PRT-218

Beyond RIFT: Improved Metrics to Manage Cost and Schedule

Presented at the
International Cost Estimating and Analysis Association (ICEAA)
Professional and Training Workshop
Portland, Oregon
6-9 June 2017

Nick DeTore
ndetore@tecolote.com

March 2017

Preface

The Risk-Informed Finish Threshold (RIFT) introduced a new concept at the 2016 ICEAA conference. RIFT is designed to help manage a project schedule with the goal of meeting a target project finish date at some probability. While we were testing the algorithm, it became apparent there may be opportunities for a broader type of analysis. We needed a new method of looking at simulation results in order to verify RIFT calculated the correct dates. Developing the RIFT validation method inspired a general exploration of how to analyze simulation data more meticulously in order to answer more precise questions.

This paper explores the answers to these questions and lays the foundation for a broader application of the RIFT algorithm.

INTRODUCTION

Since the introduction of FICSM (Fully Integrated Cost and Schedule Model) in the JA CSRUH (Joint Agency Cost and Schedule Risk and Uncertainty Handbook) released in 2014, government cost and schedule analysts have found themselves inquiring about the other's line of work. We were able to draw on experience in the oil and gas industry, construction industry and others since these professions have employed integrated cost and schedule analysis for decades.

With this merging of disciplines, an entirely new dimension of understanding the relationship between cost and schedule becomes possible. This is not without its downsides. There are aspects of schedule uncertainty models that cannot be analyzed and understood in the same manner as that of a cost model. This paper identifies these differences and provides techniques to deal with them. In addition, these techniques provide not just insight into the Schedule Risk Analysis (SRA) but can also be used to analyze the cost results of a FICSM.

This paper encourages analysts to break free from thinking we ought to analyze schedules in many of the same ways we analyze cost models. This paper introduces a general method of analyzing simulation-generated data that does not involve every iteration. We believe this new approach yields new and powerful insights into a project's potential to complete on time, on budget. This paper also introduces a new algorithm, in the appendix, for detecting clusters in data that might lead to dramatically different results to methods that do not take the clustering into account. This new toolkit provides analysts a vast new depth of understanding their models and more importantly, the projects they are analyzing.

ANALYSIS ON THE WHOLE VS THE PART

Implicit in the analysis of a cost uncertainty model is the idea of using *every* iteration generated from the simulation to calculate metrics. Each iteration result is summed throughout the model to produce results at the parent level. The more iterations, the "more accurate" the results (up until results have converged). A cost model is literally the sum of its parts; everything adds together thus every element contributes proportionally to the total.

When considering the relationship an element (part) of a cost model has to the whole, the mathematical relationships in the model cause the magnitude of cost and uncertainty to be proportional to the total (whole).

A schedule model has an essential difference that may be overlooked: a schedule is not the sum of its parts. Instead, a schedule is the result of individual task durations, but more importantly, the logic that links them together.

The relationship between an element (part) of a schedule and the total project duration (whole) is often presumed proportional to the total duration, as in a cost model. Many methods used to analyze a schedule rely on this assumed behavior. We should not take for granted that a schedule should be analyzed using the same tools and concepts used in cost models.

THE CONCEPT OF LOCAL ANALYSIS

The concept is simple: calculate a metric using only a subset of all simulation results.

In order to provide clarity in the analyses presented in this paper, we informally define two words, "cut" and "slice":

- *Cut*: a single value that separates the data into two distinct groups (Figure 1)
- *Slice*: a subset of data forming a region within the scatter; two adjacent cuts (Figure 2)

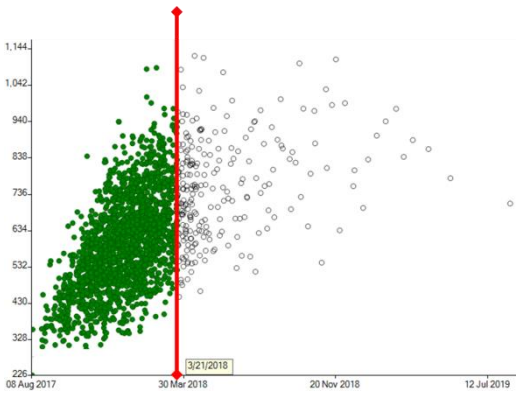

Figure 1: Cut

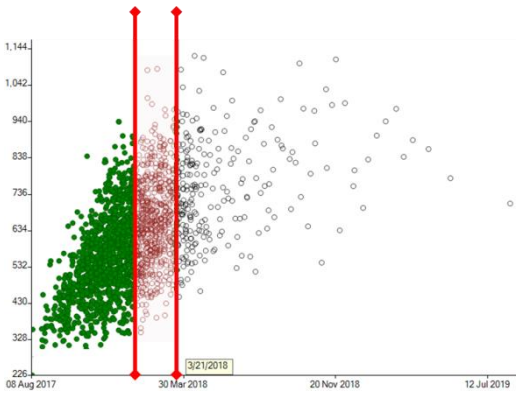

Figure 2: Slice

The most convincing way to demonstrate the need for a different way to analyze schedule results is to view a collection of typical scatter plots. These relationships are not contrived, nor are constraints used to artificially construct anomalous relationships. The scatters in Figure 3 are actual simulation results from a single, simple FICSM model constructed for this paper. These scatter plots illustrate each iteration result for two selected elements from the model; these elements are various combinations of a duration, finish date, or cost in order to exhibit examples of the complex relationships that manifest in an SRA and especially in a FICSM.

Note: each axis of each chart in Figure 3 is either a cost, duration or finish date but since these examples are for demonstrative purposes, specific labeling has been left out.

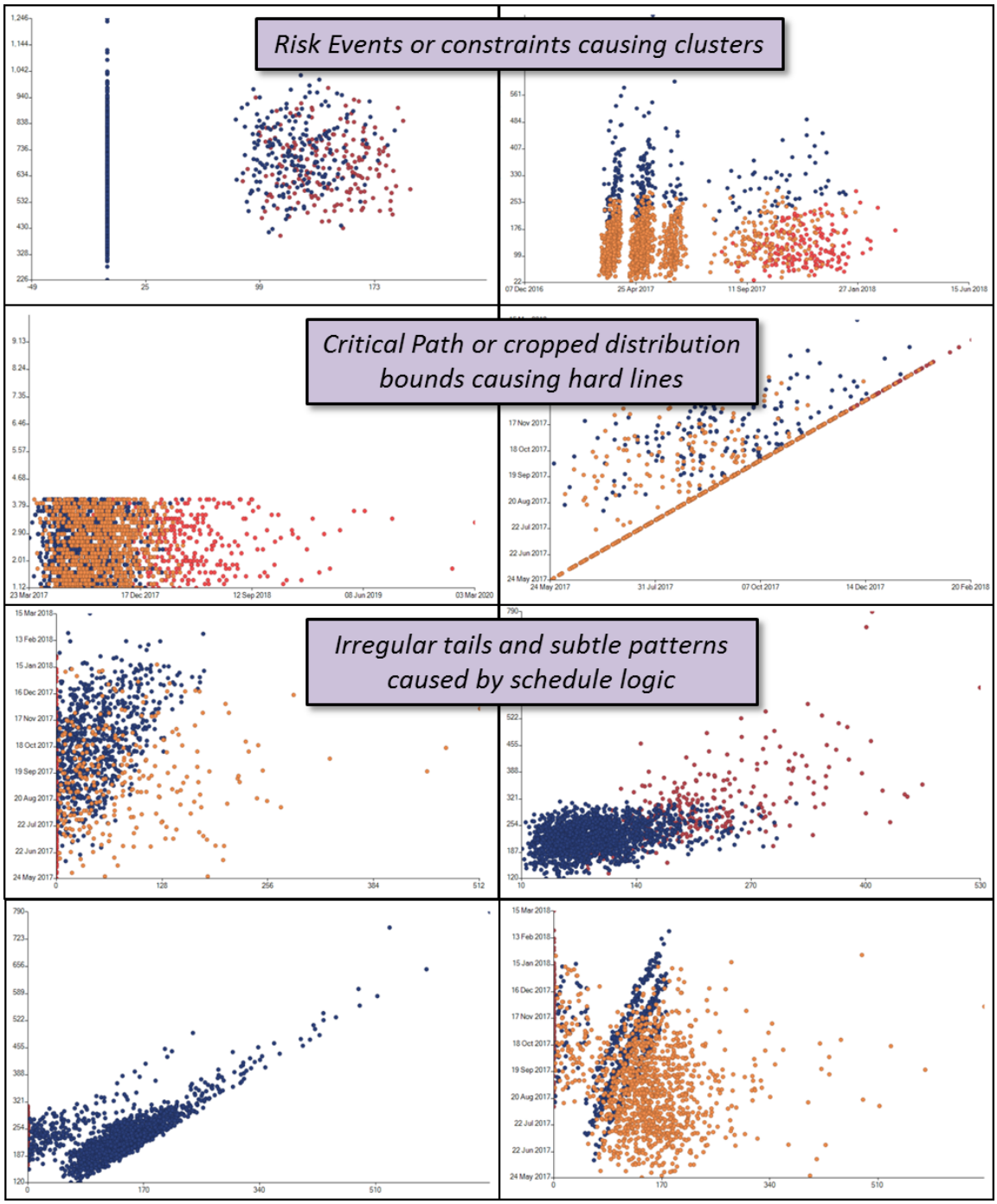

Figure 3: Relationships within a Schedule

The colors in the scatters in Figure 3 depict each of the two element's critical path status on that iteration, with four combinations possible: both on (red), one on and one off (dark orange), one off and one on (dark blue), both off (orange). The color is used to underscore another dimension of relationships that this paper helps analyze.

The averages (and other descriptive measures) within a *cost model* are justifiably accepted statistical results of all possible outcomes. The mean, along with and the standard deviation, accurately captures the general resulting behavior of the element's relationship to the total cost after a simulation.

The averages (and other descriptive measures) within a *schedule model* may be misleading, as hinted at in Figure 3. Additional metrics may be necessary to fully describe the element's relationship to the total project after a simulation.

Schedule models also require a comprehensive understanding of the critical path (the shortest path through the schedule). Additional metrics such as criticality are available to describe how often/likely a task lands on the critical path, which directly affects the project finish date.

All the atypical relationships shown in Figure 3 (and many more) can be understood by the metrics discussed in this paper.

LOCAL ANALYSIS DEFINED

Local analysis creates metrics based upon a condition. The concept of conditional metrics is a well-defined statistical definition, but the term "local analysis" in this context is not seen in the common toolkit of SRA or FICSM analysis; this paper devised "local analysis" for purposes of this discussion. To measure a subset of related data often provides more insightful, precise results than that same measurement based upon the entire dataset.

DEMONSTRATE A NEED FOR LOCAL ANALYSIS

A common metric utilized for schedule uncertainty analysis is the criticality index. It answers the question - "What is the probability this task is on the critical path?"

The answer to that question is technically an average; on each iteration of the simulation the task is either on or off the critical path, 1 or 0. The criticality index is an average value: the sum of the number of iterations the task lands on the critical path divided by the total number of iterations.

To demonstrate a need for greater context regarding the criticality index, consider the unassuming, uncertain schedule in Figure 4.

