Analytics team for the opportunity to expand Polaris, and Cody Dunne, John Alexis Guerra-Gómez, Awalin Sopan, and Krist Wongsuphasawat for discussion.

References

[1] Y. H. Kwak and L. Ingall. Exploring monte carlo simulation applications for project management. *Risk Management*, 9:44–57, 2007.

[2] K. Molokken and M. Jorgensen. A review of software surveys on software effort estimation. In *Int. Sym. on Empirical Software Engineering*, pages 223–230, 2003.

[3] K. Wongsuphasawat, J. A. Guerra Gómez, C. Plaisant, T. D. Wang, M. Taieb-Maimon, and B. Shneiderman. Lifeflow: visualizing an overview of event sequences. In *CHI*, pages 1747–1756, 2011.