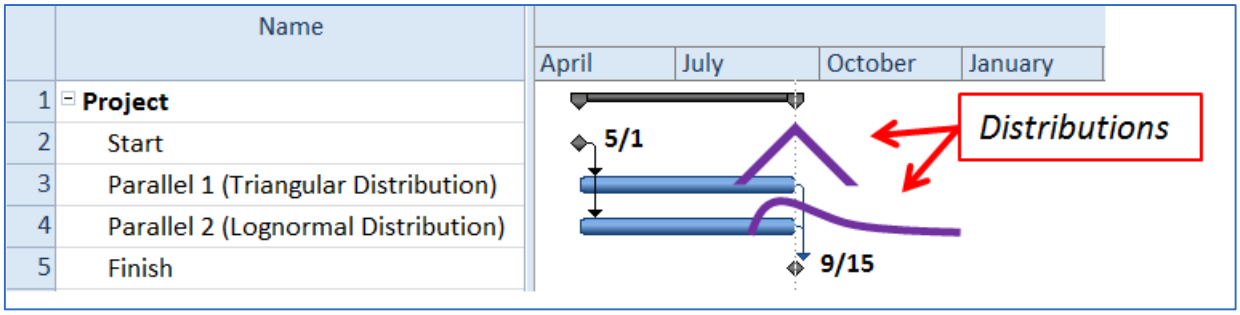

Figure 4: Simple Schedule Example

This schedule contains just two parallel tasks, same duration, with two different distributions (triangular and lognormal) with moderate duration correlation. The goal is to finish the project by 10/1/2017 (which is about the 70% project finish date result). Table 1 contains results from a simulation that reveals both tasks (seemingly) equally impact the project.

Task	Mean Duration	CV	Correlation to Project Duration	Criticality Index
Parallel 1 (Triangular)	100	0.45	0.77	53%
Parallel 2 (Lognormal)	100	0.50	0.86	49%

Table 1: Simple Schedule Uncertainty Results

The metrics in Table 1 are calculated based upon all possible outcomes. General, broad calculations like these potentially overlook vital behavior within the schedule. Taking a closer look at the relationship between each task and the project reveals something remarkable. Figure 5 contains a scatter plot of the iteration results of each task finish date versus the project finish date.

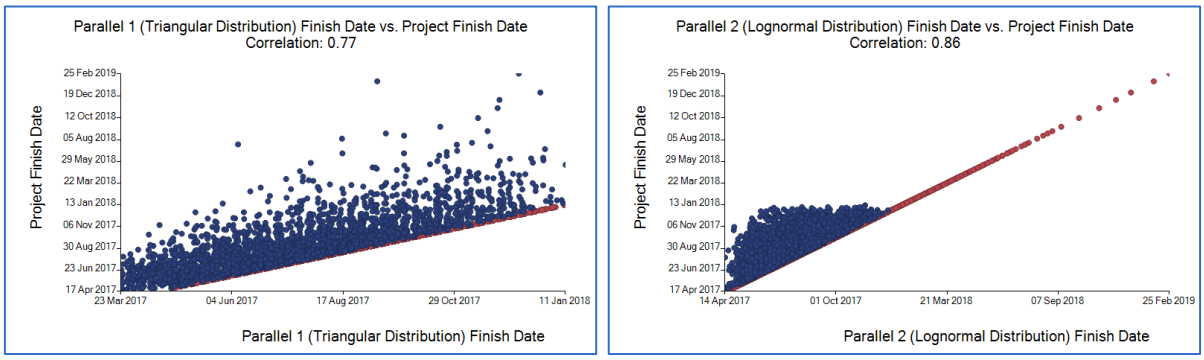

Figure 5: Comparing Parallel 1 and 2 Scatters

"Parallel 1" task and Project clearly have a different relationship than "Parallel 2" task and Project that is not captured using the common metrics seen in Table 1. The red dots in each scatter represent the iterations in which that task lands on the critical path; the blue does represent when that task did not fall on the critical path (i.e. the other task did on that iteration).

Figure 5 reveals that the *longer tail* of "Parallel 2" task causes the task to eventually becoming critical every time it extends beyond a specific duration. This is seen in the straight line of iteration results that reveal a one-to-one relationship between "Parallel 2" task and the Project finish date once "Parallel 2" task extends far enough out. Metrics relating to a task, as well as the impact it has on the project, need to account for this behavior.

Upon further review of the schedule, Figure 6 shows how the behavior of "Parallel 2" task differs drastically based upon where it finishes.

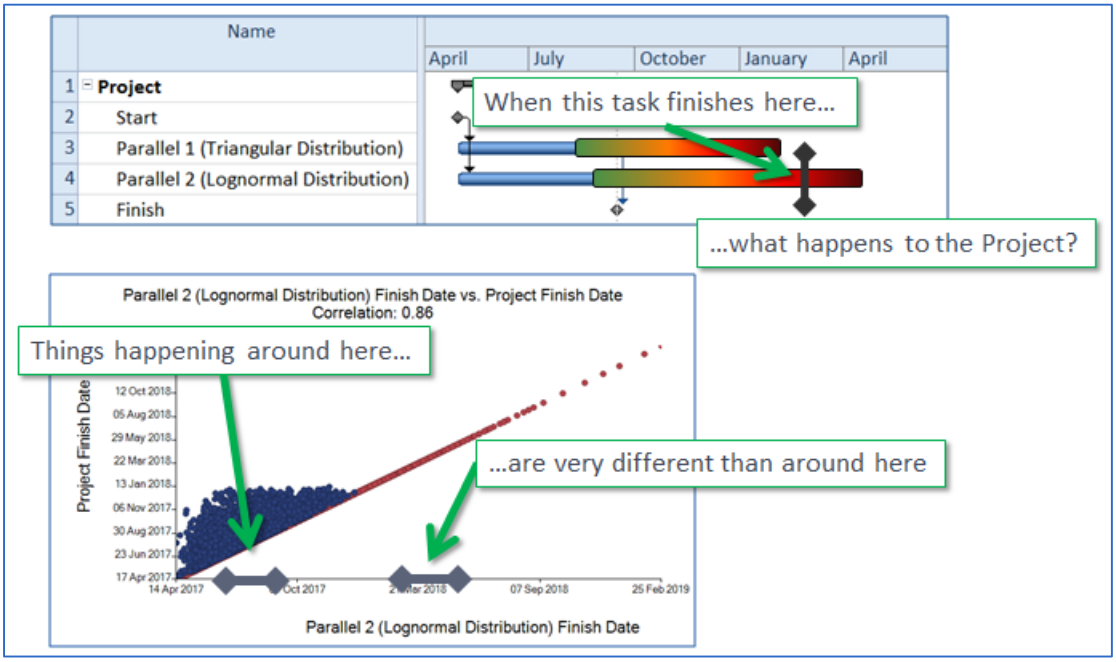

Figure 6: A Need to Ask the Right Questions

This leads to the conclusion that calculating metrics based upon all possible outcomes does not provide enough information.

CRITICALITY SINGLE-CUT DEMONSTRATION

With the help of local analysis, a straightforward solution exists to the question:

"What is the probability this task lands on the critical path *given it extends past a given date?"*

To calculate, we *cut* the data at a selected date and then use only those iteration results beyond that date to calculate criticality. Figure 7 is a visual representation of this idea. Figure 7 shows the different criticality results that are calculated when the data from the sample schedule is cut at the task planned finish date.

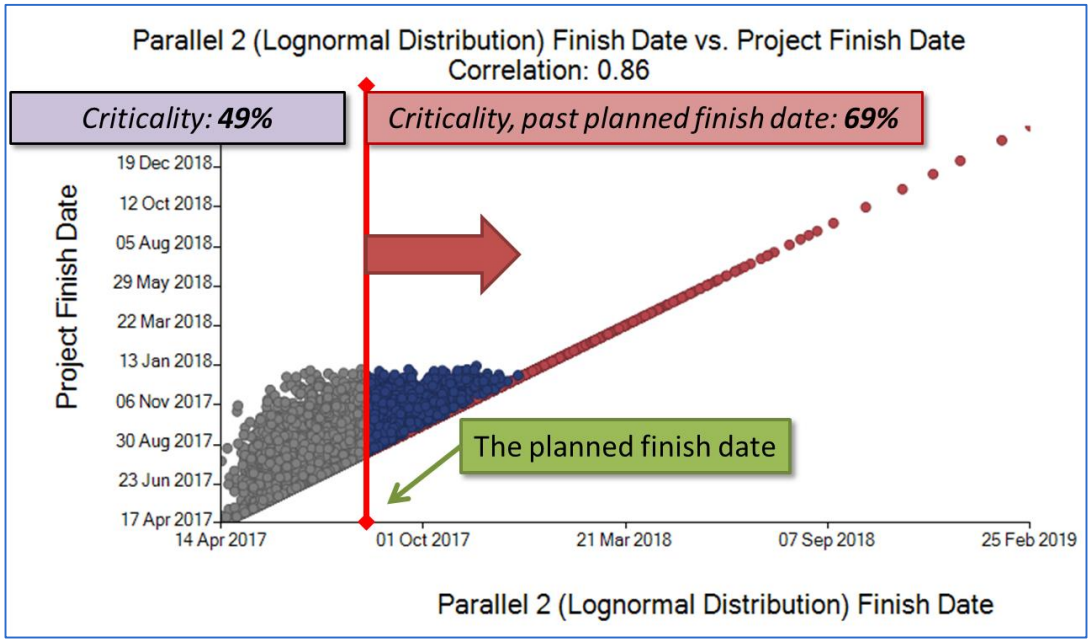

Figure 7: Criticality Example with Cut at Plan

To capture how the distribution tail affects criticality, the cut could also be placed at the 85% finish result of "Parallel 2" task. If a manager wanted to know "how bad can it get?" each task should be cut at a chosen probability level and criticality values compared. Adding this conditional criticality the metrics in Table 1 provide a far more robust understanding of the project, demonstrated in Table 2.

If "Parallel 2" task was cut at the 85% result, the criticality jumps up to 87% which is actually *higher* than the criticality of "Parallel 1" task cut at the same place.

Task	Criticality Index	Criticality past Plan Finish	Criticality past 85% Finish
Parallel 1 (Triangular)	53%	71%	71%
Parallel 2 (Lognormal)	49%	69%	87%

Table 2: Conditional Criticality Report

CRITICALITY SINGLE-CUT DRIVER REPORT

A quick way to incorporate this concept into current driver reports is to select where the cut should be defined, e.g. the planned finish date or the mean finish date, and perform that cut on all tasks. A comparison of the top list of the original (unconditional) criticality and the conditional criticality provides uniquely vital insight into tasks worth paying attention to.

A simple schedule, seen in Figure 8 was created from scratch for this paper in order to demonstrate concepts. The schedule contains only triangular and lognormal distributions, three risk events, and only "finish-to-start" relationships.

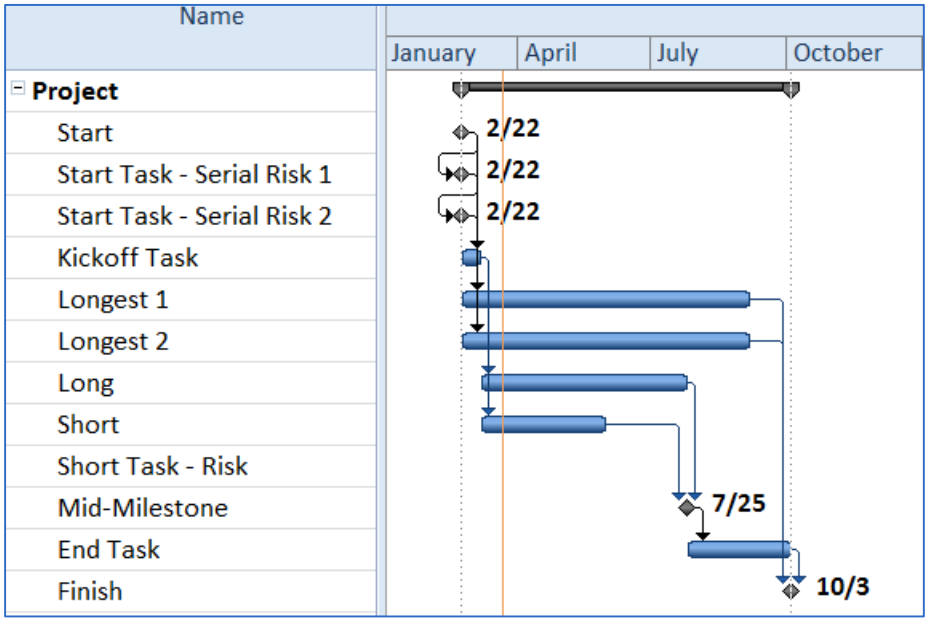

Figure 8: Simple Example Schedule

Figure 9 reveals a graphical approach comparing top drivers switching places when considering a scenario of local analysis.

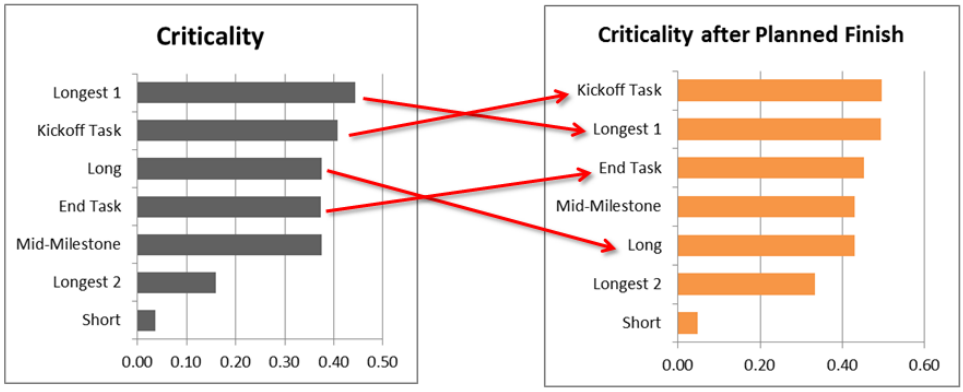

Figure 9: Comparing Criticality

With this added restriction, the answer has an added piece of information. The answer states what happens during a specific condition (e.g. period of time), rather than what might happen during all possible outcomes; more specifically, local analysis allows the program manager to understand which tasks should be focused on to ensure they do not overrun planned finish dates (or any chosen date result).

CRITICALITY MULTI-CUT DEMONSTRATION

Using a single cut provides insight into behavior given a single condition has occurred. The next logical step in this process is to understand the behavior of an element as it continually progresses. Graphically viewing the criticality of an element as it gets pushed further out provides vastly more information than a single, average value.

With the groundwork laid, the analysis continues simply by adding cuts throughout the simulation data. Figure 10 shows what continuing the cuts from Figure 7 looks like.

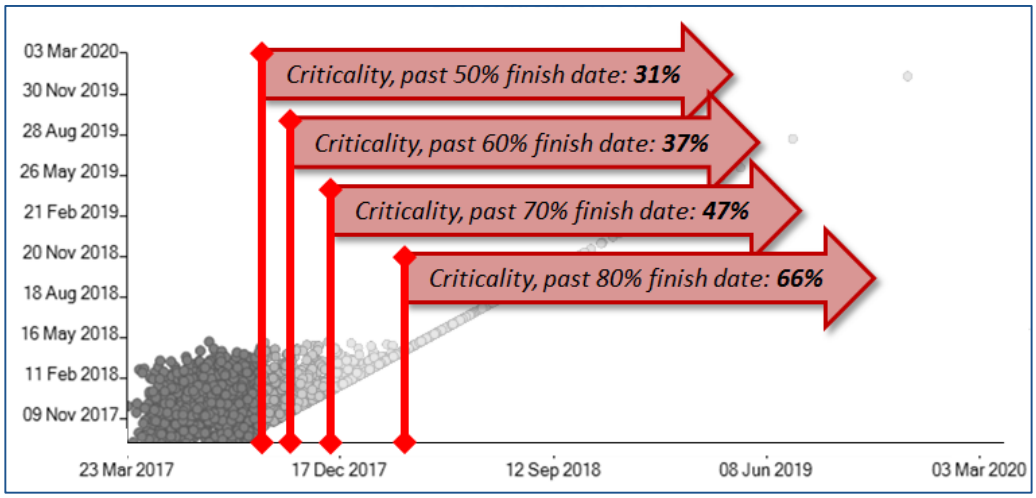

Figure 10: Criticality Example with Multiple Cuts

The information presented in Figure 10 can be succinctly represented in a basic line chart seen in Figure 11; this chart does not have a standard name nor is it prevalent in any common analysis tools, thus it is quite possibly a brand new presentation of data. The x-axis dates are all possible task finish dates as generated by the simulation. The y-axis is the likelihood the task lands on the critical path *given the task extends past the finish date.* The first point on this chart therefore represents the original (unconditional) criticality index calculated based upon all possible finish dates.

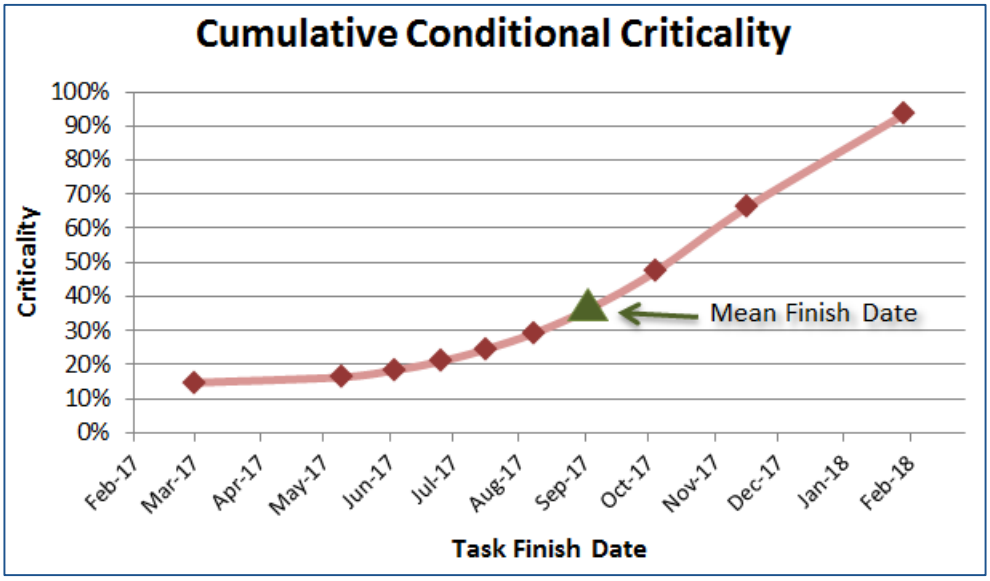

Figure 11: Cumulative Conditional Criticality

A useful variation of the data in Figure 11 changes the x-axis into the probability level results instead of the actual finish dates to provide a standard scale (Figure 12). This allows more than one task to be plotted so that the change in criticality over time can be compared between two or more tasks, with the caveat of course being the loss of relative duration between each task.

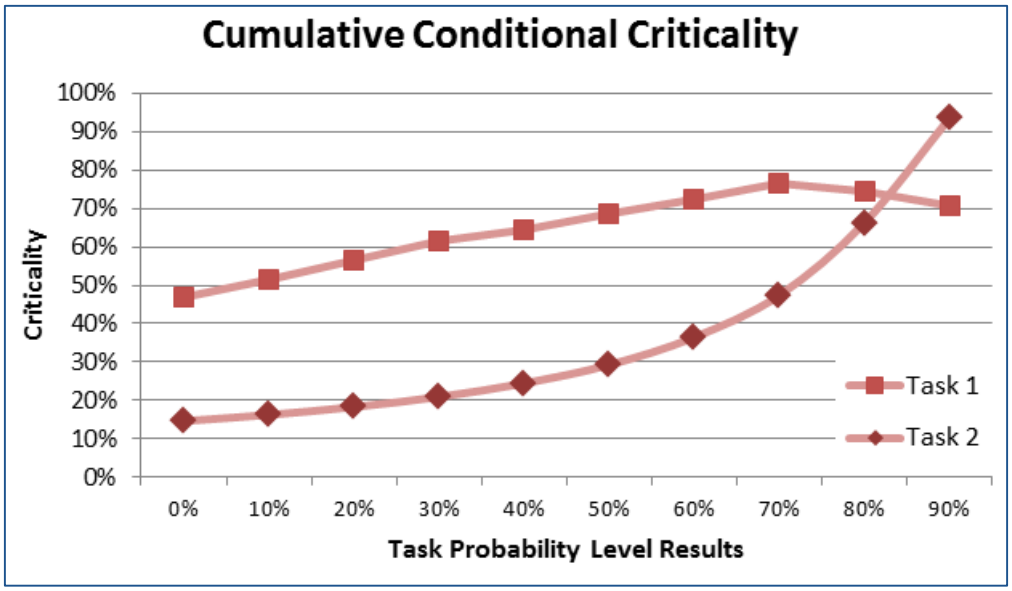

Figure 12: Comparing Cumulative Conditional Criticality

There is yes another variation to consider. This is a simple modification to the x-axis, seen in Figure 13 to show days beyond the planned finish date, rather than a date.

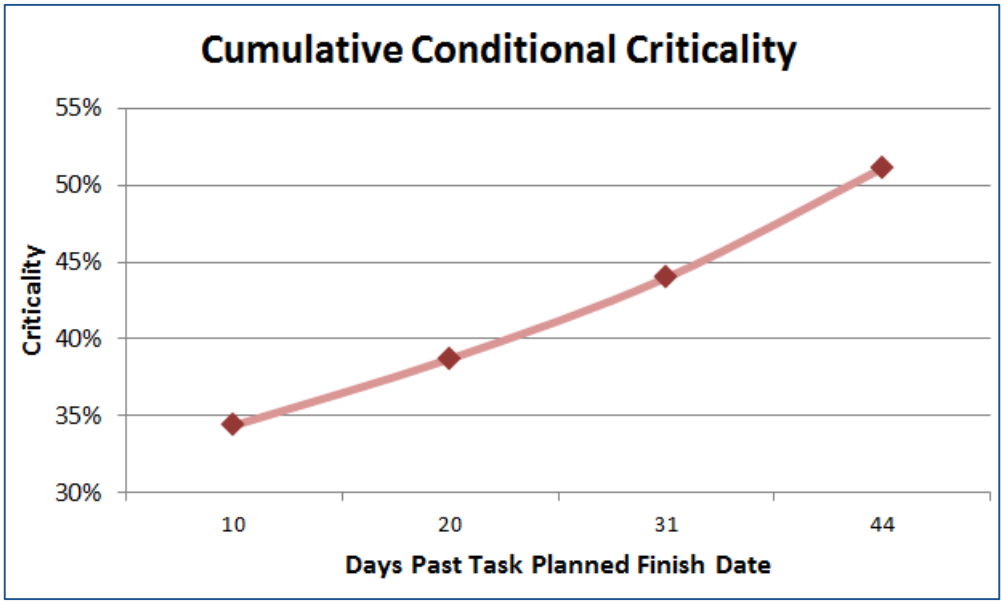

Figure 13: Planned Finish Date Delta - Cumulative Conditional Criticality

SLICE-DEMONSTRATION OF LOCAL ANALYSIS

Cumulative views of data are a fundamental aspect of what gets reported after any uncertainty analysis. The probability of a duration being "at most", or of the budget "not exceeding", are examples of cumulative statistics. When a cut is placed in data, the resulting metric is cumulative.

Local analysis does not need to be based upon a single cut, it can also be based upon a slice of data.

For example, imagine there is a Program Event (or Milestone or Gateway) in the schedule that marks the finish of a major phase in the project that has a highly restrictive budget or contractual restriction. Standard analysis provides a cumulative set of probable costs: "there is a x-percent chance it will cost y-amount or less." Local analysis provides a broad new set of results. For example, "If I finish *on* this date, what is the projected average cost?" Figure 14 demonstrates how a slice of data provides context to the other standard metrics.

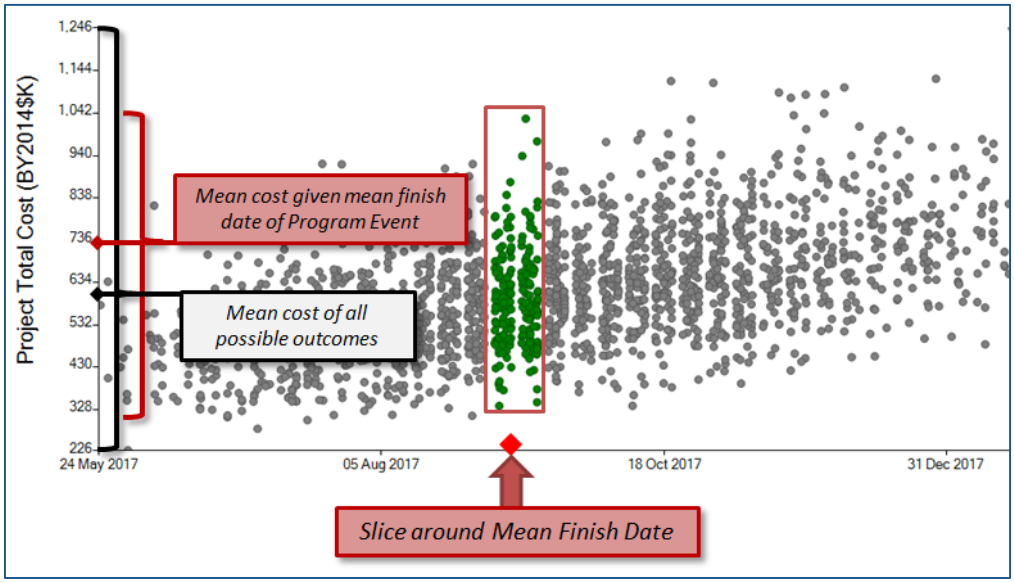

Figure 14: Slice Metric

A slice compared to a cut is analogous to a histogram compared to an s-curve. A slice provides information given a very specific set of criteria has been met. The issue with this approach, as with histograms, is the size of the slice, i.e. the bin width. There are a variety of equations that provide a value for the width, but there is no well-defined "right" answer. Appendix A of the "Joint Agency Cost Schedule Risk and Uncertainty Handbook" (see CSRUH, 2014) contains a concise table with formulas for choosing bin counts.

NEW SETS OF QUESTIONS THAT CAN BE ANSWERED

A goal of a FICSM is to gain an understanding of the relationship between cost and schedule. The JA CSRUH discusses how to model uncertain time-independent (TI) and time-dependent (TD) costs to a task. This is the vital step in linking the duration of a task to the total cost of the project. Once this is achieved, there is an entirely new dimension of data available. The analysis concludes with answers such as "what is the probability I finish at or before this date *and* at or below this cost?" Local analysis can provide much more insight.

Local Question: "What is average total project cost *given my project extends beyond the target finish date*?"

Local Question: "What is the average cost of this task (or summary) *given it finishes as planned*? *Given it extends to the 70% finish date*?"

Local Question: "Which tasks have the most dramatic rate of cost increase most when the task gets pushed out further?" (Calculate a collection of slices and the tasks with the largest cost rate increase should be flagged for review)

CONDITION ON THE PROJECT INSTEAD OF THE TASK

That same conditioning on a task discussed thus far in this paper can also be placed on the project results. This measures how tasks directly impact the project under certain conditions.

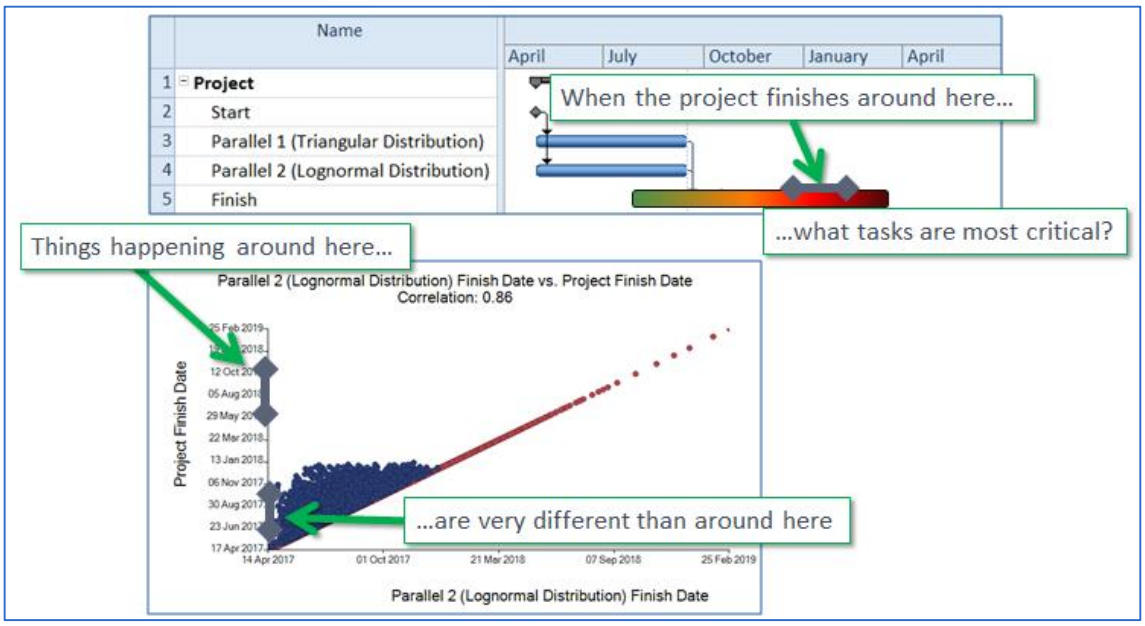

Figure 15: Conditioning Upon the Project

Figure 15 is analogous to Figure 6 except the question is asked about the Project rather than the task. Given a condition about the Project finish dates, task metrics are then calculated, compared and ranked. For example, listing each task's criticality given a specific Project finish date allows managers to understand what tasks are most influential in the environment they are managing towards.

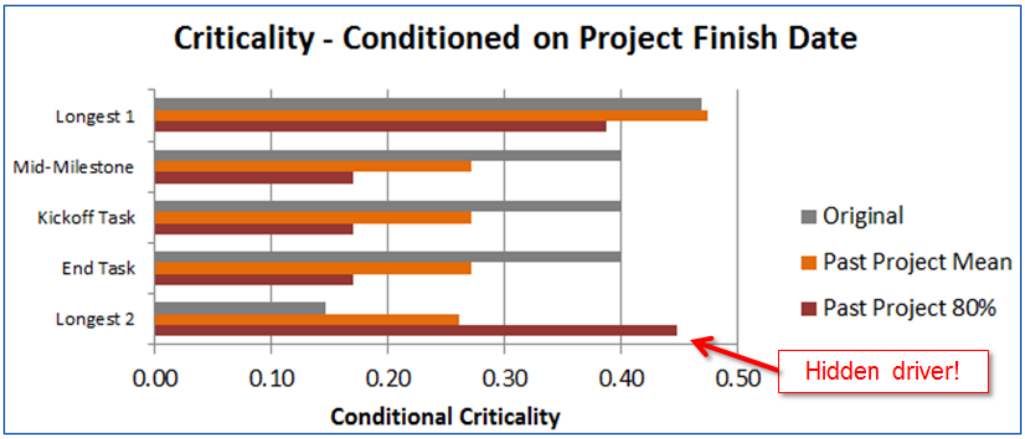

Figure 16: Conditioning Upon the Project – Criticality

Figure 16 reveals that the original ranking of most critical tasks changes in this example schedule if the Project finishes past the 80% result.

TARGETED LOCAL ANALYSIS

Local analysis discussed thus far relates to a measurement between two sets of data. While the concept is about adding a condition before the measurement takes place, there is a method for an even more specific form of analysis.

This paper defines "targeted local analysis", or simply "target analysis", as a measurement relating a *collection of data* to a *single, fixed data point* (often a Project finish date target or a total budget). This provides unique measurements of how elements of the schedule impact a target/goal for the project.

Using concepts from local analysis, additional questions can be answered using target analysis.

Question: "If my task extends beyond the plan, what is the probability of achieving the *Project Target Finish Date*? If my task extends beyond the plan what is the probability of staying under *budget*?

Answer: Using only the iteration results greater than the task plan (or any given date), calculate the probability of the Project finish date target within that corresponding subset, i.e. take the Project finish date iterations that correspond to the task finish dates greater than the task plan and find where the target date lies within that Project finish date subset.

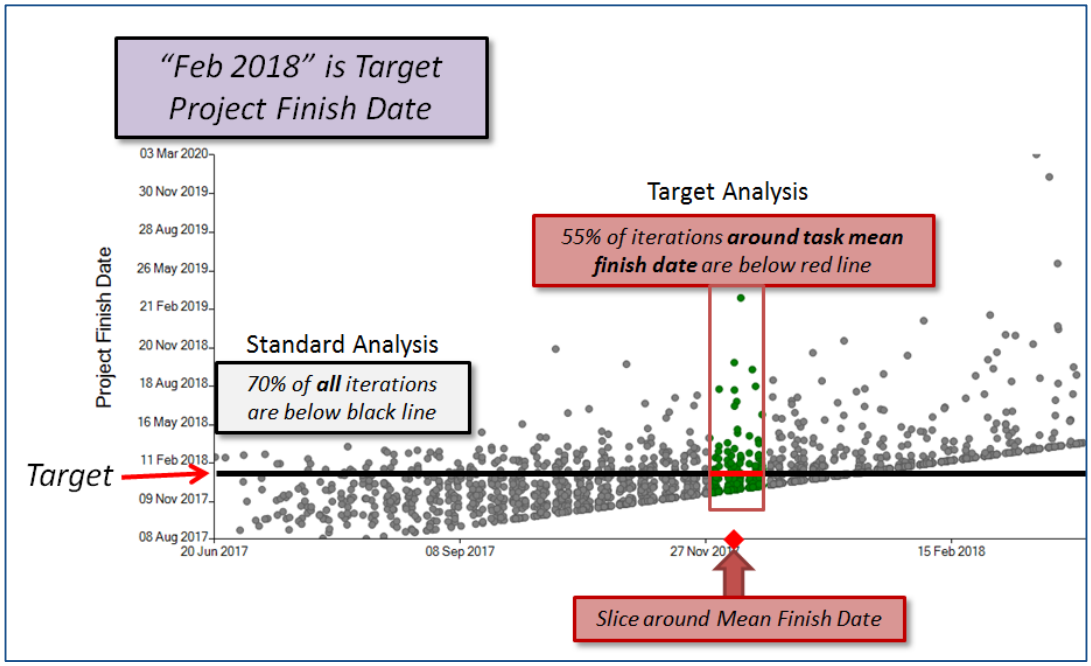

Figure 17: Single-Slice Target Analysis

SLICE IT UP!

Target analysis is not limited to using a single cut in the data. Slices of data reveal the behavior of a task over the entire span of all possible outcomes, providing enticing new insight.

Question: "*As my task continues to slip,* how does this impact the probability of achieving my Project finish date target?"

Answer: Repeat the calculations listed in the previous answer but instead of the criteria "iteration results greater than the task plan", take multiple slices of iteration results and recalculate the probability of the target date within each new resulting subset (slice) of Project finish date results.

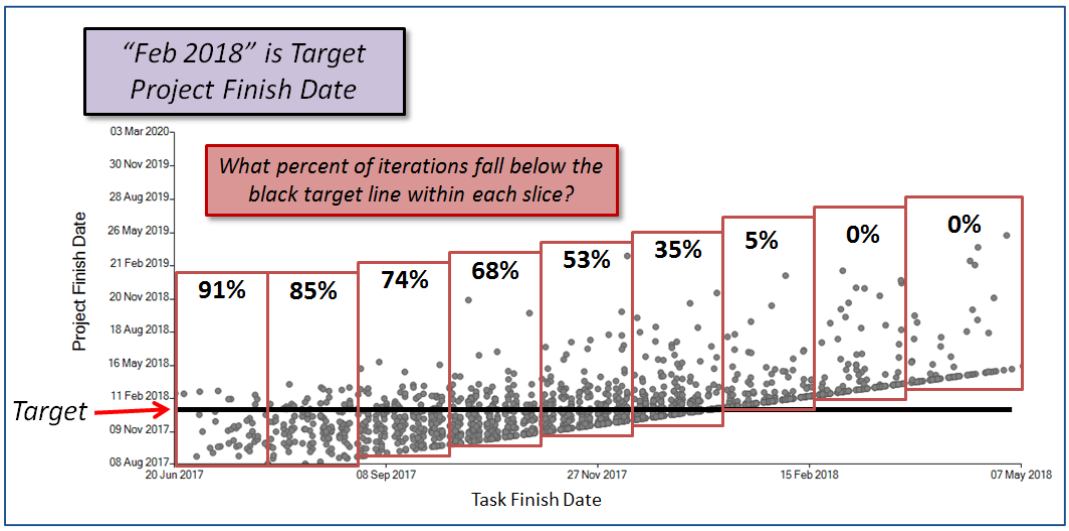

Figure 18: Multiple-Slice Target Analysis

Figure 18 shows a task finish date result being sliced and the probability of achieving the target calculated in each bin; as the task extends further and further, the probability of achieving the target decreases.

The rate at which the probability of achieving the target decreases can be used as a single value to rank tasks that are directly affecting the target. This rate is related to correlation (between a task and the Project) but it is a more tangible measurement that also forecasts the direct impact of task delays.

A useful visualization of the data calculated in Figure 18 is analogous to the line charts of Figure 11, Figure 12, and Figure 13. We create a separate chart for the target analysis results. Only one example will be shown, but the same variations for ways to plot the x-axis exists for these target analysis results as exists for the line charts (dates, probabilities, deltas).

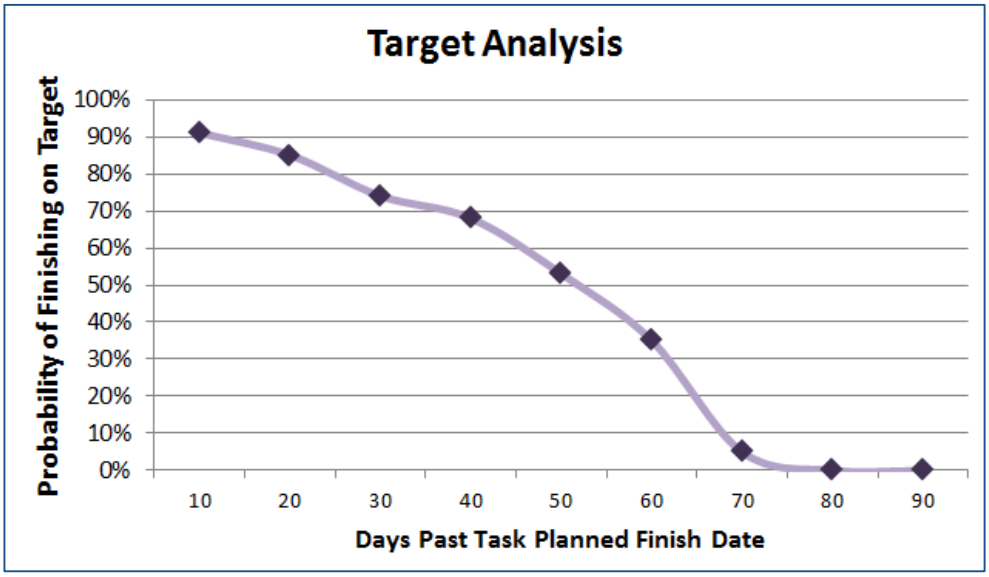

Figure 19: Plotting Target Analysis

Figure 19 plots the values from Figure 18 on a separate line chart. This analysis reveals something *no other standard metric can*: once this task extends past about 70 days beyond the planned finish date, there is virtually no chance this sample project's target finish date can be achieved.

ACTIVE TARGET ANALYSIS

This paper defines "active target analysis" as using local analysis on a specific target to *find* the condition needed to achieve the goal. Rather than measuring a dataset relative to a single value, a single value can be solved for based upon the dataset. This allows an analyst to *solve* for a desired target.

Question: "*What date* does my task need to finish on in order to maintain the probability of my Project finish date target?"

Answer: After dividing up the entire set of results into slices, the target's probability can be measured within each slice (seen in Figure 19). Within (at least) one of these slices, the probability of the target is the same as the original probability (i.e. the target relative to the entire set of results). As long as the task finishes within this range of finish dates, the target probability is maintained. Additionally, as long as the task finishes *before* this, the probability of the target is improved. There is actually an analytic formula to calculate this date: The Risk-Informed Finish Threshold (RIFT). See the paper presented at ICEAA 2016 for full details on the RIFT date.

Question: "*Which tasks* are most likely to impact my Project Target Finish Date the most?"

This question sounds like classic driver analysis will provide the solution but the question specifically relates to a single target/date. Unfortunately, classic driver charts measure against *all possible* finish dates or costs.

Long Answer: This is not quite as simple as previous answers since there are nuances that need to be taken into account. Tasks whose RIFT date is closest to the planned finish date of the task are those tasks that have very little "reserve" or "buffer". Additionally, the uncertainty of the task must be taken into account; simply because a task finish date may lie near its RIFT date does not mean it is likely to slip past it unless the task is uncertain.

Short Answer: Tasks with the smallest gap between planned finish date and their RIFT date that also have high CV, some criticality, and at least moderate correlation are most likely to impact the Project Target Finish Date.

Consider Figure 19 once more. This line chart shows the effect a task has on the probability of achieving the target project finish date. If the manager needs to ensure the project does not slip past the 70% project finish result, this chart reveals how the task must not extend 40 days beyond the planned finish date. This chart can be used to *find those dates* for any selected elements that need to be finished on or before in order to keep the project on target.

The RIFT date is an analytic solution to this concept. Local Analysis can be used to not only search for the RIFT date but explore what happens if a task passes this date. The slope of the line connecting the first point nearest the target probability to the last point on the chart. This line is shown in Figure 20.

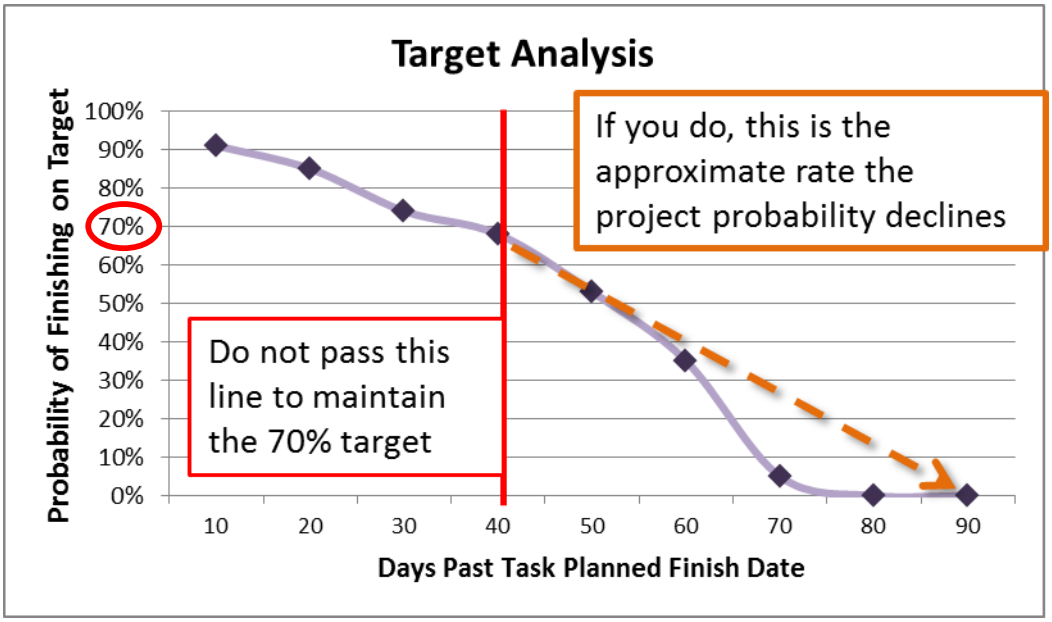

Figure 20: Advanced Target Analysis

This information can form an entirely new driver report: a ranking of every task's rate of decline *past the RIFT date* finds those task that directly impact that singular target date the most.

SUMMARY AND CONCLUSION

LOCAL ANALYSIS REVIEW:

- **Take a single cut** (Figure 21) – Analyze data corresponding to all values greater/less than that cut
  - Example: given task finishes past date X, what is average finish of Project?
  - Example: given task finishes before date X, what is that task's average cost

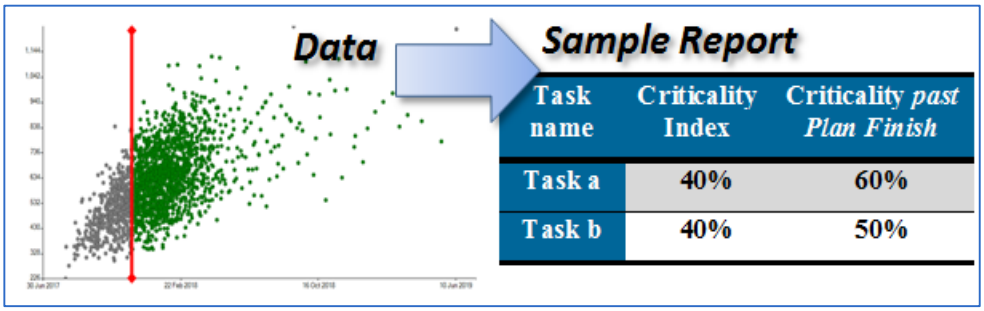

Figure 21: Single Cut with sample Report

- **Take multiple cuts** (Figure 22) – Analyze data corresponding to all values greater/less than each cut, cumulative measurements
  - Example: How does criticality change as my task extends? Which tasks criticality increase the most?
  - Example: What tasks cost increase the most dramatically as each extends?

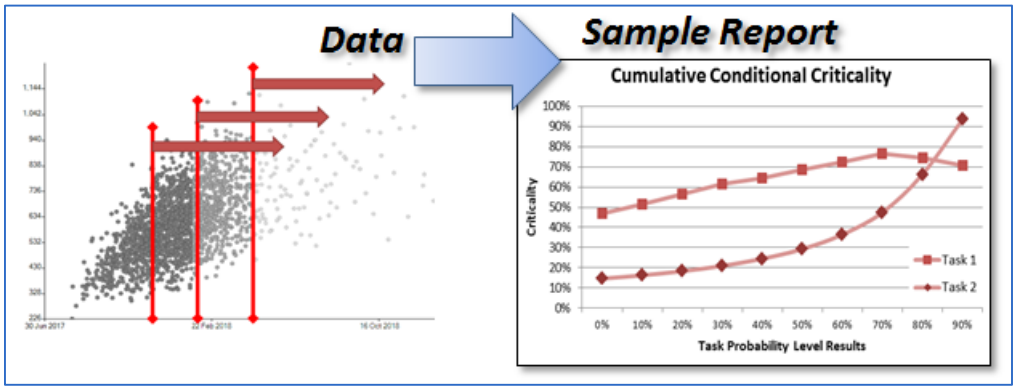

Figure 22: Multiple Cuts with Sample Report

- **Take a single slice** (Figure 23) – Analyze only the data corresponding to only those values within the slice
    - Example: What is the probability of landing on the critical path if my task finishes at the mean finish date?
    - Example: What is the probability of achieving my target Project finish date if my task finishes at the mean finish date?
    - Example: What is my total Project cost if my milestone finishes as planned?

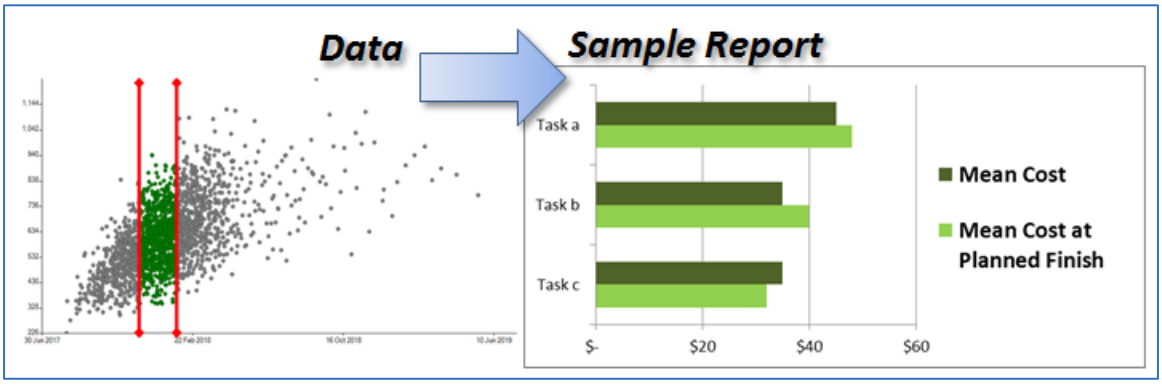

Figure 23: Single Slice with Sample Report

- **Take multiple slices** (Figure 24) – Analyze data in a collection of bins
    - Example: At what milestone finish date does my target Project finish date drop below the desired probability? What is the rate at which this probability decreases if the milestone is delayed further?
    - Example: What tasks increase their probability of landing on the critical path the most as they each extend?

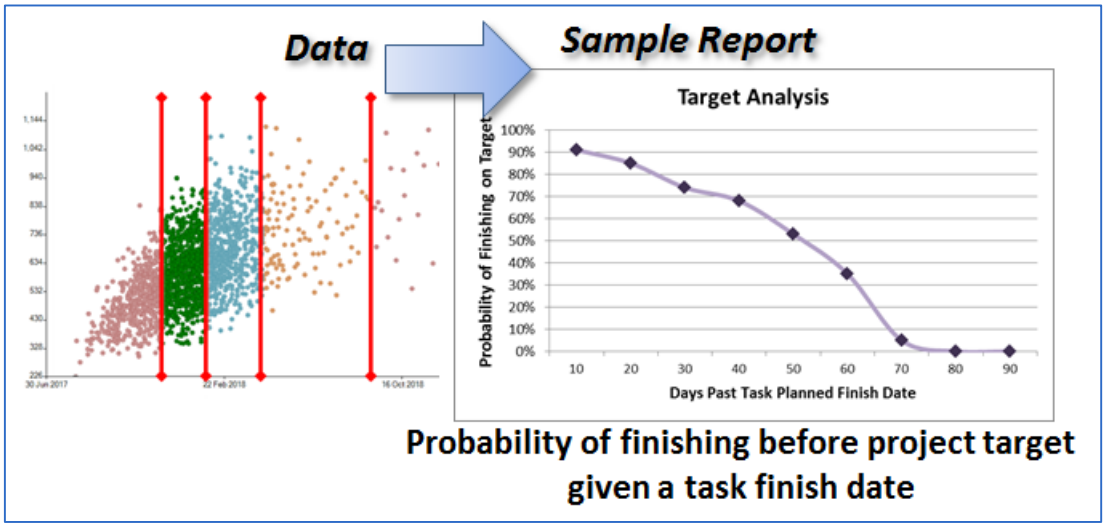

Figure 24: Multiple Slices with Sample Report

CLUSTER ANALYSIS OVERVIEW

Before any analysis (local or otherwise) is performed, the cluster algorithm presented in the appendix is a quick and efficient calculation that can be done on all rows in any model to flag a "yes" or a "no". If a task is flagged "yes", care needs to be taken when reviewing metrics for the task, otherwise the metrics will likely be nonsense or misleading. If a cluster goes undetected, Figure 25 shows how a task might have a mean finish date that is not even a possible finish date. Finding the cluster and locally analyzing the data provides vastly more robust and actionable results.

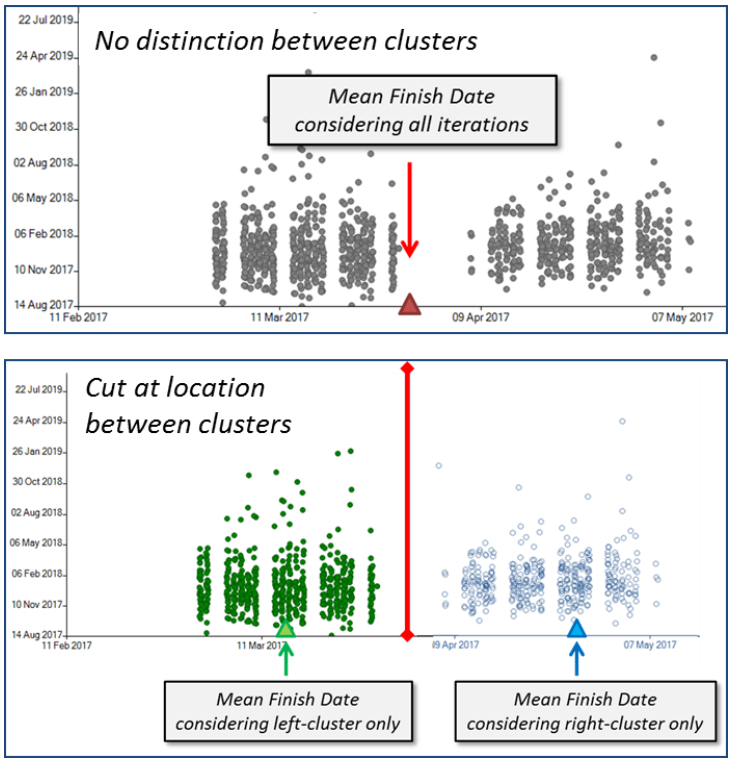

Figure 25: Danger of an Undetected Cluster

FINAL THOUGHTS

This paper is intended to improve the analysis of a SRA or FISCM model by providing new, innovative tools to analyze simulation data. Managers are inundated with new ways to assess project cost and schedule models. The methods introduced in this paper deliver a relevant result using language that helps managers manage. The barriers between cost analysts and schedule analysts are beginning to fall away. The methods and metrics used to analyze simulation data between the two disciplines should not always the same. This paper points out where there are vital fundamental differences that need to be addressed to generate useful, accurate information.

RIFT analysis required an ability to analyze a subset of iteration results (i.e. local analysis) in a controlled manner to verify the algorithm. This requirement led to the concept of locally analyzing data. It became a purpose independent of a RIFT date, opening up an entirely new set of metrics needed to fully understand the complexities of uncertain integrated cost and schedule models.

Understanding if data is clustered (discussed in the appendix) was another vital piece of information RIFT analysis revealed was a necessary part of a robust analysis. Performing the standard metrics on an unknown clustered dataset could lead to misleading conclusions and actions. Knowing a cluster exists is a good first step. Knowing *where* the split occurs is even more powerful when the analysis understands how to utilize local analysis.

The charts and reports presented in this paper are examples of the fruit of this labor but the higher aspirations here are to open the doors to a new generation of analytic tools that answer questions that traditional analysis simply cannot.

APPENDIX A – CLUSTER ALGORITHM

CLUSTERS

We have seen a collection of schedule simulation results that require careful local analysis to fully understand the behavior of the model. There is a type of result that has not yet been discussed: clustered data. Once again due to the structure of schedule models, in ways cost models inherently do not behave, it is possible for results to be clustered. To be fair to cost models, introducing risk events can create discrete jumps in costs (i.e. clusters), but often it is recommended to not model the risk within the same context of the simulation results and instead provide that analysis separately. Modeling risk events into a FICSM or SRA is an essential step in building the model. Even without discrete risk, a schedule may contain clusters if constraints, such as "finish no later than", have been placed upon tasks (although constraints are not recommended by most uncertainty analysis scheduling best-practice guidance).

This section discusses the definition (or lack thereof) of a cluster, the importance of understanding if and where the schedule contains clustered results, how to detect them, and how to use local analysis to deal with them.

This paper presents a straightforward solution, a *new algorithm*, to add into the arena of cluster analysis.

The application of this algorithm is narrow relative to the most common cluster algorithms. A fundamental drawback to the widely used methods is that they are generalized in what they are able to analyze. With the flexibility of most methods comes ambiguity and subjectivity, i.e. they lose the ability to *definitively* answer the questions they seek to resolve.

The algorithm presented in this paper is designed to answer two specific questions

1. Is my data clustered?
2. If so, where is the gap in the data?

A modification will be discussed briefly that provides the ability to detect more than one split in the data, although in schedule models, the likelihood of multiple partitions in the data decreases dramatically after the first.

WHAT IS A CLUSTER?

Cluster analysis is a tremendously *vast* and *vague* topic. Cluster analysis is *vast* in the sense that any field involving data mining or statistical analysis needs to be aware of the shape of the data and relationships within the data; it is *vague* in the sense that the term "cluster" actually has no precise definition. Intuitively a cluster is a group of related data within a larger dataset; the problem is that a "group" and how the data is "related" are not strict, statistical notions.

HOW ARE CLUSTERS DETECTED?

A cluster algorithm is any calculation which attempts to find groupings within a dataset. There are different categories of algorithms but each of them must define what constitutes a cluster in the context of that algorithm. This field is overflowing with subjectivity. There is no objectively "best" algorithm since there is no universal definition of a cluster.

Whenever an algorithm claims to detect clusters, ground rules and assumptions must be formulated carefully in order to judge the validity of the results.

COMMON DRAWBACKS TO CLUSTER ALGORITHMS:

- *Subjective input*: The user must specify the number of clusters before running the algorithm
    - "Centroid-based" algorithms, such as the common "k-means" technique, group the data into the closest *k* cluster centers (the value *k* is user-specified)
    - "Density-Based" algorithms, such as DBSCAN, require two user-specified inputs that define the cluster size and shape.
- *Additional interpretation*: The output of the algorithm is a graph showing many possible partitions of the data without a single objective result
    - "Hierarchical" algorithms often produce a dendrogram which graphs an extensive hierarchy of many possible clusters
- *Baseless assumptions*: The algorithm itself is testing against an assumption of the distribution of the clusters
    - "Distribution-based" assumes each cluster belongs to (or produced by) an explicit distribution. Fitting real data to distributions can be problematic.
- *Time-consuming*: Almost all algorithms require non-trivial computation. They are recursive computations or approximations to a system of equations with no closed-form solution.

The cluster algorithm introduced in this paper addresses each of these issues:

- *Subjective input*: No input needed before calculation. There is a single value that can be modified for a more advanced analysis.
- *Additional interpretation*: The output is simply "yes" or "no".
- *Baseless assumptions*: No assumptions regarding the underlying data
- *Time-consuming*: A distinguishing aspect of this new algorithm is that it does not involve solving any equations, not a single iteration nor convergence is necessary, making this a near-trivial calculation.

NEW CLUSTER DETECTION ALGORITHM

Ground rules and assumptions:

- *Univariate analysis*: analyzing a single array of data
    - When analyzing schedule uncertainty results, nearly all clusters seen in a scatter plot can be detected by analyzing a single array of results; a cluster in the context of our analysis can almost always be found by looking at a single elements iterations results.
- *Gap*: A difference between two points large enough to be considered "space between two clusters," defined as being at least 2.5 times greater than the standard deviation of the next largest space between two points.
    - The value of 2.5 is based upon a collection of simulation results: Continuous distribution iteration results have naturally occurring spaces between data points (blue dots in Figure 26); i.e. "natural gaps" due to the nature of the random draws. 2.5 times the standard deviation of the "natural gap" values was found to be a threshold that accounted for virtually all these values derived from continuous distributions.
    - Figure 26 shows what randomly distributed data looks like compared to data with an "unnatural" gap (e.g. a risk event occurred)
- *Cluster*: when a gap exists between an ordered set of data that cannot be accounted for as caused by an outlier, the data on either side of this gap form a cluster

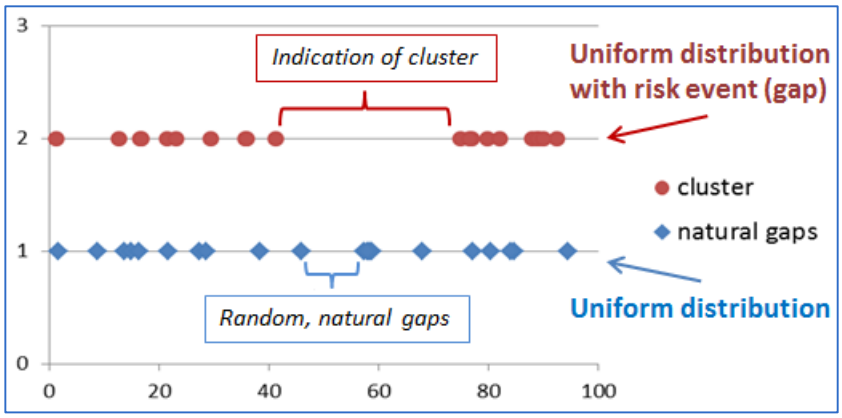

Figure 26: Natural Gaps vs Clusters

General Idea:

Simply stated, this algorithm looks for a gap in univariate data that is larger than what would occur naturally from a simulation.

There is a simple modification that allows the algorithm to test for two (or any number of) gaps in case the data is split into more than two distinct clusters, but this is a rare phenomenon in the context of schedule uncertainty analysis thus it will not be discussed at length.

In this section the steps are first outlined, then explained in detail, and then visualized.

Steps Outlined:

Given a univariate dataset, Y:

1. Order the data from smallest to largest to create Y1
2. Remove the top 1% and bottom 1% points from the dataset to create Y2
3. Calculate the difference between each successive data point to create Y3
4. Remove the single largest value from Y3 to create Y4
5. Calculate the number of standard deviations from the mean (of Y4) for each data point in Y3 to create Y5
6. Divide the single largest value in Y5 by the second single largest value in Y5. If this value is greater than 2.5, there is a cluster and the algorithm returns a "yes".

While the algorithm itself is multiple steps, there is nothing to solve, they are deterministic, i.e. non-iterative.

Steps Explained:

Given a univariate dataset, Y:

Step 1: Simplifies Step 2 (if using Excel)

Step 2: Removes data points that might just be a product of extreme tail values that could cause a false gap not caused by a cluster. If no cluster exists then removing these values does not affect the results.

Step 3: By analyzing the distance between each successive point, this transforms the problem into a type of outlier analysis. The most thorough analyst is recommended to use a calculation of skewness on the original data Y in order to ensure the outlier analysis doesn't have the added difficulty of differentiating between extremely skewed/sparse data and a single outlier.

Step 4: If there is a gap, then the largest value in Y3 is the point representing that gap. The mean and standard deviation of Y3 would be unfairly inflated if this largest value in Y3 is actually an outlier so to prevent an inflated mean and skewed measurements, it is removed before the outlier measurements in Step 5. If the largest value is not an outlier, this single removal does not significantly affect results.

Step 5: An outlier test.

Step 6: To ensure the potential gap is not a product of simulation results, even if it is a large distance from the mean, it needs to be unique. If the largest distance in Y5 is at least 2.5 the next largest distance in Y5 (i.e. the "gap ratio"), the data has a single large distance between two points, thus indicating a true gap that forms a clustered dataset.

If the data was not clustered, Step 6 would produce a value close to 1 because the gaps would be values of relatively similar magnitude. It would not matter if the largest potential gap was far

from the mean if the next largest gap is also far from the mean. That situation simply implies the data has a large tail, not a gap.

Steps Visualized:

The following set of images depicts each step involved in the cluster algorithm. Note that the x-axis label for some figures change.

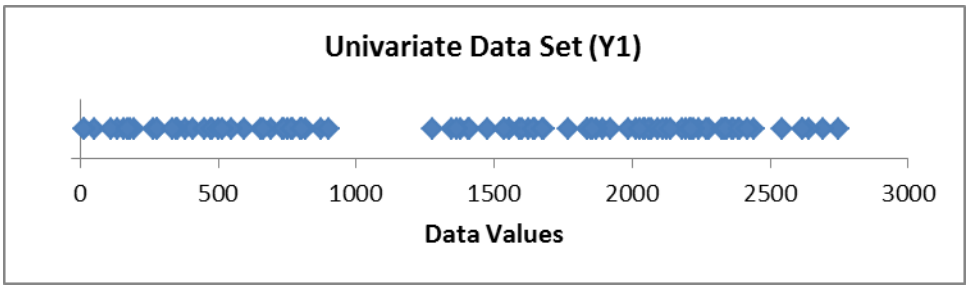

Figure 27: Step 1 - Arrange a single variable smallest to largest

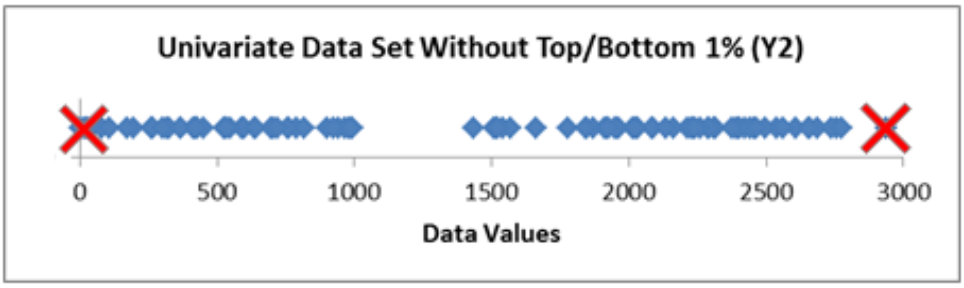

Figure 28: Step 2 – Remove top and bottom 1%

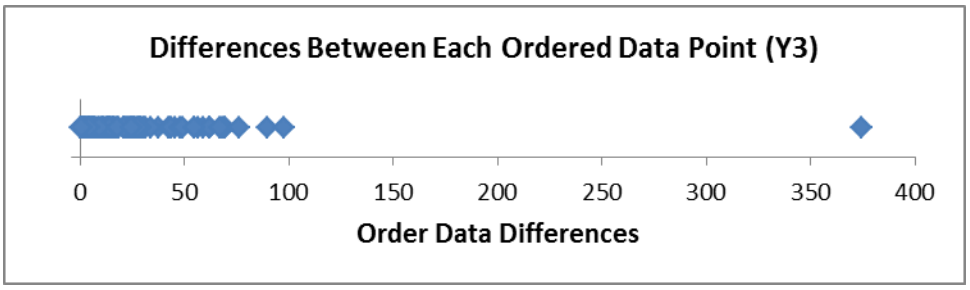

Figure 29: Step 3 – Calculate the difference between each point

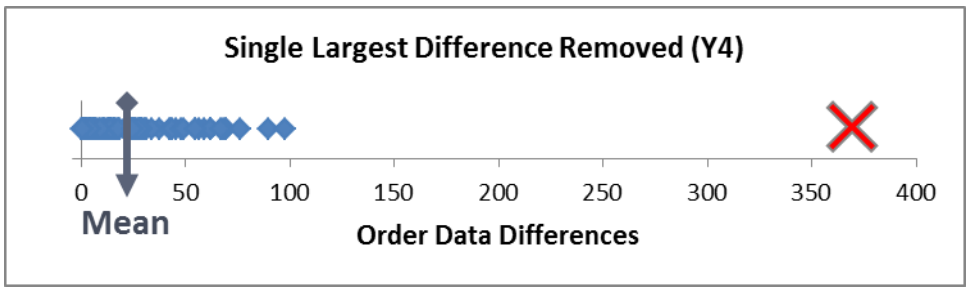

Figure 30: Step 4 – Remove largest value from Y3 then calculate mean

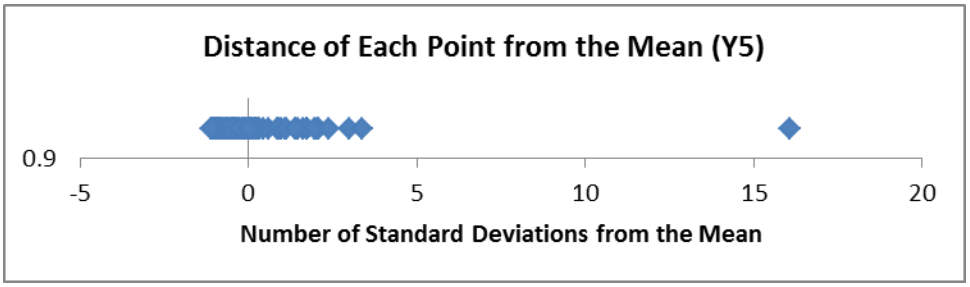

Figure 31: Step 5 – Calculate # St. Devs. from Y4 mean for each point in Y3

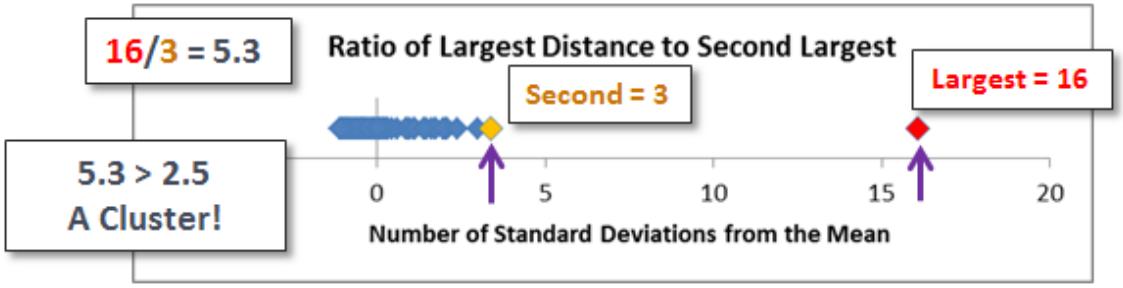

Figure 32: Step 6 – Determine if potential gap unique

Notice that if the data had no sufficiently large gap, Figure 31 and Figure 32 would be a collection of points around 0. The ratio of the largest to the second largest would be close to 1.

IMAGES OF CLUSTERS AS DEFINED BY THIS ALGORITHM

As previously mentioned, a cluster is not strictly defined. This section demonstrates the visual representation of a cluster as defined by this new algorithm by showing a sequence of figures. The algorithm is based upon the gap between ordered points, thus the exact distribution and location of the data is not vital to the results. The depictions below are not meant to be a complete representation of what this algorithm can detect, but simply a visual cue in which to get the reader familiar with what a "gap" is and is not. Recall a "gap ratio" of 2.5 or greater is considered indication of clustered data.

As can be seen in Figure 33 the exact distinction between a large space and a cluster will always boil down to some unavoidable amount of subjectivity and knowledge of the underlying data. Fortunately this algorithm contains enough information in the calculation that the algorithm's steps can be used to find out more about the data even if not flagged strictly as a cluster, such as how wide and where the largest gap appears.

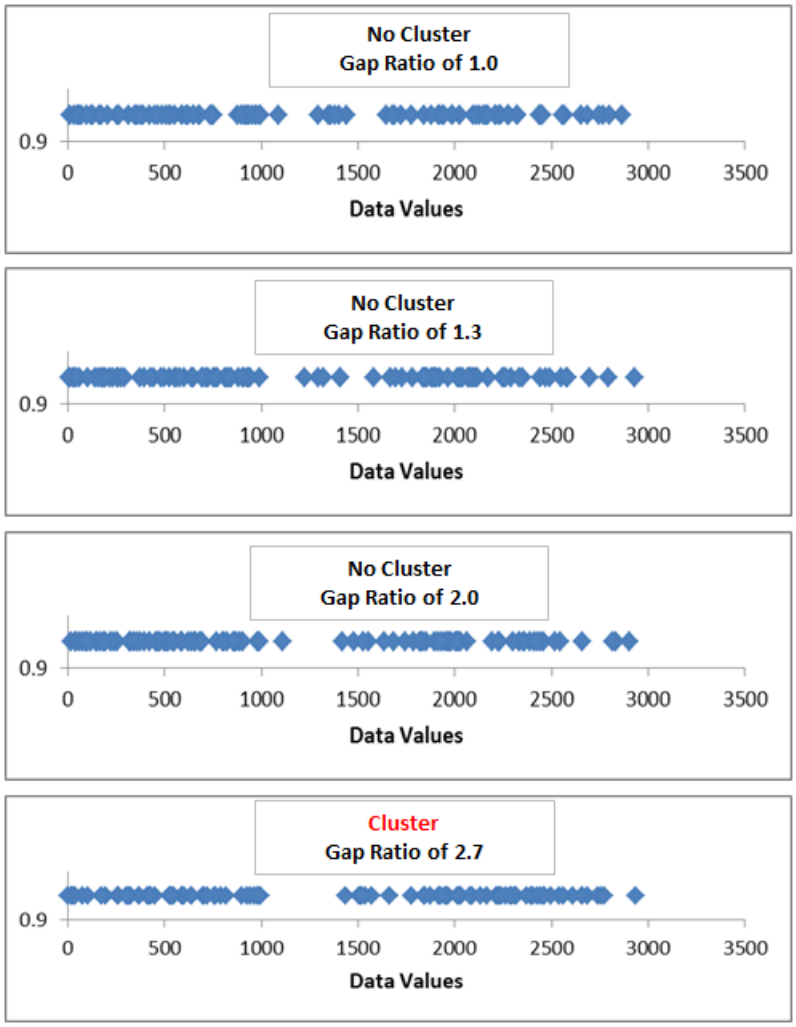

Figure 33: Spectrum of Non-Cluster to Cluster

False Gap:

Given the nebulous nature of simulated data, it is possible a gap detected but actually does not exist when the data contains a very large tail. In this situation, there is a collection of points spread out at very large distances, all far from the mean and all far from each other. Even after removing the top 1% in Figure 34, there still exists large gaps relative to the rest of the data.

Figure 34: Extremely Skewed Data

False gap results are remedied by calculating the skewness of the dataset and the data flagged if the skewness metric resulted in an extreme value, as determined by the skewness measurement.

Multiple Gaps:

On rare occasions, two distinct gaps have been seen in a schedule uncertainty scatter plot (although it was not plotting a task against the Project Finish Date but rather two tasks within the schedule). In this case, the algorithm would not find a gap since the ratio of largest to second largest gap would likely not be large enough (both values from Y5 would be nearly equally large thus a small ratio); imagine Figure 32 with two points distinctly separate from the rest. To test for two distinct gaps, take the ratio of the second gap and third largest gap.

Final comments:

Let it be clearly stated that cluster analysis is an entire field of ongoing research. This algorithm is a modest attempt at a straightforward solution within the context of SRAs and FICSMs. The ability to quickly declare that a cluster exists is immensely beneficial. While there will always be a grey area in which no objective calculation can inform the analyst with utmost certainty, the algorithm presented in this paper nonetheless aims to take great strides at equipping the diligent analyst with powerfully insightful tools in the quest for a deeper understanding of data.

RELATING CLUSTERS BACK TO LOCAL ANALYSIS

Knowing if the data is clustered provides context, especially when searching for top drivers, that without, would be an incomplete understanding of the schedule model. If a task is clustered then the pitfalls of the standard (non-local) metrics used to understand its behavior are exaggerated. The mean finish date, the criticality (which is technically simply another “average” metric), even measures of dispersion like the standard deviation are all subject to fatal flaws. For example, it would not be hard to imagine a mean finish date calculated in the middle of the gap between two clusters, especially if the clusters are made up of about equal number of iterations (Figure 25 and Figure 35). This is not only useless it is nonsensical. We should prevent that kind of a result where we can.

Another benefit of the new cluster algorithm is the ability to *locate* the gap. The center of a gap in the simulation data is a perfect place to *cut* the data to perform local analysis on the two separate clusters. This provides a powerful result of being able to say “there is a gap in the results of this task between date X and Y, if the task finishes *before* date X, this is the average time it spends on the critical path and this is its impact to the target; if the date finishes *after* date Y, the time spent on the critical path and impact to the Project target finish date drastically increases"

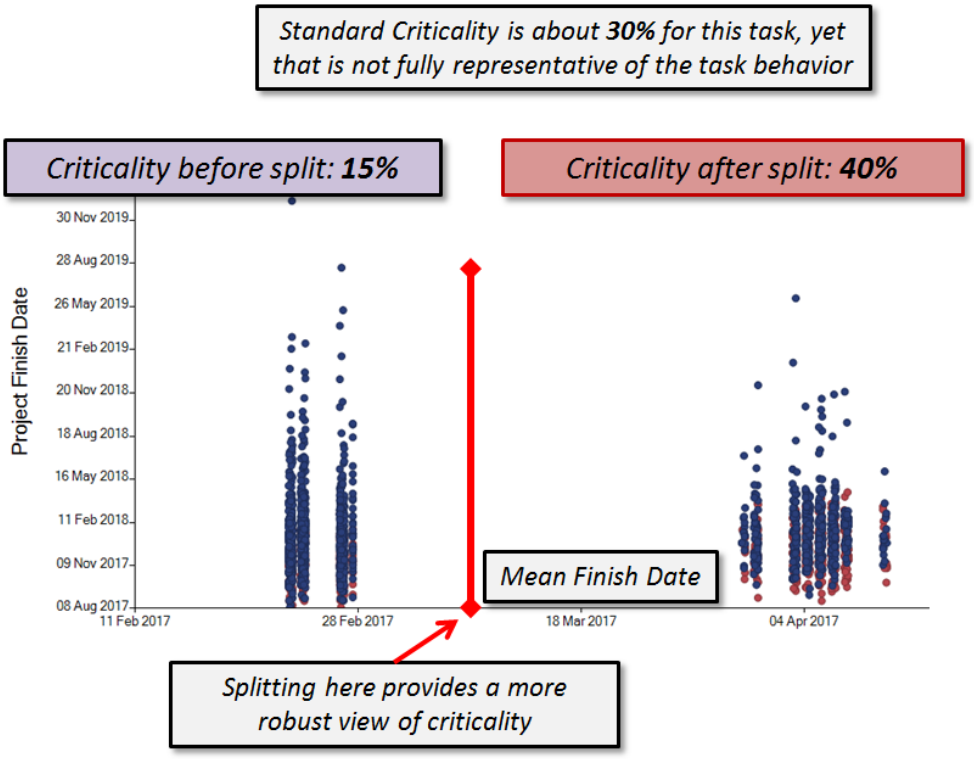

Figure 35: Benefits of Locating Cluster

Figure 35 shows a task with two distinct clusters in the x-axis. The task is never critical before the split (in the first cluster) but dramatically increases the chances of landing on the critical path when the task duration pushes out into the second cluster. This behavior cannot be captured by the standard metrics.

At first glance Figure 35 seems to have up to *five* clusters, but the definition of a cluster as presented in this paper considers the smaller gaps to be too small. In actuality, what looks like gaps are simply weekends. This is a great example of why visual subjectivity cannot always be relied upon.

REFERENCES

1. DeTore and Frederic (2016). Introducing RIFT to Protect Your Uncertain Schedule. International Cost Estimating and Analysis Association workshop, Atlanta GA, 2016.
2. Joint Agency Cost and Schedule Risk and Uncertainty Handbook (CSRUH), 16 September 2014, https://www.ncca.navy.mil/tools/csruh/index.cfm
3. Seo (2002). A Review and Comparison of Methods for Detecting Outlies in Univariate Data Sets. University of Pittsburgh, 2006.
4. Estivill-Castro (2002). Why So Many Clustering Algorithms – A Position Paper. SIGKDD Explorations, Volume 4, Issue 1.
5. Dubes and Jain (1988). Algorithms for Clustering Data. Prentice Hall.
6. Kaufman and Rousseeuw (1990). Finding Groups in Data: an Introduction to Cluster Analysis. John Wiley and Sons.